Scene Analysis for Smart Devices and Immersive Technologies

Vladislav Li

Submitted in accordance with the requirements for the degree of
Doctor of Philosophy

1st Supervisor: Prof. Vasileios Argyriou
2nd Supervisor: Prof. Jean-Christophe Nebel

School of Computer Science and Mathematics

Kingston University, London

June 2024

Statement of originality

I, Vladislav Li, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material. I accept that the university has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Signature: Vladislav Li Date: June 10, 2024

Details of publications:

- **Vladislav Li**, Barbara Villarini, Jean-Christophe Nebel, Argyriou Vasileios. "A Modular Deep Learning Framework for Scene Understanding in Augmented Reality Applications", In IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), 2023.
- **Vladislav Li**, Barbara Villarini, Jean–Christophe Nebel, Thomas Lagkas, Panagiotis Sarigiannidis, Vasileios Argyriou. "Evaluation of Environmental Conditions on Object Detection Using Oriented Bounding Boxes for AR Applications", 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), 2023.
- **Vladislav Li**, George Amponis, Jean-Christophe Nebel, Vasileios Argyriou, Thomas Lagkas, Savvas Ouzounidis, Panagiotis Sarigiannidis. "Super Resolution for Augmented Reality Applications", In IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2022.
- **Vladislav Li**, Georgios Amponis, Jean-Christophe Nebel, Vasileios Argyriou, Thomas Lagkas, Panagiotis Sarigiannidis. "Object recognition for augmented reality applications", In Azerbaijan Journal of High Performance Computing, 2022.

Under review:

- **Vladislav Li**, Ilias Siniosoglou, Vasileios Argyriou, Thomai Karamitsou, Anastasios Lytos, Ioannis D. Moscholios, Sotirios K. Goudos, Jyoti S. Banerjee and Panagiotis Sarigiannidis. "Enhancing 3D Object Detection in Autonomous Vehicles Based on Synthetic Virtual Environment Analysis", under review, 2024.
- Georgios Tsoumplekas, **Vladislav Li**, Vasileios Argyriou, Anastasios Lytos, Eleftherios Fountoukidis, Sotirios K Goudos, Ioannis D Moscholios, Panagiotis Sarigiannidis. "Eval-

uating the Energy Efficiency of Few-Shot Learning for Object Detection in Industrial Settings", under review, 2024.

- Georgios Tsoumplekas, **Vladislav Li**, Vasileios Argyriou, Anastasios Lytos, Eleftherios Fountoukidis, Sotirios K Goudos, Ioannis D Moscholios, Panagiotis Sarigiannidis. "Toward green and human-like artificial intelligence: A complete survey on contemporary few-shot learning approaches", under review, 2024.

Abstract

Augmented Reality (AR) has emerged as an innovative technology with promising applications across various domains, including gaming, education, healthcare, and manufacturing. Enhancing the visual fidelity and efficiency of AR systems is crucial for delivering immersive and seamless user experiences. The AR systems rely on scene analysis techniques to obtain information about the surrounding real environment and use that information to superimpose digital 2D and 3D information. This thesis presents a novel scene analysis methodologies that integrates cutting-edge techniques for AR such as super-resolution, oriented bounding boxes, 3D bounding boxes, and data augmentation for FSL with energy-efficiency in mind. The thesis describes techniques of object detection model evaluation using modern tools and standards. Considering object detection, in the first part of the thesis, super-resolution techniques are analysed as a solution to the distant object detection, leveraging deep learning architectures to enhance the resolution and detail of captured scenes. By employing convolutional neural networks trained on high-resolution images, low-resolution inputs from AR devices can be upsampled to significantly improve visual quality and fidelity. Another perspective that integrates the oriented bounding boxes facilitates more accurate object detection and tracking in complex scenes. By representing objects as oriented bounding boxes aligned with their major axes, the proposed approach enhances the robustness and precision of object localisation, particularly in scenarios with rotated objects or occlusions. In addition to 2D bounding boxes, the thesis considers usage of 3D bounding boxes to enable more immersive spatial understanding and interaction within AR environments. By extending traditional bounding box representations into the three-dimensional space, the system enhances depth perception and facilitates realistic object manipulation and occlusion handling. Moreover, to address the challenge of limited labelled data in AR applications, data augmentation techniques for few-shot learning are evaluated. By artifi-

cially generating diverse training samples from a limited dataset, the system enhances the generalisation and adaptability of deep learning models, thereby improving object recognition and scene understanding in diverse environments. In addition to the data augmentation, energy efficiency is considered as part of the FSL design to ensure optimal performance on resource-constrained AR devices. By employing lightweight network architectures, efficient algorithms, and hardware-accelerated processing, the method minimises computational complexity and energy consumption while maintaining real-time performance and visual quality. The proposed object detection methodologies offer a comprehensive evaluation of the recent machine learning algorithms capable to support AR systems, enhancing visual fidelity, accuracy, adaptability, and energy efficiency. Through empirical evaluations and real-world implementations, this thesis demonstrates the effectiveness and practical viability of the proposed techniques, paving the way for more immersive and efficient AR experiences across various applications and domains.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Prof. Vasileios Argyriou, for his unwavering support and guidance throughout my studies. His mentorship and encouragement have been invaluable, and I am grateful for the opportunity to pursue my postgraduate studies under his supervision.

I am also indebted to Prof. Jean-Christophe Nebel for his insightful comments and recommendations. His analytical perspective has been instrumental in refining my work and highlighting important aspects of my writing.

Finally, I extend heartfelt thanks to my friends and family for their unwavering support and encouragement throughout this challenging period. Their belief in me has been a constant source of motivation, and I am grateful for their presence in my life.

Contents

List of Figures

List of Tables

List of abbreviations

AR	Augmented Reality
VR	Virtual Reality
MR	Mixed Reality
XR	eXtended Reality
AI	Artificial Intelligence
CV	Computer Vision
SR	Super Resolution
LR	Low Resolution
FSL	Few-Shot Learning
OBB	Oriented Bounding Boxes
DA	Data Augmentation
HMD	Head-Mounted Display
PC	Personal Computer
CNN	Convolutional Neural Network
R-CNN	Region-based Convolutional Neural Network
Fast R-CNN	Fast Region-based Convolutional Neural Network
FasteR R-CNN	FasteR Region-based Convolutional Neural Network
YOLO	You Only Look Once
SSD	Single Shot MultiBox Detector
GAN	Generative Adversarial Network
SLT	Statistical Learning Theory
HOG	Histogram of Oriented Gradients
SIFT	Scale-Invariant Feature Transform
NMS	Non-Maximum Suppression
SPPNet	Spatial Pyramid Pooling Network
RoI	Region of Interest
ReLU	Rectified Linear Unit

Chapter 1

Introduction

Contents

Immersive technology, and augmented reality in particular, stands at the forefront of modern human-computer interaction. In recent years, AR technology and AR applications have gained significant attention [115]. Observing our everyday routines, it is evident that the integration of AR applications into society has begun. There are some examples of AR applications that modern society has experienced, including the one that allows people to project virtual furniture inside their homes using regular smartphones [77]. Another example, in the field of sport, modern football matches utilise Video Assistant Referee (VAR) that projects data onto a screen to aid on-field referee draw a conclusion [149]. Researchers and engineers develop AR solutions, such as smart crossings [157], that aim to warn drivers and pedestrians to prevent accidents. In the entertainment business, a popular mobile video game called Pokemon Go utilised AR technology in order to facilitate novel gameplay mechanics [62].

More impactful examples could be observed blossoming in the healthcare field which have benefited from AR technology in tandem with state-of-the-art AI algorithms. For example, AR technology is capable of superimposing patient's information without completely obstructing the view of the user, whereas AI solutions are capable to detect objects to allow devices such as AR headsets to retrieve spatial information of a patient's body [32] to facilitate the projection of the digital data, like patient's information, on the real 3D environment. Together, AR technology and AI solutions allow surgeons to obtain vital information during an operation hands-free [177]. Thus, the surgeon's decision making process might be significantly improved and, consequently, decrease the overall mortality rate. AR technology could be applied to provide general practitioners with virtual patients and simulate their interactions using AI models. Such approach could help doctors at their earlier career stages to reduce the number of medication errors before working with the real people.

Another example of an application of AR technology is maintenance and remote maintenance in industry. AR maintenance has gained a lot of traction due to advancements in computer technology. It includes activities that aim at restoring product functionality [120]. AR maintenance has a great impact on improving the technical maintenance tasks and enhancing the maintenance managerial decision making processes. AR maintenance is encompassed with AI solutions that, for example, are used to detect hazardous environments or spot early signs of malfunctioning hardware. The examples discussed in [32] and [120] illustrate how the cost associated with repair work could be reduced if the hazards are detected early. Furthermore, AR maintenance could be applied in the aforementioned repair works. An AR application could visualise the steps that are needed to restore the functionality of a piece of equipment. Logically, such AR visualisations could also be communicated to an offsite expert who has greater knowledge removing the need for an expert to be physically on site [88]. This approach is better than a regular phone or video call to a support team because it provides the maintenance and expert with the spatial information acquired from the sensors of the AR headset using AI models. Consequently, such approach would affect the downtime times and the relevant costs.

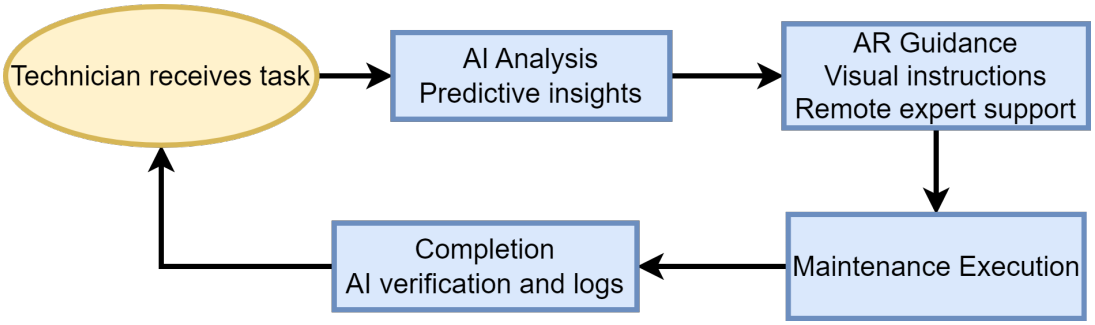


Figure 1.1: A flowchart visualising the concept of augmented reality and artificial intelligence in remote maintenance.

It is worth mentioning, AR technology could be applied in the military domain. Military AR applications could provide extra information in conjunction with spatial data acquired from sensors processed by AI algorithms. AR technology with AI solutions could act either as a first responder to notify the user about potential threats and pinpointing the information in the 3D environment or as a secondary source of information in situations where connection is limited or absent [143]. For example, special type of goggles could display visual hints whilst running in offline mode using local data and update once the connection to a server is re-established. However, to better appreciate the contributions to modern society that AR technology has made it is essential to understand the relevant computer technology advancements accumulated over the past century.

1.1 Smart Devices and Immersive Technology

Smart devices are such devices that are connected to another devices through networks and can interact, collect, analyse, and act on data autonomously. These devices range from everyday consumer products like smart thermostats and wearables to more complex systems used in industrial applications. Immersive technology, on the other hand, refers to technologies that create or enhance sensory experiences by merging physical and digital realities. This includes virtual reality, augmented reality, and colloquially referred to together as mixed reality and used interchangeably, which are designed to provide users with an engaging and interactive experience that blurs the line between the real and digital worlds.

Smart home devices include products like smart speakers, thermostats, lighting systems, security cameras, and smart appliances. These devices work together to create an interconnected ecosystem that can be controlled remotely and can automate various household tasks. Wearable technology includes devices such as smartwatches, fitness trackers, and health monitors that collect data on the wearer's activities, health metrics, and environment, providing valuable insights and promoting healthier lifestyles. In the context of smart cities, technologies like smart traffic lights, waste management systems, and public safety devices are used to improve urban living by enhancing efficiency, reducing waste, and ensuring safety. Industrial IoT (IIoT) devices, such as predictive maintenance sensors, smart meters, and connected machinery, are transforming manufacturing and other industries by enabling real-time monitoring and optimisation of operations. An example of a workflow is demonstrated in figure 1.1, it depicts the process of AI assisting maintenance by detecting early causes of breakdowns, or potential hazards that should be addressed where AR technology provides hands-on instructions and guidance to facilitate smoother maintenance execution alongside a remote assistance by an offsite expert if required. Lastly, the process could be further verified by AI system and generate new tasks based on the output of the system.

Immersive technology denotes a mechanism that embeds the user within the application's environment, fostering enhanced adaptability to the virtual landscape. This immersive engagement facilitates the seamless transmission of vital information through natural sensory channels such as sight, sound, and touch. Positioned as a focal point within the domain of mixed reality, inclusive of augmented and virtual realities, immersive technology is representative of contemporary discourse. The facilitation of immersive technology spans diverse modalities including 360-degree videos, virtual reality, augmented reality, mixed reality, and brain-computer interfaces (BCIs). Although, the aforementioned modalities are the most common types, there are other modalities that support the immersive technology. These modalities are haptic feedback, spatial audio, gesture recognition, and spatial mapping through scene analysis. These examples of immersive technology modalities are mediated

via hardware devices such as headsets, for example, an AR glasses.

Am important invention that shaped modern AR is Head-Mounted Displays (HMDs). In 1957, what was considered to be the first HDM was filed as a patent [72], titled Stereoscopic-Television Apparatus for Individual Use [51]. A Head-Mounted Display or HMD is a hands-free information source [107]. The initial prototypes were bulky and remained confined to research laboratories. However, thanks to the continuous technological progress, modern examples of HMDs, such as Microsoft HoloLens [111], Meta Quest 3 [109], and Apple VisionPro [78] became possible. Unlike more common devices like smartphones or PCs, HMDs are headsets that allow users to free their hands, thus broadening the spectrum of applications. For example, remote AR maintenance. AR glasses are a hands-free wearable that utilises advanced AI algorithms to achieve better immersion, subsequently, improving the overall experience.

Recently, immersive technology has gained a lot of traction due to advancements in computer technology [25], AR glasses achieve immersion by capturing and processing data from a broad spectrum of sensors. For example, positional tracking sensors such as Inertial Measurement Units (IMUs) provide rotational data from a gyroscope, and infer positional data from an accelerometer. There are instances where VR headset rely on lighthouse base stations to emit infrared light to track the position of the headset and controllers. Recent immersive technology wearables utilise inside-out tracking cameras. These cameras are built-in AR headsets to perform scene analysis to determine the user's position. The algorithms used to perform scene analysis must adapt to diverse environmental conditions like different light conditions, wide-ranging camera angles, numerous object poses, and even several weather conditions, e.g., raining and fog. The earlier mentioned immersion, along with the positional tracking sensors, is facilitated through gesture recognition sensors. Gesture recognition plays a crucial role in modern AR as it allows the user to interact with the environment hands-free. Gesture recognition sensors depend on infrared and depth cameras, e.g., Microsoft Kinect [183]. The depth cameras contain such information that allows software to construct a 3D

representation of the captured data. The 3D representation could be depicting hand interactions and specifically designed AI algorithms could infer those interactions and let software perform certain tasks like opening a menu in HoloLens [111].

In healthcare, smart devices and immersive technology are revolutionising patient care and medical training. Remote patient monitoring devices allow for continuous health tracking and telemedicine services, while VR and AR applications provide immersive training for medical professionals and support for rehabilitation exercises. However, this progress faces challenges, such as ensuring the accuracy and reliability of health data from remote monitoring devices, safeguarding patient privacy, and complying with healthcare regulations. Integrating VR/AR technologies into medical training and rehabilitation demands high levels of precision and real-time performance. Moreover, accessibility, affordability, and overcoming technical limitations in resource-constrained environments remain significant hurdles to widespread adoption. In education, smart devices and immersive technology offer interactive and personalised learning experiences, making it easier for students to grasp complex concepts and stay engaged. Despite these benefits, ensuring equitable access to these tools in under-resourced settings is a critical challenge. Developing engaging, effective content for diverse learning styles is complex, while maintaining user privacy, securing data, and integrating these technologies seamlessly into existing curricula and teacher training add further layers of difficulty. The entertainment and media industries are leveraging these technologies to create more interactive and immersive experiences, such as virtual concerts, augmented reality games, and immersive storytelling. Nonetheless, smooth real-time performance is essential to prevent disruptions during immersive experiences, particularly in live virtual events. Producing high-quality, interactive content accessible across various devices and platforms is resource-intensive. Additionally, addressing user privacy, data security, and free access to these technologies for diverse audiences is crucial. Retail and marketing sectors are using AR to enhance shopping experiences, allowing customers to virtually try on products and receive personalised recommendations. However, ensuring accurate virtual try-on experiences that realistically represent products in different lighting and environments is chal-

lenging. Privacy concerns around collecting and processing personal data, such as body measurements, need careful attention. Moreover, seamless AR integration across devices, maintaining real-time performance, and ensuring accessibility for all customers add to the complexity. In the workplace, immersive technologies facilitate virtual meetings, remote collaboration, and smart office environments, improving productivity and communication. Yet, reliable connectivity and smooth real-time performance are critical, particularly in resource-constrained environments. Safeguarding sensitive work data, ensuring seamless integration with existing workplace systems, and addressing user adoption and training needs are significant challenges. In maintenance, AR headsets provide hands-free remote assistance facilitated by scene analysis techniques that allow detection of hazard environments or early equipment malfunctions as well as hands-on instructions to maintain the equipment by projecting digital information onto the real environment. Challenges include ensuring accurate hazard detection and malfunction identification in dynamic settings, maintaining real-time performance for seamless instruction delivery, and addressing connectivity issues. Moreover, integrating AR systems with existing equipment and software, making the technology intuitive for users with varying technical expertise, and addressing privacy concerns related to sensitive operational data require careful planning and execution.

Eye tracking sensors enable gaze-based interactions. For example, situations, where surgeon's hands are busy, a specialised software designed to take advantage of gaze detection AI algorithms could provide vital information and superimpose it on the patient's body using AR technology. Current MR headsets, such as Meta Quest 3 [109], integrate infrared eye trackers to track the position of the user's eyes and pupils movements. Although, not seen in commercial off-the-shell devices, an electrooculography (EOG) sensors are used in research to establish a better quality-of-experience (QoE) assessments of AR/VR headsets [18]. A different domain of emotion recognition could also influence immersive technology by means of biometric sensors such as brainwave sensors [116]. Emotion recognition could enable personalised experience thus further immersing the user. Immersive technology fostered with AR headsets is reliant on sensors. The sensors are devices that collect data around AR user

including RGB cameras, depth sensors, motion sensors, IR sensors, light sensors. The collected data is then processed using AI algorithms to retrieve spatial information about objects of interest like cups, bowls, cars, people, airplanes, etc. For example, AR headset could adjust the brightness of the virtual environment based on the information about surrounding real environment collected from light sensor to better match the real one. Another example, AR application that runs on AR headset could superimpose a digital 3D object onto a real physical 3D object by determining its position, rotation, and scale from RGB cameras available on the device.

One of the main challenges of smart devices and immersive technology is interoperability, as integrating and ensuring compatibility among different devices and platforms can be complex. Security and privacy concerns are paramount, given the vast amount of data these devices collect and transmit. Ensuring robust security measures and protecting user privacy are critical to maintaining trust and safeguarding against cyber threats. User experience is another key consideration; designing intuitive and user-friendly interfaces is essential for widespread adoption and effective use of these technologies. Additionally, technical limitations such as latency and performance bottlenecks can impact the quality of immersive experiences, requiring ongoing innovation and optimisation.

Advancements in machine learning are significantly enhancing the capabilities of smart devices and creating more sophisticated and personalised immersive experiences. The rollout of 5G technology promises to improve the performance and connectivity of smart devices and immersive applications, enabling faster data transmission and more reliable connections. Edge computing is playing an increasingly important role in reducing latency and improving the efficiency of smart devices by processing data closer to the source. Emerging technologies, such as BCIs, have the potential to revolutionise the landscape of smart and immersive technology by providing new ways for humans to interact with digital environments and devices.

Although, according to the author in [24], the idea of AR was exploited since the early

days of humankind when people started changing their environment to improve their lives. For example, [24] states that when, in ancient times, people adorned their dwellings with drawings regarding their life experiences, it could be considered a form of augmenting reality. In this thesis, a contemporary definition provided by the same author would be taken into account rather then the broader idea discussed there earlier. AR is a medium in which digital information is overlaid on the physical world that is in both spatial and temporal registration with the physical world and that is interactive in real time [24]. While the concept of AR has a long history, modern advancements have enabled its practical implementation through HMDs and other technologies. The most recent approaches that allow overlaying of digital information are facilitated via different scene analysis techniques. Modern AR applications, particularly those utilising HMDs, often leverage AI for features like hand-tracking, object segmentation, and object detection. It is worth noting that the term Augmented Reality is used interchangeably with Mixed Reality (MR) and Extended Reality (XR) throughout the thesis.

The efficacy of immersive technology, with a particular focus on AR, hinges significantly on scene analysis, a concept rooted in Computer Vision (CV). Scene analysis entails the comprehension of the surrounding environment through data gathered from diverse sensors, including standard RGB cameras, infrared (IR) cameras, and depth cameras. The AR utilises the AI models to retrieve spatial information that enables immersive technology by superimposing digital information on the surrounding real 3D environment. Researchers employ state-of-the-art (SOTA) machine learning algorithms to process this data, addressing CV objectives such as object classification, localisation, and detection within the user's vicinity while utilising AR glasses.

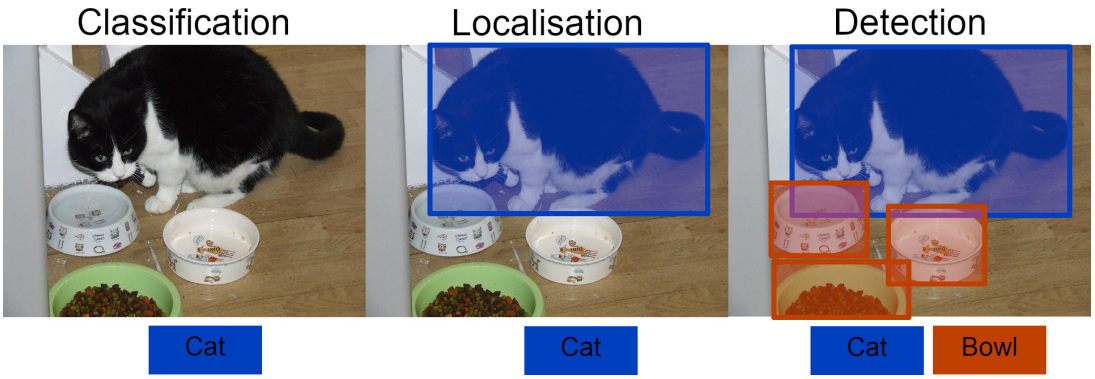


Figure 1.2: An example diagram highlighting the difference between the three tasks, from left to right: classification, localisation, and detection.

1.2 Detecting Objects on Smart Devices using Neural Networks

To understand object detection as a CV task, it is helpful to consider three computer vision tasks in increasing order of complexity [144]: a) classification, b) localisation, and c) detection. The classification task involves methods that are designed to assign a label to an image. Such assignment is performed in such a way as to match the label of an object in the image, i.e. ground-truth. The localisation task is developed on top of the classification task with an addition of *predicting* a bounding-box. In its most simple form, the bounding-box is represented as an axis-aligned rectangular covering the classified object in the image. Eventually, the object detection task is a continuation of the localisation task where the output is the classification with localisation of multiple objects in a single image as depicted in the figure 1.2.

In recent years, the advancement of neural networks, coupled with the ubiquity of smart devices, has disrupted the field of computer vision. Object detection is at the core of the scene analysis process that provides essential information about the objects present within a scene. It aims to locate and classify objects within an image or a video frame. With the proliferation of smart devices equipped with cameras, such as smartphones, tablets, and surveillance cameras, there has been a growing interest in deploying object detection algorithms

directly onto these devices. This shift towards on-device object detection brings numerous advantages, including real-time processing, enhanced privacy, and reduced dependency on cloud-based services. On the other hand, AR emerges as a game-changing technology, seamlessly integrating digital elements into the physical world, thus elevating users' experience and interaction with their immediate surroundings. Thus, central to AR experiences is the accurate recognition and tracking of real-world objects, enabling seamless integration of virtual content into the user's environment via scene analysis.

Neural networks have emerged as the cornerstone for object detection tasks, owing to their ability to learn complex patterns and features directly from data. Convolutional neural networks, in particular, have demonstrated remarkable performance in various computer vision tasks, including object detection. CNNs leverage hierarchical layers of neurons, which extract features of increasing complexity from input images. This hierarchical feature extraction makes CNNs well-suited for detecting objects of varying scales and orientations. One of the pioneering architectures in object detection is the Region-based Convolutional Neural Network (R-CNN) family, which includes R-CNN, Fast R-CNN, and FasteR R-CNN. These models adopt a two-stage approach, where candidate object regions are first proposed using region proposal algorithms (e.g., Selective Search), followed by classification and bounding box regression using CNNs. While effective, these approaches suffer from slow inference speeds, hindering their deployment on resource-constrained smart devices. To address the limitations of two-stage approaches, single-stage detectors such as You Only Look Once (YOLO) and Single Shot MultiBox Detector (SSD) have gained prominence. These models operate by directly predicting object bounding boxes and class probabilities in a single pass through the network, leading to faster inference speeds. This efficiency makes single-stage detectors particularly well-suited for on-device deployment, where real-time performance is crucial.

Despite advancements in neural network architectures, deploying object detection models on smart devices presents several challenges. One significant challenge is the trade-off

between model complexity and inference speed. While complex models may achieve superior accuracy, they often require a large number of parameters and computational resources, making them unsuitable for deployment on resource-constrained devices. Another challenge is the limited availability of labelled training data that is representative of the target deployment environment. Object detection models trained on datasets that differ significantly from the target domain may suffer from poor generalisation and performance degradation when deployed on smart devices. Furthermore, on-device object detection must adhere to strict constraints on memory usage, power consumption, and processing time. These constraints necessitate the optimisation of neural network architectures, inference algorithms, and hardware accelerators to achieve efficient real-time performance on smart devices. Object detection on smart devices using neural networks represents a promising frontier in computer vision. Leveraging the power of neural networks, researchers and developers can create efficient and accurate object detection solutions tailored for deployment on resource-constrained devices. Addressing the challenges associated with on-device object detection requires innovative approaches in model design, optimisation techniques, and hardware integration. By overcoming these challenges, on-device object detection has the potential to enable a wide range of applications, including augmented reality, autonomous vehicles, remote maintenance, and smart surveillance systems, thereby enhancing the capabilities of smart devices and improving user experiences.

1.3 Object Recognition for Augmented Reality Applications

Object recognition in AR typically involves the identification and localisation of objects within the user's field of view using computer vision algorithms. Several techniques have been developed to achieve robust object recognition in real-time AR scenarios. Feature-based methods, such as SIFT (Scale-Invariant Feature Transform) [101] and SURF (Speeded-Up Robust Features) [11], detect distinctive keypoints and descriptors in the image, allowing for accurate matching and localisation of objects. These methods are robust to changes in scale, rotation, and illumination, making them suitable for a wide range of AR applications.

Template matching is another popular approach for object recognition in AR, where a template image of the object of interest is compared against regions in the input image to find the best match. While simple and intuitive, template matching may suffer from scalability issues and sensitivity to variations in appearance and viewpoint. Deep learning-based approaches, particularly Convolutional Neural Networks (CNNs), have gained prominence in recent years for object recognition in AR. CNNs learn hierarchical representations of objects directly from data, enabling highly accurate and robust recognition across diverse object categories. Models such as FasteR R-CNN, YOLO, and SSD have been adapted for real-time object detection in AR environments, offering superior performance compared to traditional methods.

Accurate object recognition is essential for delivering compelling and immersive AR experiences to users. By recognising and understanding the user's surroundings, AR applications can overlay relevant digital content, such as annotations, 3D models, and contextual information, seamlessly onto physical objects. This integration of virtual and real-world elements enhances users' perception and interaction with their environment, enabling novel applications in gaming, education, navigation, retail, and beyond. Furthermore, object recognition enables AR applications to interactively respond to changes in the environment, such as object occlusions, lighting variations, and dynamic scenes. Real-time tracking and relocalisation of objects ensure the continuity and stability of AR overlays, providing users with a seamless and immersive experience.

Despite the advancements in object recognition for AR, several challenges remain to be addressed. These include robustness to occlusions, variations in lighting and viewpoint, real-time performance on resource-constrained devices, and scalability to large-scale environments. Future research directions in object recognition for AR may involve leveraging multimodal sensor data, such as depth sensors and inertial sensors, to enhance the robustness and accuracy of object localisation and tracking. Additionally, techniques for semantic understanding of objects and scenes could further enrich AR experiences by enabling contex-

tual interactions and intelligent content placement. Object recognition plays a critical role in enabling immersive and interactive AR experiences by accurately detecting and tracking real-world objects in the user's environment. Leveraging a combination of traditional computer vision techniques and deep learning approaches, AR applications can seamlessly integrate virtual content into the physical world, unlocking new possibilities for entertainment, education, productivity, and beyond. As research and technology continue to advance, object recognition will remain a cornerstone of AR development, driving innovation and enhancing user experiences in the years to come.

1.4 Challenges

Scene analysis is a critical component of smart devices and AR applications, allowing them to interpret and interact with the surrounding environment. However, this process faces numerous challenges that must be addressed to ensure seamless and effective implementation. In this section, we explore the multifaceted challenges encountered in scene analysis for smart devices and AR, encompassing technical, computational, and practical considerations. The complexity of real-world scenes poses a significant challenge for scene analysis algorithms. Scenes can contain a diverse array of objects, textures, shapes, and lighting conditions, necessitating robust algorithms capable of handling occlusions, clutter, and ambiguities. Accurately recognising and tracking objects within a scene is fundamental but challenging. Algorithms must contend with occlusion handling, scale and viewpoint variations, real-time performance, and robustness to changes in lighting and environmental conditions.

Smart devices and AR applications demand real-time performance from scene analysis algorithms to deliver responsive and interactive experiences. The real-time performance of scene analysis algorithms is considered to be speed of execution of an algorithm. Such speed is usually measured in frames per second (fps) or milliseconds and the common target is 24 fps or 41.67 ms. Although, considering that scene analysis algorithms could run as background processes and depending on the task the target speed could change. For example,

static surveillance cameras that monitor slow moving objects like pedestrians target 15 fps or 66.67 ms. Nonetheless, achieving high frame rates and low latency while maintaining accuracy and robustness poses significant computational challenges, especially on resource-constrained devices. Scalability is essential as scenes become larger and more complex. Algorithms must scale to handle increasing amounts of data and computational complexity, requiring efficient data structures, parallel processing techniques, and distributed computing approaches. Memory and power efficiency are crucial for smart devices with limited resources. Optimising algorithms for efficiency without sacrificing performance is necessary for sustainable user experiences.

Training scene analysis models requires large amounts of labelled data, which may be costly or challenging to acquire. Developing efficient data acquisition pipelines and automated annotation tools can help accelerate model development. Privacy and security concerns arise when processing sensitive information, such as images or videos captured by smart devices. Ensuring data privacy, compliance with regulations, and mitigating risks of data breaches are paramount for maintaining user trust. Integration into smart devices and AR applications requires seamless compatibility with existing platforms, APIs, and development tools. Consideration of hardware constraints, software frameworks, and deployment environments is essential for successful integration. Overcoming the challenges of scene analysis for smart devices and augmented reality requires interdisciplinary research efforts spanning computer vision, machine learning, sensor technology, and human-computer interaction. By addressing technical, computational, and practical hurdles, researchers and developers can unlock the full potential of scene analysis technologies, enabling smart devices and AR applications to interpret, understand, and interact with the world in increasingly intelligent and immersive ways.

1.5 Contributions

Recent years have witnessed substantial progress in the domain of computer vision and object detection, particularly concerning scene analysis for smart devices and AR. These advancements have propelled the development of more robust, efficient, and accurate algorithms for interpreting and understanding the surrounding environment. This thesis presents a comprehensive analysis of various methods and frameworks for enhancing AR systems through advanced scene analysis techniques. Key contributions include:

1. This research contributes to the field of real-time object detection and classification for AR/VR by proposing a novel framework for smart mobile devices. While advancements in machine learning and powerful processors enable near real-time object recognition, this work focuses on overcoming limitations for mobile integration. The proposed approach realises a novel end-to-end framework that utilises modern SR techniques based on GANs for the object detection task. The proposed framework combines SR and object detection processes allowing an optimisation of SR reconstruction error. Furthermore, a comparative study was undertaken using the end-to-end framework to evaluate the performance under variety environmental conditions that could affect the immersive experience in AR. Finally, a new dataset that captures a broad spectrum of environmental conditions was introduced. This framework aims to take a low-resolution input and generate a high-resolution model capable of object recognition. This research explores the potential of this SR-infused framework for mobile AR/VR applications.

2. The objective of AR is to add digital content to natural images and videos to create an interactive experience between the user and the environment. Scene analysis and object recognition play a crucial role in AR, as they must be performed quickly and accurately. This contribution evaluates the proposed approach that involves using oriented bounding boxes with a detection and recognition deep network to improve performance and processing time. The approach is evaluated using two datasets: a real image

dataset (VisDrone dataset) commonly used for computer vision tasks, and a synthetic dataset that simulates different environmental, lighting, and acquisition conditions. The focus of the evaluation is on small objects, which are difficult to detect and recognise. The results indicate that the proposed approach tends to produce better Average Precision and greater accuracy for small objects in most of the tested conditions.

3. This contribution addresses the challenge of limited labelled data in AR applications by proposing a data augmentation method specifically designed for few-shot learning (FSL). This method aims to improve object recognition and scene understanding in diverse environments by artificially diversifying the available scarce data to reach competitive performance with the methods that rely on large datasets. Furthermore, the method prioritises energy efficiency by, in addition to DA, incorporating lightweight and efficient network architecture, i.e., YOLO facilitated by SOTA PAN and GELAN, and hardware-accelerated processing. This focus on efficiency ensures optimal performance on resource-constrained AR devices, making the solution well-suited for real-world applications such as remote AR maintenance.

4. This contribution introduces a comprehensive framework for AR systems, emphasising advanced 3D scene analysis techniques. Key contributions include the integration of 3D object detection algorithm to enhance spatial understanding and interaction within AR environments. The 3D object detection algorithm extends SOTA anchor-free technique allowing a methodology that improves depth perception and facilitates realistic object manipulation, thereby enhancing the overall AR experience.

1.6 Outline of the Thesis

The thesis begins with an overview and objectives, situated within the context of AR scene understanding. Following with the Related Work chapter. This chapter reviews existing literature on AR scene understanding, environmental conditions' impact on object detection, techniques for data augmentation in few-shot learning, and methods for 3D object detec-

tion in AR. The Modular Deep Learning Framework for Scene Understanding in Augmented Reality Applications chapter presents a novel deep learning framework for AR scene understanding, this chapter discusses its architecture, individual modules, integration, and experimental results. The Evaluation of Environmental Conditions on Object Detection Using Oriented Bounding Boxes for AR Applications chapter examines object detection in varying environmental conditions, this chapter evaluates the effectiveness of oriented bounding boxes, detailing the experimental setup, dataset used, and analysis of results.

The Data Augmentation for Few-Shot Learning Object Detection chapter addresses challenges in few-shot learning for object detection, this chapter proposes data augmentation techniques, emphasising their importance, describing methods, and evaluating impact. Lastly, the Analysis of 3D Object Detection for Augmented Reality Using a Synthetic Dataset chapter utilises a synthetic dataset, this chapter analyses 3D object detection methods in AR, discussing dataset construction, evaluation methodology, and result interpretation. Finally, the Conclusions chapter summarises key findings, discusses contributions to the field, and suggests potential future research directions in AR scene understanding and object detection.

Chapter 2

Related work

Contents

2.1 Foundational Theories in Object Detection

Object detection serves as a cornerstone in scene analysis for smart devices and immersive technology, such as AR applications, facilitating the interpretation and interaction with the surrounding environment. In this section, we explore foundational theories in object detection within the domain of computer vision, focusing on their relevance to scene analysis for smart devices and AR [92]. By understanding these theories, we can gain insights into the underlying principles driving object detection algorithms and their implications for practical implementation in real-world applications.

According to [117], the initial ideas of AI were conceptualised between 1943-56. The authors in [105] conceived a model of artificial neural networks where each neuron was considered to be in binary state, either **on** or **off**. The model has proved that simple structures could learn. A simple concept inspired by the inner working of human brain such as like this has enabled a research in the field of computer vision and later became one of the crucial enablers of the modern AR technology.

Statistical Learning Theory (SLT) is one of the foundational theories in modern object detection, even though its origins date back to the late 1960s [158]. Despite its age, the core principles of SLT remain highly relevant. SLT focuses on the concept of learning from data, drawing inspiration from the field of statistics. It supports supervised learning, where labelled data and algorithms are used to train models. Notably, SLT allows to analyse model's complexity and reason about trade-off between accuracy and efficiency. This is crucial for real-world applications where resource constraints exist. Furthermore, SLT delves into the concept of generalisation, which empowers models to perform well even on unseen data – a critical capability for robust object detection.

Feature representation forms the basis of object detection algorithms, enabling the extraction of relevant information from input images. Traditional methods, such as Histogram of Oriented Gradients (HOG) [31] and Scale-Invariant Feature Transform [101] (SIFT), rely on handcrafted features to capture distinctive patterns in images. These methods provide a foundation for early object detection approaches but are limited in their ability to generalise to diverse object categories and variations in appearance. In contrast, deep learning-based approaches modernise feature representation by learning hierarchical representations directly from data [144], [56]. CNNs automatically learn meaningful features from raw pixel values, capturing both low-level visual patterns and high-level semantic information [131], [132], [133]. The hierarchical nature of CNNs enables them to capture complex object representations, making them well-suited for object detection in diverse scenes and environments.

Object detection frameworks provide a systematic approach for localising and classify-

ing objects within images. Traditional methods, such as sliding window-based approaches and cascade classifiers, sequentially evaluate candidate regions in the image to detect objects. While effective, these methods suffer from high computational complexity and limited scalability, particularly in real-time applications on resource-constrained devices [131], [74]. Modern object detection frameworks, such as R-CNN and its variants (Fast R-CNN, FasteR R-CNN), adopt a two-stage approach [56], [134]. They first generate region proposals using techniques like Selective Search or Region Proposal Networks (RPNs) and then classify and refine these proposals using CNNs. This two-stage paradigm improves detection accuracy and efficiency, paving the way for real-time object detection on smart devices and AR platforms.

Contextual understanding plays a crucial role in object detection, enabling algorithms to leverage contextual cues and relationships between objects to improve detection performance. Traditional methods often treat object detection as an isolated task, focusing solely on local image features. However, objects in the real world exhibit contextual dependencies, such as spatial relationships, co-occurrences, and semantic associations. Recent advancements in object detection incorporate contextual understanding through techniques such as contextual modelling [12], graph-based representations [176], and attention mechanisms, [48]. These approaches enable algorithms to exploit contextual information to refine object localisation and classification, leading to more accurate and robust detection results, especially in complex scenes and cluttered environments.

The foundational theories in object detection outlined earlier have profound implication for scene analysis on smart devices and AR applications. Firstly, real-time performance, deep learning-based object detection frameworks, optimised for efficiency and speed, enable real-time performance on resource-constrained smart devices. This facilitates seamless integration of scene analysis capabilities into AR applications, empowering users with interactive and responsive experiences. Secondly, robustness and generalisation, by leveraging hierarchical feature representations and contextual understanding, object detection algorithms

demonstrate improved robustness and generalisation across diverse scenes and environments. This enhances the reliability and accuracy of AR overlays and interactions, ensuring consistent performance in various real-world scenarios. Thirdly, privacy-preserving solutions, with growing concerns about data privacy, advancements in object detection enable privacy-preserving solutions by performing scene analysis directly on-device without transmitting sensitive information to external servers. Techniques such as federated learning and differential privacy ensure that user data remains secure and confidential [85], mitigating privacy risks associated with cloud-based processing.

Foundational theories in object detection describe the concepts and the ideas for scene analysis on smart devices and augmented reality applications. By understanding the principles of SLT, feature representation, detection frameworks, and contextual understanding, researchers and developers can design more robust, efficient, and privacy-preserving object detection algorithms tailored for real-world deployment. These theories serve as the building blocks for advancing the capabilities of smart devices and AR platforms, driving innovation and enhancing user experiences in various domains.

2.2 Object Detection in Various Domains

Object detection, a fundamental task in computer vision, plays a pivotal role across various domains, each presenting unique difficulties and opportunities. In the context of autonomous driving [141], object detection algorithms are essential for enabling vehicles to perceive and navigate safely through complex environments. These algorithms must accurately detect and localise objects such as vehicles, pedestrians, cyclists, and traffic signs in real-time, amidst diverse lighting conditions, occlusions, and unpredictable movements. Within the domain of surveillance and security [150], object detection assumes a critical role in monitoring and analysing video feeds from surveillance cameras deployed in public spaces and high-security facilities. These algorithms enable the identification and tracking of suspicious activities, unauthorised individuals, and potential threats, facilitating timely intervention and effective

security measures.

An interesting application of object detection in combination with AR glasses could be observed in the industrial domain, more specifically, maintenance or remote maintenance. The idea is not new and has been on the radars of the research interest for some time [8]. However, with the advent of computer technology and raise of AR, the concept of AR remote maintenance became popular [104]. The concept revolves around the process of monitoring, diagnosing, troubleshooting, and repairing equipment or systems from a location that is physically distant from the equipment or systems being serviced. The AR remote maintenance would connect an off-site expert with the on-site technician who seeks support or guidance to service the equipment or systems. A whole spectrum of technologies are involved in enabling the AR remote maintenance concept such as internet connections, AR glasses, and specialised software capable of projecting digital information onto the surrounding environment as well as retrieving information from the physical environment, e.g., spatial mapping, and communicating that information back and forth.

In medical imaging [166], object detection algorithms serve as valuable tools for assisting radiologists and clinicians in diagnosing diseases and localising abnormalities. For instance, in mammography, these algorithms aid in the early detection of breast cancer by accurately detecting and localising breast lesions [6]. Similarly, in radiology, object detection algorithms identify and quantify abnormalities such as tumours, cysts, and fractures, contributing to accurate diagnosis and treatment planning [175]. As a subsequent turn, these data could be then digitally overlaid onto patients to assist general practitioners during their work whilst wearing an AR headset. Another example, revolves around improving the caregiver job by bridging Internet-of-Things (IoT) with AR. The authors in [83] argue the need for AI assisted AR technology to better facilitate intelligent healthcare for ageing population. In industrial automation [140], object detection is integral to tasks such as object manipulation, quality inspection, and inventory management. These algorithms identify defective products, locate components on assembly lines, and optimise warehouse operations, thereby improving effi-

ciency and productivity. The industrial AR automation benefits from object detection in the same manner as the regular automation with an addition of overlaying digital information onto physical devices using scene analysis to improve the industrial processes and reduce interruptions. Automatic detection of hazards or faults on the line with an intuitive depiction of the problem by the means of AR glasses hold a great potential in reducing costs and outages. Overall, object detection plays a crucial role in enhancing safety, security, efficiency, and decision-making across diverse domains. Continued research and innovation in object detection methodologies are essential for advancing the capabilities and applications of computer vision technology to address real-world challenges effectively.

2.3 Evolution of Object Detection Algorithms

The progression of object detection algorithms in the field of computer vision has been characterised by significant advancements in accuracy, speed, and robustness. Beginning with traditional methods reliant on handcrafted features and classifiers, the field has undergone a transformative shift towards deep learning-based approaches, fundamentally altering the landscape of object detection methodologies [86]. Traditional techniques, such as Haar cascades [160], illustrated in figure 2.1, HOG [31], and support vector machines (SVM) [15], laid the groundwork for object detection by extracting manually engineered features and training classifiers to identify objects within images. However, these methods often encountered challenges with complex backgrounds, occlusions, and variations in scale and viewpoint.

For example, in crowded urban scenes, cluttered backgrounds consisting of various objects make it difficult for these methods to differentiate pedestrians accurately, leading to false positives or missed detections [39]. Additionally, pedestrians can be partially occluded by objects or other pedestrians, disrupting the patterns captured by Haar cascades or HOG features and resulting in incomplete or incorrect detections. Furthermore, variations in scale and viewpoint of pedestrians, influenced by their distance from the camera and walking direction, pose challenges for traditional methods that often rely on fixed-size templates or

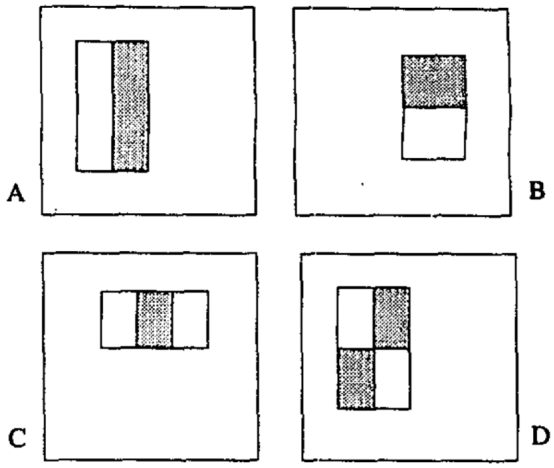


Figure 2.1: Example of rectangle features shown relative to the enclosing detection window. The sum of the pixels which lie within the white rectangles are subtracted from the sum of pixels in the grey rectangles. Two-rectangle features are shown in (A) and (B). Figure (C) shows a three-rectangle feature, and (D) a four-rectangle feature. [160].

descriptors, leading to missed detections or false alarms. These limitations highlight the need for more advanced techniques, such as deep learning-based object detection models, which can learn rich representations of objects and effectively handle these challenges through end-to-end training on large datasets. In parallel, augmented reality applications rely on machine learning and computer vision techniques to detect physical objects in the real world. This allows virtual objects to be added and rendered in real-time. In recent years, the use of deep CNNs has significantly enhanced the performance and accuracy of computer vision tasks such as object detection and recognition allowing it's better synergy with the augmented reality [57], [56], [68], [131], [133].

One of the machine learning methods that contributed to transition towards the deep neural networks is the OverFeat methodology [144], introduced by Sermanet et al. in 2013, transformed object detection by proposing a unified framework that integrates recognition, localisation, and detection tasks using CNNs. Unlike traditional methods that handle these tasks separately, OverFeat demonstrates the effectiveness of jointly optimising them within a single architecture. It employs a sliding window approach for object detection, where a window of varying sizes is slid across the input image, and a CNN is applied to each win-

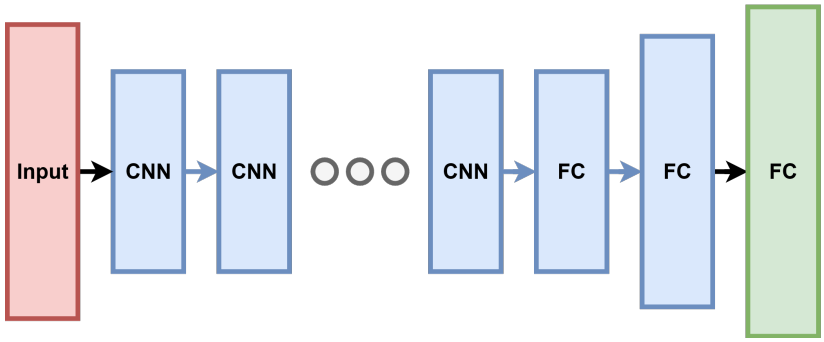


Figure 2.2: An abstract representation of the OverFeat deep neural network architecture. FC is a Fully Connected layer.

dow for classification and localisation. This method enables OverFeat to handle objects of different scales and aspect ratios efficiently in a single pass through the network.

The architecture of the OverFeat methodology, depicted in figure 2.2, is rooted in a deep CNN tailored to integrate object recognition, localisation, and detection tasks seamlessly. It begins with an input layer accepting images of varying sizes, which are typically resized to a fixed dimension. Convolutional layers follow, serving to extract hierarchical features from the input image through convolutional operations. These features are then passed through max pooling layers for downsampling, aiding in spatial reduction while preserving vital information. Subsequently, fully connected layers process the extracted features to perform high-level representation and classification, ultimately outputting class probabilities for object recognition.

OverFeat employs a sliding window mechanism where a window of varying sizes slides across the input image, with each window region considered a candidate detection area. These window regions are then resized to a fixed dimension and fed into a fully connected layer of the CNN. The fully connected layer processes the features extracted from each window region to predict class probabilities and bounding box coordinates. The fully connected layer outputs class scores representing the probability of an object belonging to each predefined class, as well as bounding box coordinates representing the spatial extent of the detected object within the window region. OverFeat utilises non-maximum suppression (NMS)

to filter out redundant bounding boxes and refine the final detections [144].

One of the key innovations of OverFeat is its incorporation of multi-scale feature maps at different layers of the CNN. By fusing information from multiple scales, OverFeat improves the robustness of object detection and localisation, particularly for small objects or objects at different depths within the image. Moreover, OverFeat adopts an end-to-end training approach, allowing the entire network architecture to be trained jointly from raw pixel data. This enables OverFeat to automatically learn hierarchical representations of objects, leading to superior performance compared to handcrafted feature-based methods. In terms of performance, OverFeat achieves impressive results, at the time when it was published, in object localisation and detection accuracy on benchmark datasets such as PASCAL VOC and ImageNet [144]. By jointly optimising recognition, localisation, and detection tasks, OverFeat significantly advances the state-of-the-art performance in object detection. Its contributions have had a profound impact on the field of computer vision, inspiring subsequent research efforts in deep learning-based object detection methodologies and paving the way for further advancements in complex vision tasks.

In 2014, Girshick et al. introduced the R-CNN method for object detection [57]. This approach involved identifying potential object boxes through selective search and rescaling each box to a fixed-size image for input into a CNN model trained on AlexNet [160] for feature extraction. Object detection was then performed using a linear SVM classifier. While R-CNN achieved a significant improvement in mAP compared to previous methods, it also faced a drawback of slow detection speed. The R-CNN methodology employed a dual-stage process. Firstly, it commenced with the utilisation of "selective search" to identify prospective object boxes within an image. Selective search effectively partitioned the image into multiple regions or proposals that were considered as likely candidates harbouring objects. These regions were thereby considered as candidate boxes for potential object localisation.

The second stage of the R-CNN procedure entailed the resizing of these candidate boxes into standardised, fixed-size images, rendering them amenable for subsequent analysis. These

standardised images were subsequently subjected to a CNN architecture, specifically pre-trained on the AlexNet model [160]. The principal role of this CNN was to perform feature extraction, thereby discerning and capturing highly distinctive features inherent to the object. Upon feature extraction, the final step of the R-CNN methodology involved the employment of a linear SVM classifier. The SVM classifier was instrumental in effecting classification of the extracted features, thereby ascertaining the presence or absence of a given object within the candidate box. This classification process was the crucial part of object identification and localisation.

The outcomes of the R-CNN approach bore substantial significance. It led to a noticeable increase in performance using the mAP metric, a versatile measure of the efficiency and precision of object detection algorithms. Effectively, it surpassed antecedent methods in its competence to identify objects within images, marking a substantial progression in the arena of computer vision. Nevertheless, it is imperative to acknowledge a noteworthy constraint accompanying this advancement. Notably, the R-CNN method was afflicted by a comparatively prolonged detection timeframe. The sequential nature of its operations, encompassing selective search, CNN-based feature extraction, and SVM classification, caused computational intensiveness and increased processing times, thereby limiting its utility in scenarios necessitating real-time object detection.

Consequently, while the R-CNN approach bestowed enhanced object detection capabilities, its computational demands and temporal constraints demanded further research endeavours aimed at enhancing its operational speed [68]. Nonetheless, its establishment solidified a significant milestone in the progression of object detection algorithms, laying the groundwork for subsequent innovations and catalysing advancements within fields such as autonomous systems, robotics, and diverse facets of computer vision applications. In an effort to tackle the persistent challenge of slow detection speed in object recognition and localisation, He et al. presented the Spatial Pyramid Pooling Network (SPPNet) as an innovative solution in their seminal work [69]. This architectural paradigm marked another

notable milestone in the evolution of computer vision, offering a profound remedy to a long-standing predicament in the field.

The importance of the SPPNet's success lay in its strategic incorporation of a Spatial Pyramid Pooling (SPP) layer, an important component that transformed the object detection process. The distinctive feature of this SPP layer was its ability to generate a fixed-length representation that remained invariant to alterations in image size and scale. The SPP layer achieves this by dividing the input feature map into a grid of sub-regions and then pooling features separately within each sub-region. This pooling operation aggregates information from each sub-region, allowing the network to capture spatial information at multiple scales. Importantly, the pooling is performed in a manner that ensures the output representation has a fixed size, irrespective of the input image size and scale. This attribute had far-reaching implications, particularly in terms of mitigating overfitting issues that had previously plagued object recognition systems.

The invariance to image size and scale was a notable development. It meant that the SPPNet was endowed with the unique capability to seamlessly handle images of varying dimensions during the training phase. This adaptability was a critical departure from prior approaches, which often required resizing or normalisation of images before processing. By eliminating this constraint, SPPNet opened the door to greater flexibility and robustness, rendering it suitable for applications where objects of interest might appear in various sizes and scales. Within the sphere of object detection, the SPPNet contributed to an ingenious paradigm shift. It introduced the concept of calculating feature maps just once for the entire image, a departure from the conventional methods that performed feature extraction independently for each candidate region. After this initial feature extraction step, the SPPNet employed a sub-region pooling mechanism. This operation entailed dividing the image into spatial bins, enabling the aggregation of features from each bin to create fixed-length representations that were conducive for detector training.

One of the most notable outcomes of this innovative approach was a remarkable acceler-

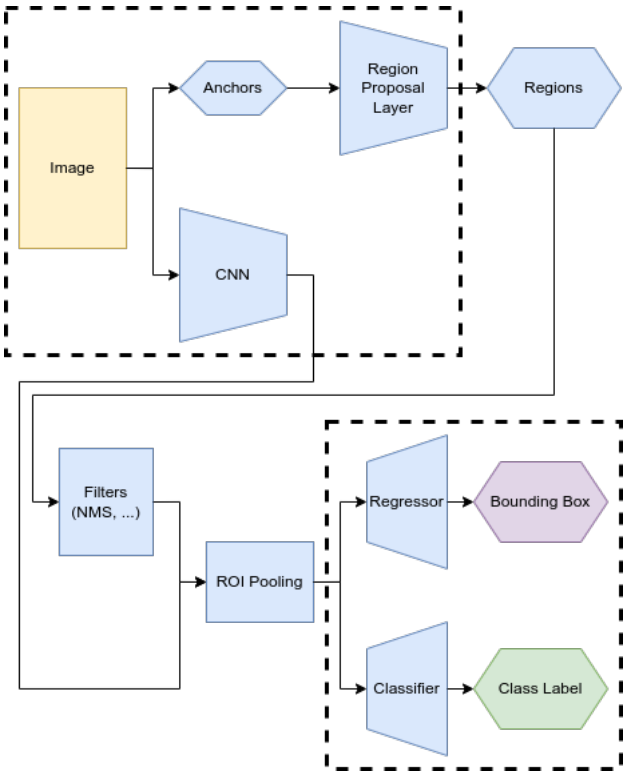


Figure 2.3: An abstract of the FasteR R-CNN architecture.

ation in processing speed, especially during testing [125]. The SPPNet method proved to be a significant leap forward, with testing times ranging from 24 to 102 times faster than the previously established R-CNN approach. This acceleration in speed held profound implications for real-time and time-sensitive applications, particularly in contexts like autonomous vehicles, robotics, and augmented reality. Following the R-CNN model, in 2015, Girshick introduced an enhancement to the existing architectural paradigms in the form of the Fast R-CNN [56]. This novel network configuration entailed the simultaneous training of both an object detector and a bounding box regression component, all within the same unified architecture. The Fast R-CNN model, distinguished by its unified architecture, stands as a hallmark of efficiency. It exhibited the remarkable capability to concurrently train two fundamental components within the same framework: an object detector and a bounding box regression module. This integrated approach was a significant stride towards a more streamlined and coherent training process.

Furthermore, the Fast R-CNN model introduced a groundbreaking technique known as Region of Interest (RoI) pooling, which efficiently extracts features from proposed regions, enhancing the model's ability to precisely locate objects within images. By seamlessly integrating RoI pooling into its architecture, Fast R-CNN demonstrated exceptional performance in accurately identifying and delineating objects even in complex scenes. This comprehensive approach not only accelerated the training process but also facilitated advancements in real-time object detection systems, significantly impacting the field of computer vision applications. However, it is noteworthy that the issue of computational speed constraints persisted despite this development. In the same time, Ren et al. introduced the FasteR R-CNN detector [134], as viewed in figure 2.3, a groundbreaking endeavour that charted a course toward the realisation of real-time object detection through the prism of end-to-end training. The FasteR R-CNN architecture marked a seminal turning point in the pursuit of faster detection capabilities. At its core, it introduced the Region Proposal Network (RPN), a component specifically designed to accelerate the object detection process. The RPN's mandate involved the generation of region proposals, an aspect that greatly enhanced the network's adeptness in efficiently discerning objects within complex scenes.

The architecture of FasteR R-CNN can be dissected into several key components, demonstrated in the figure 2.3. Firstly, it utilises a CNN backbone, such as VGG or ResNet, to extract features from the input image. These features serve as the basis for identifying potential RoIs where objects might be located. The key innovation of FasteR R-CNN lies in its RPN, which efficiently generates these RoIs. Unlike its predecessor, which relied on selective search to propose regions, RPN operates as a fully convolutional network, enabling it to generate region proposals in a single forward pass. Following the generation of region proposals, FasteR R-CNN employs a RoI pooling layer to extract fixed-size feature maps from each proposal. These feature maps are then fed into a series of fully connected layers, known as the R-CNN, for object classification and bounding box regression. The classification network assigns a class label to each RoI, while the regression network refines the bounding box coordinates to better fit the object within the proposal. By jointly optimising both tasks, FasteR R-CNN

ensures accurate localisation and classification of objects within the input image.

The introduction of the FasteR R-CNN model had an indelible impact on the landscape of computer vision [68]. It not only showed the possibility of near real-time object detection but also spurred a wave of innovative architectural variants. These variations, with an overarching focus on curtailing computational redundancy [30], [95], [98], explored diverse avenues to further amplify the velocity and efficiency of object detection while preserving precision. Among these progressive adaptations, the D2Det method, introduced by Cao et al. in 2020 [16], stands out as an exemplar of innovation based on the FasteR R-CNN framework. The D2Det method harnesses a sophisticated two-stage process for handling RoI features. In the initial phase, high-density local regression is employed to finetune the localisation of objects, infusing a heightened degree of precision into the detection process. Subsequently, in the second stage, a discriminant RoI pooling mechanism extracts distinctive features from the RoIs. Notably, D2Det departs from the FasteR R-CNN's offset regression by adopting a local dense regression block, thus augmenting the precision and robustness of the object detection process.

The collaborative endeavours of researchers have achieved a notable milestone in the evolution from Fast R-CNN to FasteR R-CNN and beyond. This ongoing progression signified a dynamic enhancement in the pursuit of real-time object detection capabilities. These advancements held immense potential to transform various fields such as autonomous systems, surveillance, robotics, and augmented reality [56], [68]. The relentless pursuit of faster, more precise, and efficient object detection methods remained the driving force at the forefront of innovation in computer vision and deep learning. The methodologies discussed above fell under the classification of two-stage detectors due to their characteristic two-step process: initially generating RoIs and subsequently executing detection and recognition. In 2016, Joseph et al. introduced a noteworthy departure from this convention, presenting a one-stage detector known as You Only Look Once (YOLO) [131]. YOLO introduced an observable change in the domain of object detection, conceptualised as a single network architecture

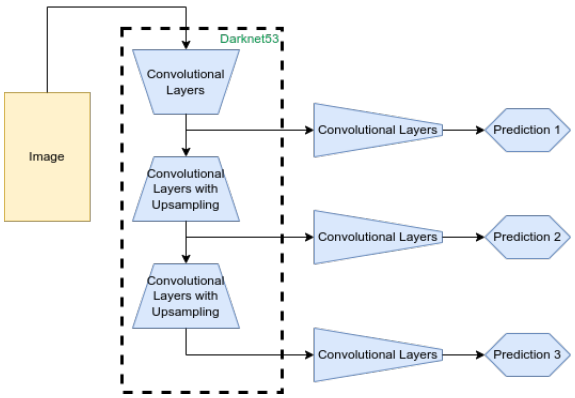


Figure 2.4: An abstract representation of the YOLOv3 architecture.

capable of processing the entirety of an image within a single step, which resulted in substantially improved processing times.

The YOLO methodology operates by segmenting the image into distinct regions and concurrently predicting bounding boxes for each of these regions. This one-step processing concept enabled a departure from the multi-step procedures of its two-stage counterparts. YOLO was a better balance between speed and accuracy in the field, representing a novel approach to object detection [131]. Unlike its two-stage counterparts, YOLO employed a single neural network architecture, capable of processing an entire image in a single pass. This unique design offered a significant advantage in terms of processing speed, effectively reducing detection times. In the subsequent years, YOLO underwent iterations with the introduction of YOLO v2 and v3, depicted in figure 2.4, aimed at enhancing prediction accuracy [132], [133]. The YOLOv3 architecture, viewed in the figure 2.4, consists of three main components: the backbone network, the detection head, and the output. The backbone network, typically a CNN like Darknet-53, extracts features from the input image. These features are then passed through several convolutional layers to capture increasingly abstract representations of the image.

The detection head is responsible for predicting bounding boxes and class probabilities for objects within each grid cell. YOLOv3 predicts bounding boxes at three different scales to detect objects of varying sizes. Each bounding box is associated with a confidence score that

indicates the likelihood of the box containing an object and class probabilities for the detected objects. Finally, the output is generated by combining the predictions from all grid cells and applying non-maximum suppression to remove redundant detections, resulting in a final list of detected objects along with their bounding boxes and class labels. While YOLO excelled in terms of speed, it encountered challenges related to localisation accuracy. This trade-off spurred further research efforts to fine-tune the model. To redress this trade-off and enhance the localisation accuracy, Liu et al. introduced the Single Shot MultiBox Detector (SSD) in 2016 [102]. As a result, the SSD network shows a slight improvement, surpassing YOLO in the PASCAL VOC detection task [46]. The SSD methodology followed the one-stage concept by integrating multi-reference and multi-resolution detection strategies, offering the capacity to detect objects at varying scales across different strata of the network. This architecture was marked by its ability to flexibly accommodate objects of diverse sizes and magnitudes within the image, mitigating the aforementioned accuracy compromise.

Building upon the foundation laid by SSD, Lin et al. presented RetinaNet in 2018 [99], representing a notable evolution in one-stage object detection. The key innovation within RetinaNet was the introduction of a novel loss function termed "focal loss." This loss function, distinct from the conventional cross-entropy loss, was designed to impart a higher degree of attention to instances that were persistently misclassified during the training phase. This heightened attention to challenging examples during training resulted in an enhanced level of prediction accuracy, outstripping the performance of its one-stage counterparts. The focal loss function modifies the standard cross-entropy loss by introducing a modulating factor that down-weights the loss assigned to well-classified examples, focusing instead on hard, misclassified examples. This is particularly useful in object detection, where the number of background (non-object) examples vastly outweighs the number of foreground (object) examples, leading to class imbalance issues. The focal loss function effectively reduces the contribution of easy examples to the overall loss, thus improving the model's ability to focus on learning from challenging examples.

Mathematically, the focal loss function $FL(p_t)$ could be defined as:

$$FL(p_t) = -(1 - p_t)^{\gamma} \log(p_t)$$

where p_t is the predicted probability of the ground-truth class and γ is a tunable focusing parameter. The term $(1 - p_t)^{\gamma}$ is the modulating factor, which increases as p_t decreases, thereby emphasising the contribution of misclassified examples. When p_t is close to 1 (i.e., well-classified examples), the modulating factor approaches 0, effectively reducing the loss contribution of these examples. Conversely, when p_t is close to 0 (i.e., misclassified examples), the modulating factor increases, amplifying the loss contribution and encouraging the model to focus on learning from these hard examples. The focal loss function is applied independently to each anchor box across all spatial locations and object classes. This allows the model to effectively handle the class imbalance and focus on learning from challenging examples, ultimately improving detection performance. Additionally, RetinaNet combines the focal loss with other components such as a feature pyramid network (FPN) and a regression loss for bounding box localisation, resulting in a robust and efficient object detection framework suitable for various applications.

In contemporary developments within the domain of object detection, there was a noteworthy shift towards anchor-free methodologies. These novel approaches, in contrast to conventional techniques, emphasised the inference of bounding box corners, rather than reliance on pre-defined bounding boxes. A prominent example of this trend is the CenterNet, an innovative framework introduced by Zhou et al. [186]. Notably, CenterNet has distinguished itself as a state-of-the-art solution for 3D Lidar-based detection and tracking, showcasing its versatility in diverse applications. CenterNet can be perceived as an evolution of the CornerNet, another anchor-free approach to bounding box detection that represents objects as pairs of keypoints, specifically the top-left and bottom-right corners. These corner keypoints are extracted through a technique known as corner pooling, which was introduced by the same authors [89]. A critical stride in the advancement from CornerNet to CenterNet

was the introduction of a central keypoint, a concept that facilitated the association of corner keypoints with objects depicted in images. This novel approach has demonstrated superior performance compared to conventional anchor-based solutions, such as FasteR R-CNN and YOLO, marking a significant advancement in object detection.

Continuing the trajectory of innovation, in 2020, Perez-Rua and colleagues introduced the OpeN-ended Center nEt (ONCE) [123]. ONCE extended the functionality of CenterNet by empowering it to detect objects from classes with limited examples in its training dataset, a noteworthy feat that holds promise for applications involving a wide variety of object categories. In the most recent developments, object detection techniques have begun to explore the capabilities of transformers, as exemplified by the DEtection TRansformer (DETR) method introduced by Carion et al. [17]. This exploration leverages the advantages of transformer architectures, which have gained prominence in natural language processing, and integrates them into the object detection domain. What sets DETR apart is its simplicity, coupled with performance that rivals other sophisticated detection techniques employed in the field.

Subsequently, Zhu et al. proposed Deformable DETR, building upon the foundation laid by DETR, with the specific objective of addressing the challenge of detecting small objects. This enhancement aimed to achieve state-of-the-art performance, underscoring the commitment of the scientific community to continuously refine and advance object detection methodologies to meet the evolving demands of real-world applications. Additionally, approaches have been developed to enhance the detection of small objects, which is particularly challenging as they have fewer visible details using Super Resolution (SR) solutions. Commonly Super Resolution solutions are relying on Generative Adversarial Networks (GAN) [59]. Such methodologies have proved to be particularly successful [9][91]. Indeed, their competitive process involving two neural networks, i.e., a generation network and a discriminant network, ensures that the generated images are as realistic as possible.

More recent, YOLOv8, or You Only Look Once version 8 [161], represented a signific-

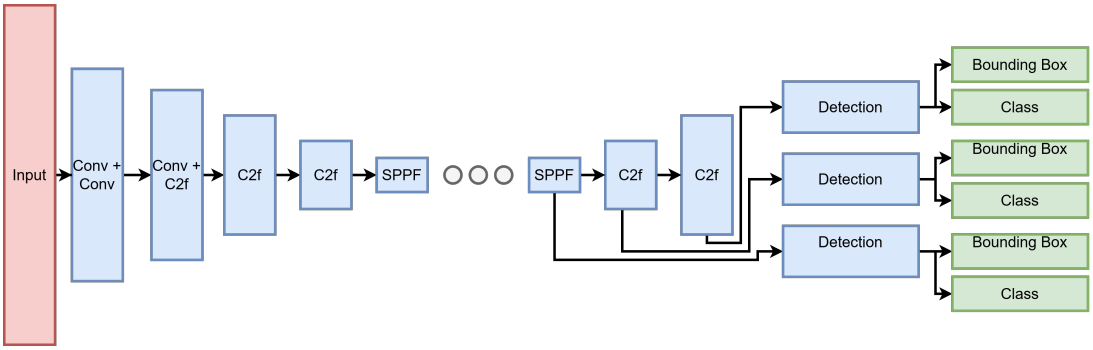


Figure 2.5: An abstract representation of the YOLOv8 architecture.

ant advancement in the field of object detection within computer vision. Building upon its predecessors, YOLOv8 integrated state-of-the-art techniques to achieve higher accuracy and faster inference speeds, making it an attractive choice for real-time applications. One of the key innovations of YOLOv8 is its architecture, displayed in figure 2.5, which combined the efficiency of a single neural network with the accuracy of feature extraction. By utilising a unified architecture, YOLOv8 is able to detect objects in an image or video feed with remarkable speed and precision, outperforming many traditional methods that rely on multiple stages of processing.

The YOLOv8 C2f (CSP Bottleneck) block is an element within the YOLOv8 architecture, important for enhancing feature extraction and overall model performance. This block is founded on the concept of Cross Stage Partial (CSP) connections, aimed at facilitating information flow between various stages of the network while simultaneously reducing computational overhead. It achieves this through a series of well-designed operations. At the outset, the C2f block initiates with the concatenation of feature maps derived from two distinct paths within the network. This concatenation process enables the network to integrate information from multiple scales and levels of abstraction, thereby augmenting its capability to discern intricate patterns and structures within the input data. Subsequently, the concatenated feature maps traverse through a sequence of convolutional layers.

These convolutional layers serve the purpose of spatial filtering operations, extracting high-level features from the input data. By applying multiple convolutional filters, the net-

work can capture a diverse range of features at varying spatial scales. Within the C2f block, a bottleneck structure is employed, which consists of a series of convolutional layers with reduced channel dimensions followed by a layer with increased channel dimensions. This bottleneck structure helps to diminish the computational cost of the convolutional operations while preserving the representational capacity of the network. In addition to the convolutional layers, the C2f block also integrates a residual connection. This connection allows the output of the block to bypass the convolutional layers and be directly added to the input feature maps. Residual connections play a role in mitigating the vanishing gradient problem during training and enable more effective information flow through the network. Finally, the output of the C2f block undergoes an activation function, typically a rectified linear unit (ReLU), which introduces non-linearity into the network and aids in capturing complex relationships within the data. Overall, the YOLOv8 C2f block significantly contributes to feature extraction and representation learning within the network, leading to enhanced object detection performance.

Another notable feature of YOLOv8 is its ability to handle a wide range of object classes with varying sizes and aspect ratios. Through techniques such as multi-scale detection and feature pyramid networks, YOLOv8 excels at detecting objects of different scales within the same image, ensuring comprehensive coverage and robust performance across diverse datasets. YOLOv8 also incorporates advanced training strategies to improve its performance further. Techniques like data augmentation, transfer learning, and curriculum learning are employed to enhance the model's ability to generalise to unseen data, resulting in more reliable object detection capabilities in real-world scenarios. Moreover, YOLOv8 places a strong emphasis on efficiency, striving to achieve high accuracy without compromising on speed. Through optimisations such as model pruning, quantisation, and parallel processing, YOLOv8 can deliver real-time object detection on resource-constrained devices, opening up possibilities for applications in fields such as autonomous driving, surveillance, and robotics.

YOLOv8 stands out in the landscape of object detection algorithms when compared to its

predecessors like YOLOv3, RetinaNet, and FasteR R-CNN. While each of these models has its strengths and weaknesses, YOLOv8 offers a compelling combination of accuracy, speed, and efficiency [81]. Compared to YOLOv3, YOLOv8 introduces several improvements in architecture and training strategies. YOLOv8 achieves higher accuracy through enhancements such as feature pyramid networks and multi-scale detection, allowing it to better handle objects of varying sizes and aspect ratios. Additionally, YOLOv8 incorporates advanced training techniques like transfer learning and curriculum learning to improve generalisation and adaptability to different datasets. In contrast to RetinaNet, which focuses on addressing the challenge of object detection in the presence of class imbalance, YOLOv8 takes a different approach by emphasising efficiency and speed. While RetinaNet achieves impressive accuracy by balancing the contributions of positive and negative samples during training, YOLOv8 leverages a unified architecture and optimisation techniques like model pruning and quantisation to achieve real-time performance without sacrificing accuracy.

Model pruning involves removing redundant or unnecessary parameters from a neural network without significantly impacting its performance. This process can reduce the model's size, making it more memory-efficient and faster to execute [66]. Pruning techniques can vary from simple methods like weight magnitude pruning, where weights below a certain threshold are set to zero, to more sophisticated approaches like iterative pruning algorithms that iteratively remove less important connections or neurons. Quantisation, on the other hand, involves reducing the precision of the model's parameters and activations, typically from 32-bit floating-point numbers to lower bit representations like 8-bit integers. By using fewer bits to represent numbers, quantisation reduces memory usage and speeds up computation, especially on hardware platforms that are optimised for integer operations. However, quantisation may introduce some loss of accuracy, particularly in models with high precision requirements. When compared to FasteR R-CNN, which relies on a two-stage detection pipeline involving RPN and R-CNNs, YOLOv8 distinguishes itself by its single-stage architecture. By eliminating the need for explicit region proposal generation and subsequent refinement, YOLOv8 streamlines the detection process, resulting in faster inference speeds

and reduced computational complexity.

An introduction of YOLO-MS or You Only Look Once - Multi Scale [20] provided us with an introduction of a new multi-scale strategy as an alternative to CSPNet or FPN. Furthermore, the authors [20] also advance the research in large kernel convolution, i.e. using uncommon kernel size greater than 3×3. The first contribution, or the authors refer to it as MS-Block [20], improves the multi-level feature extraction by implementing a hierarchical feature fusion strategy [20]. The second contribution explores the possibilities of speed improvement by utilising bigger kernel sizes in the context of performance-constrained hardware.

The most recent development in the field of single stage detectors offers YOLOv9 or You Only Look Once version 9. The authors in [164] argue that a substantial chunk of information is lost when transferred from one layer to another due to a phenomenon called information bottleneck [155]. YOLOv9 introduces PGI or Programmable Gradient Information that addresses the deep supervision for shallow architectures and GELAN or Generalised Efficient Layer Aggregation Network that unlike ELAN [163] allows the developers to integrate any block, for example CSP block, to be installed in the framework. The introduced combination boost the performance for the lightweight configurations of the YOLOv9 and utilising less parameters at the same time.

Recent advancements in Large Language Model (LLM) architectures, such as GPT [127], PaLM [21], and multimodal models like GPT-4 Vision [3], are significantly enhancing object detection and scene analysis by integrating textual and visual understanding. These models leverage cross-modal learning, where they process both textual descriptions and visual data to achieve a deeper semantic understanding of scenes. For example, by training on paired image-text datasets, LLMs with visual extensions can identify objects in a scene and contextualise their relationships, enabling more nuanced scene descriptions and reasoning. This capability surpasses traditional object detection, which primarily identifies objects and their bounding boxes, by adding interpretive layers that explain how objects interact or contribute to the overall context of the scene. Such advancements have applications in fields

like autonomous driving, robotics, and content moderation, where understanding complex scenes is critical.

Moreover, LLMs now contribute to few-shot or zero-shot learning in object detection [10] and scene analysis, reducing the dependency on large, labelled datasets. With their ability to generalise from textual prompts, these models can identify new object categories or infer contextual relationships without extensive retraining. For instance, a multimodal LLM can analyse a traffic scene and understand abstract concepts like "heavy traffic" or "pedestrian safety risks" without being explicitly trained on those specific terms. Additionally, the architectures' transformer-based attention mechanisms enable them to focus on relevant areas of an image or scene dynamically, ensuring high accuracy in tasks like detecting occluded objects or analysing cluttered scenes. This synergy of vision and language is propelling breakthroughs in AI-driven visual perception, bridging the gap between machine-level perception and human-like understanding.

The Vision Transformer (ViT) [36] represents a paradigm shift in computer vision, drawing inspiration from principles originally developed for large language models (LLMs). At its core, ViT leverages the Transformer architecture, renowned for its self-attention mechanisms and scalability, to process image data as a sequence of patches. This innovative approach replaces traditional convolutional operations with a model that can learn global dependencies across an image, akin to how LLMs understand relationships within textual sequences. By dividing images into fixed-size patches and treating them as tokens, ViT capitalises on the strengths of pre-training on massive datasets, enabling it to achieve state-of-the-art performance on vision benchmarks, particularly when fine-tuned for specific tasks. The success of ViT demonstrates the viability of cross-domain architectural paradigms, underscoring the growing convergence between vision and language modelling.

The adaptation of LLM principles into visual tasks has been particularly transformative in enabling models like ViT to handle multimodal data. By treating image patches analogously to words or tokens in text, ViT facilitates a unified representation space where vision and

language inputs can be processed seamlessly. This has spurred advancements in tasks such as image captioning, visual question answering, and cross-modal retrieval, where integrated understanding is paramount. Moreover, the use of large-scale self-supervised learning, a cornerstone of LLMs, has been adapted for vision tasks, further reducing reliance on labelled data and enhancing model generalisation. As researchers continue to refine the application of LLM-inspired methods to vision, the boundaries between these domains are becoming increasingly blurred, heralding a new era of multimodal machine learning systems.

Transformer-based architectures, such as the Vision Transformer (ViT), enhance spatial feature extraction by modelling global dependencies across an image through self-attention mechanisms, which capture relationships between distant regions more effectively than traditional convolutional neural networks [17]. This ability to consider context holistically significantly improves performance in scene analysis, as models can understand complex spatial arrangements and interactions within a scene. In real-world augmented reality applications, such capabilities contribute to more robust object detection by enabling the precise identification of objects, even under challenging conditions such as occlusions or dynamic lighting. Furthermore, these architectures can integrate multi-scale features, ensuring finer granularity in object localisation while maintaining a broader contextual understanding, which is crucial for immersive and responsive augmented reality experiences. The result is an enhanced capacity for real-time analysis and interaction with complex environments, making these models well-suited for next-generation technological applications.

2.4 Enhancing Few-Shot Object Detection Through Data Augmentation Techniques

The object detection task attempts to solve the challenge of predicting multiple objects in a single image. However, it often suffers from scarcity of data. This is a general problem for most of machine learning tasks [168]. To solve such problem, recently, Few-Shot Learning (FSL) has been proposed [49], [5]. FSL aims at generalising to new tasks using few samples.

The FSL could be applied in the terms of AR remote maintenance to quickly adjust towards the environment on the rare equipment. The FSL problem could be addressed in various ways, although, most of the solutions focus on one of the following stages of the training process: data, model, and algorithm [168]. The algorithm stage is an extra stage where a specifically designed algorithm is trying to analyse and learn the training process [49]. The model stage focuses on the architecture of an AI model able to generalise well enough even from a scarce amount of data.

The data stage relies on Data Augmentation (DA). DA is a term to describe generalisation techniques that reduce overfitting of a machine learning model by diversifying the data [87]. This is performed using techniques such as rotation, translation, shearing, cropping, flipping, colour jittering, Cutout [33], random erasing, Mixup [181], AutoAugment [26], and Style Randomisation [79]. The aforementioned techniques are used as a way of regularisation and often increasing the amount of data to train on addressing the FSL problem. However, most of the research on data augmentation techniques has focused on the classification task. The data augmentation techniques for the object detection task require to handle bounding boxes in addition to images [189].

In the domain of data augmentation, a myriad of techniques exists, broadly categorised into two primary groups [87]. The first category pertains to 'Image Manipulation,' a term necessitating explication. 'Image Manipulation' encompasses operations such as rotation and cropping, which induce alterations in the visual representation of an image. The second category comprises techniques characterised by 'Image Erasing Manipulation,' necessitating elucidation akin to the former category. Techniques within this classification include, among others, 'Jitter' and 'Cutout'.

Rotation - the rotation data augmentation technique is a fundamental method employed to enhance the diversity of training data in machine learning and computer vision applications, an example could be observed in figure 2.6. This technique involves rotating an image around its centre by a certain angle, introducing variations in its orientation without alter-

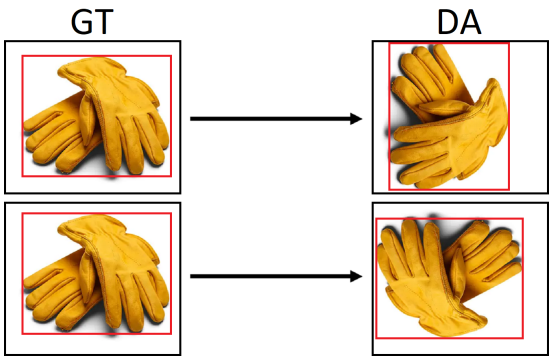


Figure 2.6: Example of the 'Rotate' Data Augmentation technique. Red colour indicates the bounding box and black colour represents the edges of the image. GT stands for Ground-Truth, DA stands for Data Augmentation.

ing its intrinsic content. By applying rotation augmentation, models become more robust to variations in object poses, contributing to improved generalisation during training [146]. The degree of rotation can be controlled to strike a balance between generating diverse training samples and maintaining the integrity of the original data. This augmentation technique is particularly valuable in scenarios where objects may exhibit different orientations, ensuring that the model learns to recognise patterns from various perspectives, thereby enhancing its overall performance and adaptability. Given a 2D rotation in a Cartesian coordinate system, the rotation of an image could be described using the formula below,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.1}$$

where (x, y) are the original coordinates of a point, (x', y') are the coordinates after rotation by an angle θ, and and $\cos(\theta)$ and $\sin(\theta)$ are the cosine and sine functions of the rotation angle.

Translation - the translation data augmentation technique is a pivotal strategy employed to augment training datasets in machine learning and computer vision, as depicted in the figure 2.7. This technique involves shifting an image along its spatial dimensions, both horizontally

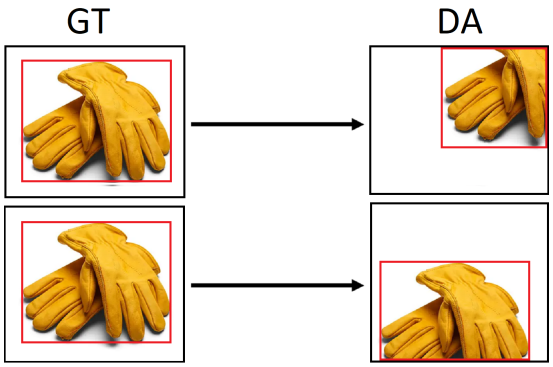


Figure 2.7: Example of the 'Translate' Data Augmentation technique. Red colour indicates the bounding box and black colour represents the edges of the image. GT stands for Ground-Truth, DA stands for Data Augmentation.

and vertically, introducing positional variations while preserving the inherent content. By applying translation augmentation, models can better generalise to diverse spatial arrangements of objects, ultimately improving their robustness [33]. This technique is particularly effective in scenarios where the precise positioning of objects is subject to variation, enabling the model to learn invariant features across different spatial locations. Controlling the magnitude of translation allows for a fine-tuned balance between generating diverse training instances and maintaining the contextual coherence of the original data. As a result, the translation augmentation technique plays a crucial role in enhancing the model's ability to recognise and adapt to objects across different positions within an image. In a similar way, given a 2D translation in a Cartesian coordinate system, the translation of an image could be described using the formula below,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{2.2}$$

where (x, y) are the original coordinates of a point, (x', y') are the coordinates after translation, and (t_x, t_y) are the translation amounts in the x and y directions, respectively.

Shearing - the utilisation of the shearing data augmentation technique is a valuable ap-

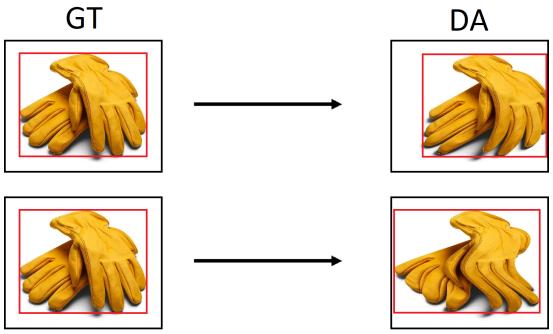


Figure 2.8: Example of the 'Shear' Data Augmentation technique. Red colour indicates the bounding box and black colour represents the edges of the image. GT stands for Ground-Truth, DA stands for Data Augmentation.

proach, aiming to enrich the diversity of training data, as demonstrated, as presented in figure 2.8. This technique involves the deformation of an image by altering the position of pixels along a specific axis, producing a shearing effect. Typically applied in both horizontal and vertical directions, shearing introduces geometric distortions that enable models to better handle variations in object shapes and orientations [33]. By incorporating shearing augmentation, the model becomes more adept at recognising objects from different perspectives, contributing to improved generalisation during training. The magnitude of shearing can be controlled to strike a balance between generating diverse training samples and maintaining the intrinsic structure of the original data. This technique is particularly beneficial in scenarios where objects may undergo deformations due to perspective changes, providing the model with a more comprehensive understanding of object geometry and fostering increased adaptability in real-world applications. Given a 2D shearing in a Cartesian coordinate system, the shearing of an image could be described using the formula below,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & s_x \\ s_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.3}$$

where (x, y) are the original coordinates of a point, (x', y') are the coordinates after shearing, and (s_x, s_y) are the shear factors that determine the amount of shearing in the x and y

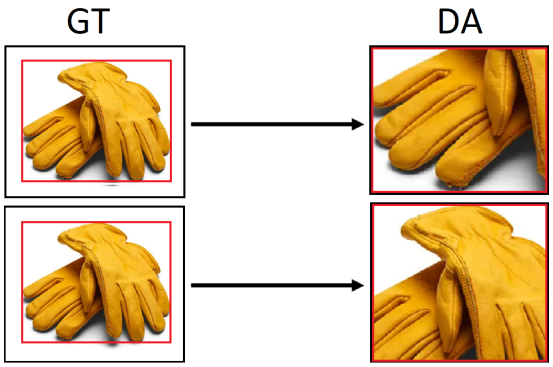


Figure 2.9: Example of the 'Crop' Data Augmentation technique. Red colour indicates the bounding box and black colour represents the edges of the image. GT stands for Ground-Truth, DA stands for Data Augmentation.

directions, respectively.

Crop - the cropping data augmentation technique is a significant strategy in the domain of machine learning and computer vision, contributing to the enrichment of training datasets, serving to augment the richness of training datasets; an example could be seen in figure 2.9. This method involves the selective removal of portions of an image, typically from its periphery or random locations, resulting in a modified spatial composition. By implementing cropping augmentation, models can better adapt to variations in object positioning and scale, enhancing their robustness. This technique is particularly advantageous in scenarios where the precise location or size of objects may vary, enabling the model to learn and generalise across diverse spatial configurations. The extent of cropping can be controlled, striking a balance between generating diverse training instances and preserving the essential contextual features of the original data. Thus, the cropping augmentation technique significantly contributes to the model's capacity to recognise and understand objects across a spectrum of spatial contexts, promoting improved performance and versatility. The cropping operation of an image I that selects a rectangular region defined by the coordinates $(x_{\text{min}}, y_{\text{min}})$ for the top-left corner and $(x_{\text{max}}, y_{\text{max}})$ for the bottom-right corner could be expressed with the following notation:

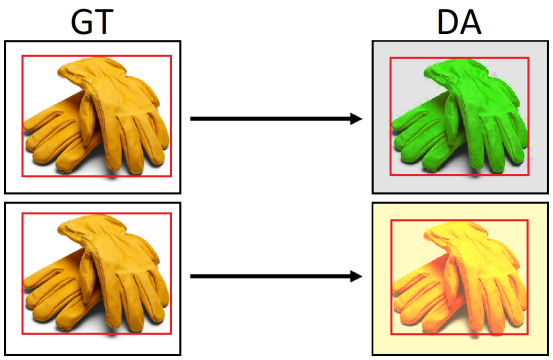


Figure 2.10: Example of the 'Jitter' Data Augmentation technique. Red colour indicates the bounding box and black colour represents the edges of the image. GT stands for Ground-Truth, DA stands for Data Augmentation.

$$I_{\text{cropped}} = \{(x, y) \mid x_{\text{min}} \leq x \leq x_{\text{max}}, \\ y_{\text{min}} \leq y \leq y_{\text{max}}\} \tag{2.4}$$

where the notation describes the set of points (x, y) that belong to the cropped image I_{cropped} and both x and y fall within the specified range.

The colour jitter data augmentation technique is a versatile method employed in machine learning and computer vision to enhance the diversity of training data, an example is shown in figure 2.10. This technique introduces random variations in the brightness, contrast, and saturation of an image, thereby generating a range of visually distinct representations. By incorporating colour jitter, models become more robust and adaptable to variations in lighting conditions and colour intensities [86]. The random adjustments mimic real-world scenarios, enabling the model to better generalise across different environmental conditions. Controlling the degree of jitter allows for a fine-tuned balance between generating diverse training instances and maintaining the essential features of the original data. This augmentation technique plays a crucial role in improving the model's ability to recognise and classify objects under varying colour and lighting conditions, contributing to overall performance and generalisation.

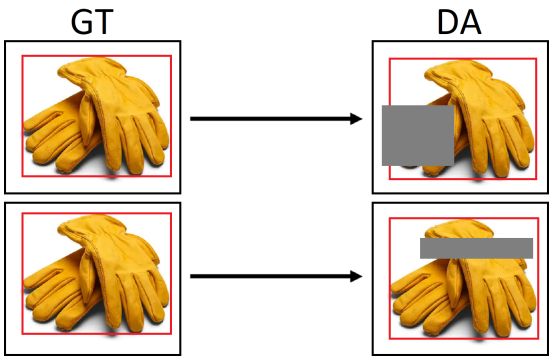


Figure 2.11: Example of the 'Cutout' Data Augmentation technique. Red colour indicates the bounding box and black colour represents the edges of the image. GT stands for Ground-Truth, DA stands for Data Augmentation.

The Cutout data augmentation technique is a method commonly used in machine learning and computer vision to enhance the robustness and generalisation of models as seen in figure 2.11. It involves randomly removing square regions, or 'cutouts,' from an image during the training process. The objective is to force the model to focus on different features and prevent it from relying too heavily on specific pixel patterns, thereby improving its ability to generalise to various image variations. Cutout helps in regularising the model by introducing a form of spatial dropout, and it is particularly effective in preventing overfitting [33], [71]. The size and placement of the cutout regions can be controlled, allowing for a balance between diversifying the training data and maintaining the integrity of the image's content. This technique has proven beneficial in scenarios where models need to perform well on images with different spatial configurations and occlusions.

The Random Erasing data augmentation technique is another method employed in machine learning and computer vision to enhance the robustness of models during training as depicted in figure 2.12. This technique involves randomly selecting and erasing rectangular regions within an image, replacing the erased regions with random pixel values or other predefined values. The primary objective of Random Erasing is to simulate occlusions and encourage the model to learn more robust and invariant features. By introducing these random erasures, the model becomes less sensitive to specific patterns or details, improving its

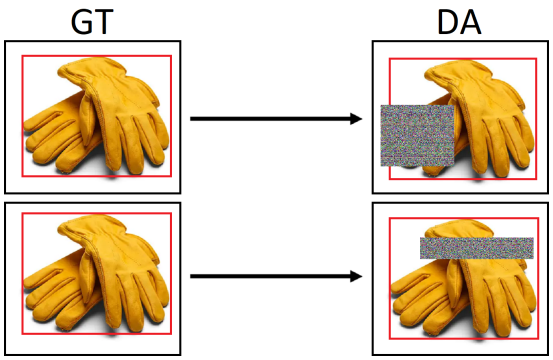


Figure 2.12: Example of the 'Random erase' Data Augmentation technique. Red colour indicates the bounding box and black colour represents the edges of the image. GT stands for Ground-Truth, DA stands for Data Augmentation.

ability to generalise to unseen data and variations [185]. Similar to other data augmentation techniques, the parameters controlling the size, aspect ratio, and probability of erasure can be adjusted to tailor the level of augmentation. Random Erasing is particularly useful in scenarios where occlusions or missing information in images are common, contributing to the model's adaptability and performance in real-world conditions.

The author of [19] introduces an innovative Context-Guided data augmentation approach designed to tackle the challenge of occluded objects within images. In contrast to the conventional method of randomly removing portions of an image, the proposed technique strategically introduces new instances of classes onto images, leveraging contextual information to simulate occlusion occurrences. The contextual information is acquired through the initial execution of a ResNet model. Subsequently, the Context-Guided method selectively positions a segmented instance onto the bounding box in a randomised manner. The aforementioned technique could be seen as an alternative to a Random Erasing data augmentation technique as pointed out by the authors. An example of the Context-Guided is presented in figure 2.13.

In the realm of advanced data augmentation techniques, a diverse array of methods can be categorised into distinct groups, each contributing uniquely to the augmentation landscape. 'Image Mixing Data Augmentations' encompasses approaches like Mixup [181] and CutMix, where images are blended or sections are transposed to create novel, hybrid samples, foster-

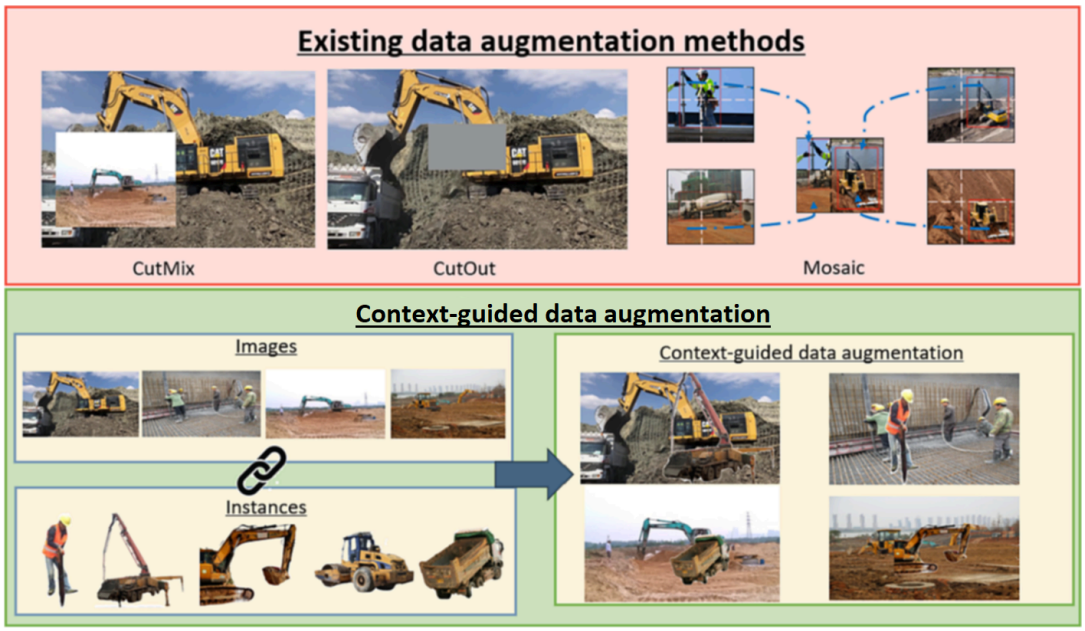


Figure 2.13: Example of the Context-Guided Data Augmentation technique from [19].

Figure 2.14: Example of the 'CutMix' Data Augmentation technique. Red colour indicates the bounding box and black colour represents the edges of the image. GT stands for Ground-Truth, DA stands for Data Augmentation.

ing improved generalisation. 'AutoAugment' introduces a sophisticated strategy that optimises augmentation policies relying on reinforcement learning, using a search algorithms to find optimal transformations [26]. 'Feature Augmentation' involves manipulating the feature space to enhance the model's resilience to variations [170]. Lastly, 'Neural Style Transfer' integrates artistic style transfer techniques to augment images, offering a creative and distinctive dimension to the data augmentation repertoire [79]. These categories collectively showcase the innovation and versatility in advancing data augmentation methodologies.

Image Mixing Data Augmentations represent a sophisticated category of techniques designed to enhance the diversity and generalisation capabilities of machine learning models. Notable methods within this category include Mixup [181] and CutMix [179]. Mixup involves blending two or more images by taking a convex combination of their pixel values, creating novel hybrid samples that lie on the line segment connecting the original images. This not only introduces diversity to the training data but also encourages the model to learn more robust and generalised features. CutMix, on the other hand, extends this concept by cutting and pasting rectangular patches from different images, fostering spatially coherent variations. By seamlessly integrating information from multiple sources, Image Mixing Data Augmentations prove valuable in mitigating overfitting and improving the performance of models across diverse datasets.

In [181], the Mixup technique generates augmented training examples through a linear combination of raw input vectors and their corresponding one-hot label encodings. Specifically, it constructs virtual training examples

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j, \tag{2.5}$$

where x_i and x_j are raw input vectors, and

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j, \tag{2.6}$$

where y_i and y_j are one-hot label encodings. The pairs (x_i, y_i) and (x_j, y_j) are randomly selected examples from the training data, and λ is a parameter within the range $[0, 1]$. Essentially, Mixup broadens the training distribution by incorporating the knowledge that linear interpolations of feature vectors should correspond to linear interpolations of their associated targets. Importantly, Mixup can be implemented with minimal computational overhead, requiring only a few lines of code.

Kim et al. have introduced a novel data augmentation technique known as Local Augment (LA) [84], aimed at profoundly altering the local bias property to produce significantly diverse augmented images, thereby enhancing the overall augmentation effect on neural networks. The methodology involves the selection of specific local patches within an image, followed by the application of distinct augmentation strategies to each of these patches. This targeted augmentation process intentionally disrupts the global structure of the object while concurrently creating locally diversified samples. This approach proves beneficial for the network as it facilitates learning the local bias property in a more generalised manner. Consequently, the Local Augment technique contributes to increased generalisability and enhanced prediction accuracy of the neural network [84]. A visual example of Image Mixing Data Augmentations is presented in figure 2.14.

More recently, there have been more developments on the scene of Image Mixing Data Augmentation. The authors of [96] introduced a novel mixup method called MiAMix, stands for Multi-stage Augmentated Mixup. A novel sampling method of the mixing ratio that is designed for multiple mixing mask. The MiAMix technique consists out of four stages: random sample pairing, sampling of mixing methods and ratios, the generation and augmentation of mixing masks, and the mixed sample output. The random sample pairing has two main difference when compared to a conventional method of mix pair sampling where the sample indices are shuffled and paired. The first difference according to [96] is the preparation two sets of random augmentation results for mixing. Secondly, they have introduced a new probability parameter that allows to generate so-called "corrupted" outputs [96]. In the next stage, sampling of mixing methods and ratios, MiAMix picks a method to generate a mask from a pool of methods. The methods are AGMix [96], MixUp [181], CutMix [179], GridMix, and FMix. Furthemore, this stage sample a parameter to set the mixing ratio for each mask. In the following stage, i.e., the generation and augmentation of mixing mask, the mask are augmented using basic Data Augmentation techniques such as rotation, shearing, and smoothing. In the final stage, the mixed sample output, the mask could be mixed either by point-wise multiplication or by summing the weighted mask, example of the technique

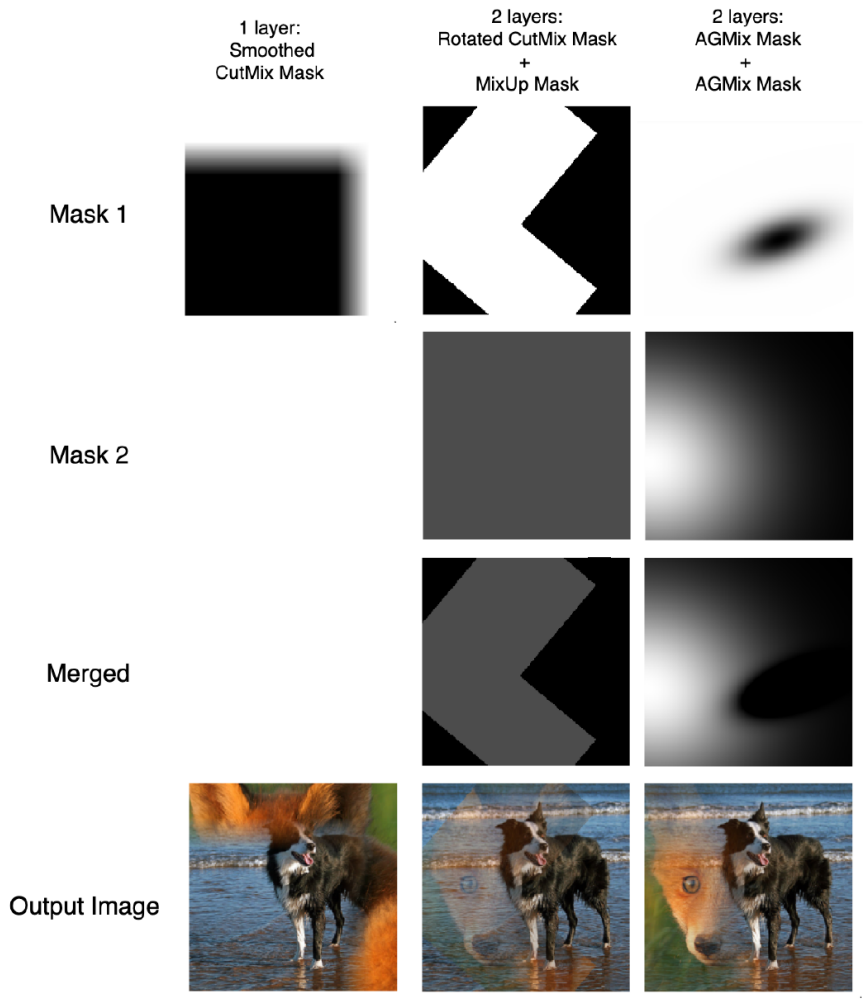


Figure 2.15: Example of the MiAMix Data Augmentation technique from [96].

could be seen in figure 2.15.

In the context of FSL, according to [173] the utilisation of limited data presents a challenge due to the potential for bias, referred to, by authors, as spurious correlation, especially when the data is subject to manual selection, as it causes a Distributional Shift. The small amount of data can lead to biases, referred to as spurious correlations. These biases arise due to the potential mismatch between the limited training data and the actual data distributions, particularly when data is manually selected. This results in the model relying on non-causal features that do not contribute to accurate predictions, alongside causal features that do. To

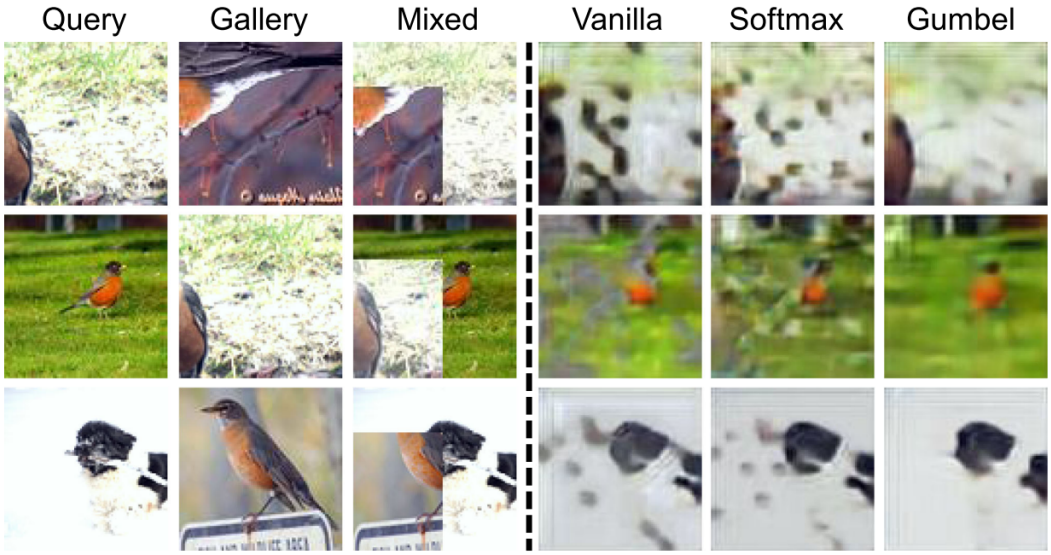


Figure 2.16: Example of the PatchMix Data Augmentation technique. Example is from [173].

address this issue, a novel Data Augmentation technique, named PatchMix, has been introduced [173]. This technique discerns between causal and non-causal features, promoting model invariance and generalisability. PatchMix achieves this by augmenting original query images with images from another query, wherein the class from the latter is incorporated into the original query image, preserving the label of the additional query differing itself from CutMix [179] by using hard-labels instead of soft-labels. The authors assert that this approach facilitates the disentanglement of causal and non-causal features, enhancing the model's ability to generalise effectively. An example of the PatchMix Data Augmentation technique could be observed in figure 2.16.

AutoAugment is a cutting-edge methodology designed for the automated improvement of data augmentation policies [26]. The approach involves a systematically constructed search space, within which each policy is composed of multiple sub-policies, as represented in figure 2.17. Notably, during training, one sub-policy is randomly selected for each image in every mini-batch. These sub-policies encapsulate two image processing functions, such as translation, rotation, or shearing, and are defined by specified probabilities and magnitudes.

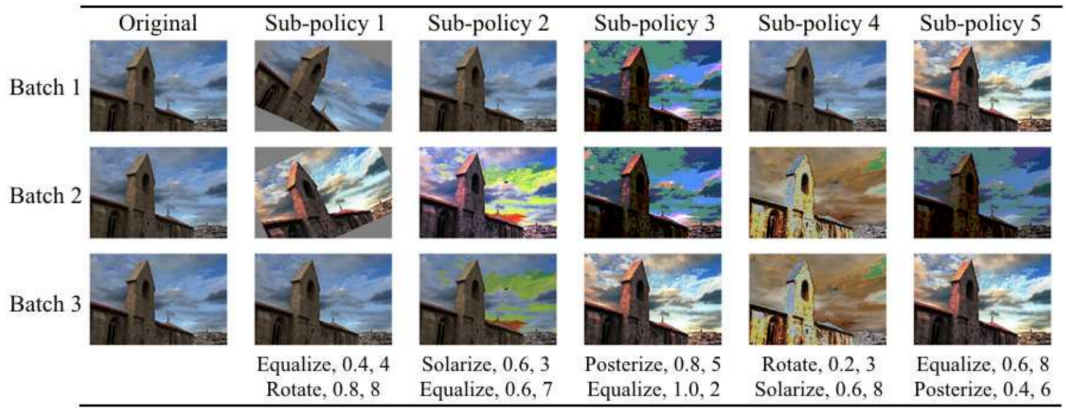


Figure 2.17: Example of the 'AutoAugment' Data Augmentation technique. Red colour signifies the bounding box and black colour signifies the edges of the image. GT stands for Ground-Truth, DA stands for Data Augmentation. A visual overview of the sub-policies from ImageNet using AutoAugment, example is from [26].

The standout feature of AutoAugment lies in its implementation of a sophisticated search algorithm. This algorithm diligently identifies the most effective policy by maximising the neural network's validation accuracy on a predetermined target dataset by relying on Reinforcement Learning principles. By integrating a systematic and automated approach to refining data augmentation strategies, AutoAugment represents a noteworthy advancement in optimising neural network performance, ultimately promising heightened accuracy and improved generalisation capabilities.

Randaugment is a data augmentation technique that introduces a randomised yet controlled approach to augmenting training data for improved model generalisation and robustness [27]. Unlike traditional augmentation methods that apply fixed transformations to all training samples, Randaugment dynamically selects a set of augmentation operations from a predefined pool and applies them with random magnitudes to each individual image, but unlike AutoAugment the selection isn't based on Reinforcement Learning principles. This randomness introduces diversity within the augmentation process, enhancing the model's ability to handle variations in real-world data. Common operations within the augmentation pool include rotation, translation, scaling, shearing, and changes in brightness and con-

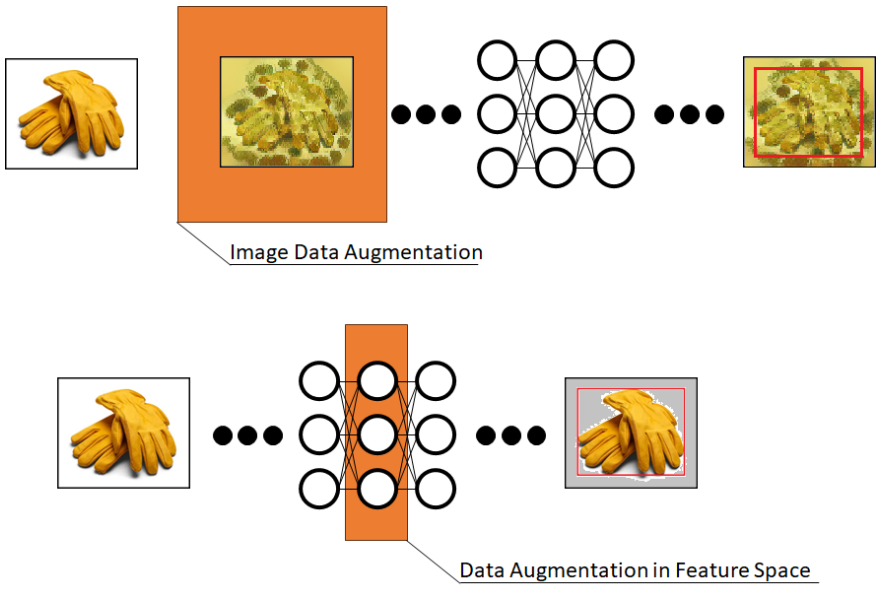


Figure 2.18: Example of the 'Feature' Data Augmentation technique. The augmentation happens in the Feature Space rather than on the image. Red colour indicates the bounding box and black colour represents the edges of the image. GT stands for Ground-Truth, DA stands for Data Augmentation.

trast. Randaugment provides a balance between injecting variability into the training set and maintaining consistency, contributing to more effective and adaptive learning for machine learning models.

Feature augmentation is a technique used in machine learning to enhance the learning capabilities of a model by manipulating the feature space. Unlike traditional data augmentation, which involves creating variations of the input data, feature augmentation focuses on altering the representation of features used by the model for training. This can involve techniques such as introducing new features, transforming existing ones, or applying mathematical operations to modify feature values. In [170], the authors delve into the advantages of incorporating feature augmentation techniques to enhance the training of machine learning classifiers. The investigation compares two distinct approaches: data warping, which generates supplementary samples through transformations applied in the data-space, and synthetic over-sampling, which introduces additional samples directly in feature-space. How-

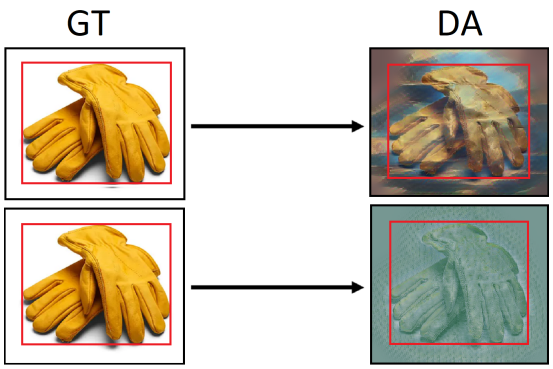


Figure 2.19: Example of the 'Style Transfer' Data Augmentation technique. Red colour indicates the bounding box and black colour represents the edges of the image. GT stands for Ground-Truth, DA stands for Data Augmentation.

ever, most of the research was done to improve the classification problem. Example image could be observed in figure 2.18.

A contemporary methodology is delineated in the work by Wang et al. (2023) [165]. This method criticises the characteristics inherent in the base dataset, the auxiliary support dataset, and the interrogative query dataset. By employing cosine similarity, it identifies the most pertinent prototypes from the base class and features from the query set. The proposed approach, denoted as Information Fusion Rectification (IFR), amalgamates two distinct types of information and rectifies the distribution of the support dataset. The cosine similarity metrics are employed as weights, aligning with their respective values, to optimally leverage the salient information embedded within both the base class data and the query set.

In [79], the authors propose a novel data augmentation approach that leverages random style transfer to enhance the resilience of CNN across both classification and regression tasks. In the context of training, style augmentation introduces randomness to texture, contrast, and colour, while preserving the underlying shape and semantic content of the data. The technique achieves this by adapting an arbitrary style transfer network to perform style randomisation, wherein target style embeddings are sampled from a multivariate normal distribution rather than being computed from a specific style image. Compared to traditional data augmentation techniques, style augmentation offers a unique approach by focusing on

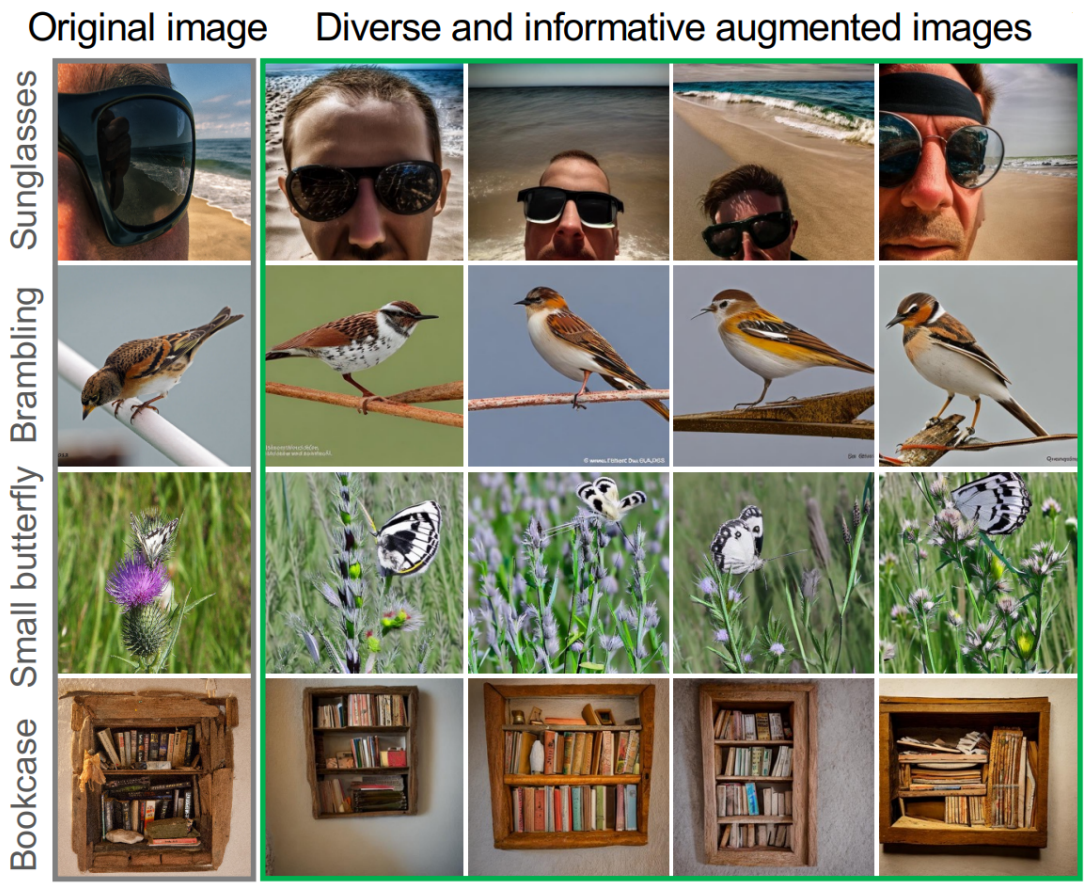


Figure 2.20: Example of the Diverse Data with Diffusions Data Augmentation technique. Example is from [47].

style-based variations. The findings suggest that style augmentation can be readily combined with these traditional techniques to improve network performance [79], [73]. An exemplary images are provided in figure 2.19.

Following the authors in [79], style transfer involves deriving representations from a pre-trained loss network, typically VGG [147]. These representations, obtained by passing images through the network, quantify style and content losses in relation to target style and content images. The process combines these losses into a joint objective function. Formally, the content and style losses are expressed as:

$$L_c = \sum_{i \in C} \frac{1}{n_i} \| f_i(x) - f_i(c) \|_F^2 \tag{2.7}$$

$$L_s = \sum_{i \in S} \frac{1}{n_i} \| G[f_i(x)] - G[f_i(s)] \|_F^2 \tag{2.8}$$

Here, c, s, and x represent the content, style, and restyled images. f is the loss network, $f_i(x)$ is the activation tensor of layer i after passing the x through f, n_i is the number of units in layer i, and C and S are sets containing the indices of the content and style layers. $G[f_i(x)]$ denotes the Gram matrix of layer i activations of f, and $\|\cdot\|_F$ represents the Frobenious norm. The overall objective is then expressed as:

$$\min_x L_c(x, c) + \lambda L_s(x, s) \tag{2.9}$$

Here, λ is a scalar hyperparameter determining the relative weights of style and content loss.

Another technique, that follows a similar approach, utilises diffusion models as described in [47]. First, this technique extracts latent features from a pretrained CLIP encoder [128], and then it uses Stable Diffusion-V2 [136] as the decoder to generate augmented images. Additionally, the result is filtered using cosine similarity to balance data diversity. Unlike previous methods, this approach does image data augmentation as well as feature data augmentation. An example could be observed in figure 2.20.

Finally, recent advancements in Large Language Model (LLM) architectures [3] have significantly influenced the field of computer vision, particularly in the domains of object detection and scene analysis. Innovations such as the Vision Transformer (ViT) and the adaptation of transformer-based methodologies to visual tasks have redefined the process of spatial feature extraction by enabling models to capture long-range dependencies and global contextual relationships with enhanced precision. These architectures demonstrate exceptional

efficacy in modelling complex interdependencies within scenes, thereby improving accuracy and robustness in object detection and scene analysis, even in intricate or real-world environments. By employing self-attention mechanisms and hierarchical representations, transformers have driven substantial progress in applications such as autonomous systems, intelligent surveillance, and Augmented Reality (AR). Framing this work within the context of these transformative developments situates it within the broader landscape of cutting-edge research, emphasising its relevance and potential to contribute to the advancement of emerging fields in visual systems.

2.5 Benchmark Datasets and Challenges

In the domain of computer vision and object detection, benchmark datasets play a pivotal role in evaluating the performance of algorithms and driving advancements in scene analysis for smart devices and augmented reality. These datasets provide standardised benchmarks against which algorithms can be tested, enabling researchers to assess their accuracy, robustness, and efficiency across diverse real-world scenarios. In the context of AR, generally speaking, the data collected for RGB cameras is applicable for AR scene analysis as the sensors used for AR are the same. However, the proximity of the working environment in AR is usually much closer than, for example, of the surveillance camera, therefore it should bare this in mind. In this section, we delve into some of the most common benchmark datasets and the challenges they present for scene analysis in the context of smart devices and AR applications.

Challenges represent the diverse set of obstacles and complexities encountered in real-world scenarios, ranging from occlusions and variations in illumination to object scale and viewpoint changes. Various cameras and sensors equipped on AR glasses are required to perform in numerous kinds of environments be it indoor or outdoor hence objects could appear in multitude of transformations, i.e., different distance from AR glasses, different light conditions, or, objects could be obscured by hands or other objects. Therefore addressing this

Table 2.1: Common Benchmark Datasets Comparison, where 2D - is to indicate whether the dataset provides 2D axis-aligned bounding boxes, OBB - is to indicate whether the dataset provides oriented bounding boxes, 3D - is to indicate whether the dataset provides 3D bounding boxes.

Common Benchmark Datasets Comparison					
Name	**Number of images**	**Number of categories**	**2D**	**OBB**	**3D**
COCO	200,000	80	✓		
PASCAL VOC	13,500	20	✓		
ImageNet	1,200,000	1,000	✓		
KITTI	15,036	8	✓		✓
VisDrone	169,636	4	✓	✓	

challenges is important. However, these challenges pose significant hurdles for computer vision algorithms, requiring robust solutions that can accurately interpret and understand visual data across diverse environments [100], [45], [139]. Addressing these challenges is essential for advancing the capabilities of computer vision systems, enabling them to perform effectively in practical applications such as autonomous driving, surveillance, and augmented reality.

Benchmarks, on the other hand, serve as standardised evaluation platforms for assessing the performance of computer vision algorithms. These benchmarks typically consist of curated datasets containing annotated images or videos, along with predefined evaluation metrics and protocols. By providing a common ground for evaluating algorithm performance, benchmarks enable fair comparisons between different methodologies and approaches. Moreover, benchmarks facilitate the development of innovative solutions by identifying key areas for improvement and driving advancements in algorithmic accuracy, efficiency, and scalability. Common benchmarks include datasets such as COCO (Common Objects in Context), ImageNet, and PASCAL VOC (Visual Object Classes), which are widely used for evaluating object detection, image classification, and scene understanding algorithms.

Common Objects in Context or COCO stands out as one of the most prevalent benchmark datasets for object detection and scene understanding tasks [100], [133]. It comprises over 200,000 images across various environments, annotated with object instances belonging to

Figure 2.21: Examples of images from COCO dataset representing four arbitrary chosen categories from left to right: car, truck, airplane, person.

Figure 2.22: Examples of images from PASCAL VOC dataset representing four arbitrary chosen categories from left to right: airplane, computer, dog, monitor.

Figure 2.23: Examples of images from ImageNet dataset representing four arbitrary chosen categories from left to right: truck, bowl, car, golden fish.

80 common object categories, examples of images could be observed in the figure 2.21. The annotations include precise bounding boxes, segmentation masks, and keypoint information, providing rich and detailed ground truth data for evaluating the performance of object detection algorithms in complex scenes with multiple objects and contextual information.

The PASCAL VOC dataset remains a cornerstone in the field of computer vision, offering a standardised benchmark for object detection, classification, and segmentation tasks [45]. It features a diverse collection of images spanning 20 object categories, each annotated with bounding boxes and class labels, see examples of image from PASCAL VOC dataset in figure 2.22. The dataset's comprehensive annotations and evaluation protocols enable fair comparisons between different algorithms and methodologies, driving advancements in scene analysis for smart devices and AR.

Figure 2.24: Examples of images from KITTI dataset representing the four arbitrary chosen categories: car, truck, person, tram.

ImageNet serves as one of the largest and most diverse image datasets, containing millions of labelled images across thousands of object categories [139]. Examples of images from ImagetNet dataset could be observed in the figure 2.23. While ImageNet is primarily utilised for image classification tasks, it also serves as a valuable resource for pre-training deep learning models for object detection and scene analysis. The dataset's vast scale and diversity enable researchers to train more robust and generalisable models capable of handling a wide range of object categories and variations in appearance.

The KITTI dataset [54], a benchmark in the field of computer vision, stands as a vital resource for research and development in autonomous driving and scene understanding. Comprising a comprehensive collection of high-resolution images, along with associated lidar point clouds, camera calibrations, and ground truth annotations, KITTI offers a rich and diverse dataset for various tasks such as object detection, tracking, segmentation, and depth estimation, examples of images are presented in figure 2.24. Furthermore, the availability of stereoscopic data could be applied for AR scene analysis in the autonomous driving domain. Captured in real-world urban and rural driving scenarios, the dataset encompasses a wide range of challenging conditions, including occlusions, varying lighting conditions, and diverse traffic patterns. Its meticulously annotated ground truth labels provide invaluable information for training and evaluating algorithms, enabling researchers to develop robust and accurate solutions for tasks critical to autonomous driving systems. From detecting and classifying objects such as cars, pedestrians, and cyclists to estimating accurate depth maps

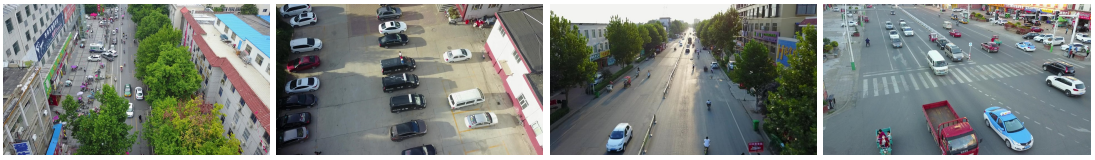


Figure 2.25: Examples of images from VisDrone dataset representing four main arbitrary chosen categories: pedestrian, car, van, truck.

and understanding complex traffic scenes, the KITTI dataset serves as a comprehensive testbed for pushing the boundaries of computer vision algorithms in real-world settings. Its widespread adoption and continued expansion underscore its significance as a benchmark dataset, driving innovation and progress in the field of autonomous driving and beyond.

VisDrone is a prominent benchmark dataset in the field of computer vision, specifically tailored for visual understanding tasks related to unmanned aerial vehicles (UAVs) or drones, demonstrated in figure 2.25. The VisDrone dataset stands out due to its comprehensive coverage of various real-world scenarios and challenges encountered in aerial imagery analysis. It comprises high-resolution images and videos captured by drones across diverse environmental conditions, including urban areas, highways, and natural landscapes. The dataset encompasses a wide range of annotated attributes, such as object detection, tracking, counting, and behaviour analysis, making it suitable for evaluating a broad spectrum of computer vision algorithms.

Comparing the datasets in table 2.1, all of the datasets provide information about axis-aligned 2D bounding boxes, but not all of them provide OBB and 3D bounding boxes. It is worth mentioning that ImageNet doesn't provide bounding boxes for all of the images as it often used for pretraining purposes. Another large dataset similar to ImageNet is COCO dataset. This dataset provides bounding boxes and segmentation mask for all of the images in the dataset, and it is a common choice for pretraining models because the the image data is considered diverse enough to let allow well-generalised AI models. One of the key strengths of the VisDrone dataset is its focus on addressing the unique challenges posed by aerial imagery analysis. These challenges include varying scales and viewpoints of objects, motion

blur, occlusions, and changes in illumination due to changing environmental conditions. By providing annotated ground truth data for these challenges, VisDrone enables researchers to develop and evaluate algorithms specifically tailored for aerial imagery analysis, with applications spanning surveillance, disaster response, urban planning, and environmental monitoring. Moreover, VisDrone serves as a valuable resource for advancing the field of autonomous drone navigation and interaction, facilitating the development of intelligent drones capable of autonomously navigating and interacting with their surroundings in real-time.

Benchmark datasets serve as invaluable resources for evaluating and benchmarking scene analysis algorithms for smart devices and augmented reality applications. By providing standardised benchmarks and evaluation protocols, these datasets enable researchers to assess the performance of algorithms across diverse real-world scenarios, driving advancements in accuracy, robustness, and efficiency. Moving forward, continued investment in benchmark datasets and the development of innovative methodologies are essential for addressing the challenges of scene analysis and unlocking the full potential of smart devices and AR technologies.

2.6 Evaluation Metrics for Object Detection

For completeness, this section defines the main key performance indicators (KPIs) used in the project: (i) The confusion matrix, (ii) the mean average precision, and (iii) the processing time. The KPIs selected for this work are considered standard evaluation metrics for the object detection task [119]. The mean average precision combines precision and recall metrics thus allowing us to focus on a single score of mAP. The processing time is critical for measuring real-time performance of an algorithm considering AR applications. Although, the confusion matrix is usually used in the classification task, it is also widely used in the evaluation of the object detection task. It provides a detailed visualised insight into the performance of a model.

The Confusion Matrix allows visualisation of object classification performance according to a user's set criteria (here Intersection over Union value). It has the following properties:

1. The horizontal rows represent the ground truth classes
2. The vertical columns represent the predicted classes
3. In addition, the final row and column correspond to "false negative" and "false positive", in order to indicate an undetected class, or a detected class that was not in the ground truth, respectively.

This third property is necessary as otherwise there would be an issue whenever a predicted bounding box measures an Intersection over Union (IoU) below the set threshold. If one only considers the label, it is considered as a True Positive (TP), however, since the overlap is insufficient, the prediction has to be assessed as a False Positive (FP) [122]. Confusion matrices can be computed for individual images as well as a batch of images or a video. Furthermore, confusion matrices can be represented according to the hierarchical class structure of object recognition and object identification when relevant. The perfect result, i.e., all objects are detected with their correct label, would result in a confusion matrix that has only entries on its main diagonal (all numbers would add up to the amount of ground truth instances). Results that are off the diagonal indicate predictions that have sufficient overlaps with existing ground truth instances but were labelled wrongly. The last column and row show if objects were completely missed or predicted in "thin air", respectively. Yet, the low number of entries in the last column suggests that almost all objects were detected, i.e., the recall value is good (figure 2.26).

The mean Average Precision (mAP) was first introduced in [97] as a way of representing object detection performance according to a user's set criteria. This metric is now widely applied, e.g., [[99], [131]]. A key element to understanding the average precision is the precision-recall graph as exemplary shown in figure 2.27. To calculate the average preci-

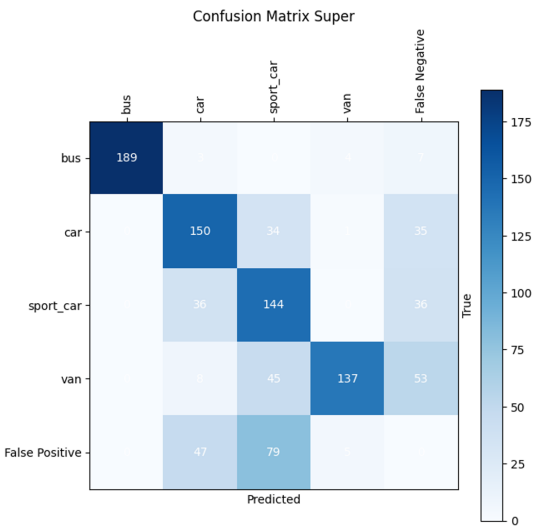


Figure 2.26: An example of a confusion matrix with random values for demonstration purpose for the object detection task.

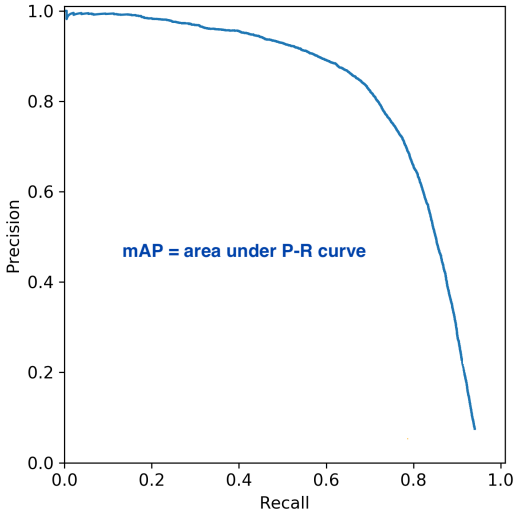


Figure 2.27: An example diagram of a precision-recall graph for the object detection task. The area under the precision-recall graph is the visualisation of mean average precision.

sion values, all the predictions for a specific class for all the images are collected and sorted by their confidence level. The average precision is defined as an approximation of the area under the precision-recall curve, where first the so-called envelope of the precision-recall curve is computed.

In addition to averaging over all the relevant classes, [97] suggest the mAP to be averaged over different IoU thresholds α as well. Doing so aims at giving some insight into how precise the predicted boxes are matching the ground truth annotation. As this is highly dependent on the quality of the annotation itself, this topic is discussed controversially in the literature [139], [106]. Comparisons with state-of-the-art methods rely on the mAP, a standard metric introduced in 2014 to quantify object detection performance based on a user-defined set of criteria [100]. It is defined as the mean value of the average precision of the individual classes:

$$mAP = \frac{1}{n} \sum_{k=1}^{n} AP_k \tag{2.10}$$

where AP_k is Average Precision of class k, and n is the number of classes.

Processing time is measured with respect to the average inference time per image. Consequently, it does not include the time required to load the architecture and/or weights. In order to achieve a representative value, the processing time is measured over the entire batch of testing data and then divided by the number of images that were evaluated. This approach achieves an averaging effect from different numbers of instances per image and different input sizes. In the literature both the absolute processing time per frame (in milliseconds) as well as the average number of frames per second are common units to express this KPI [167].

Chapter 3

A Modular Deep Learning Framework for Scene Analysis in Augmented Reality Applications

Contents

3.1 Introduction

AR applications leverage cutting-edge technologies to empower users with immersive experiences, facilitating dynamic interactions with their surrounding environment. By seamlessly superimposing digital visuals onto the physical world through the live camera view of a device, these applications transcend traditional boundaries, bridging the gap between the virtual and real realms. This transformative fusion of the physical and digital realms not

only enhances user engagement but also opens up unprecedented opportunities for novel forms of entertainment, education, navigation, collaborative experiences, and remote maintenance. AR applications act as enablers, reshaping the way users perceive and engage with their surroundings, ushering in a new era where the boundaries between the tangible and the virtual blur harmoniously to redefine the nature of human-computer interactions.

The overarching objective is to elevate the richness of the physical world by seamlessly integrating virtual information, encompassing textual data, images, videos, and intricate 3D models, into the real-time scenes captured through a camera [154]. This ambitious goal aligns with the core essence of AR, wherein the synergy of the tangible and the virtual realms creates an immersive tapestry of experiences. Through the judicious amalgamation of diverse digital elements with the live feed from a camera, AR aims to go beyond traditional limits of perception, providing users with an elevated and enhanced view of their environment. This symbiotic blend of virtual overlays with real-world imagery not only cultivates a more profound understanding of the environment but also lays the foundation for innovative applications.

Moreover, the burgeoning progress in the capabilities of computer systems, coupled with the rapid evolution of high-speed communication infrastructures and cutting-edge computer vision technologies, has catalysed an unprecedented surge in the demand for human-digital interaction. This surge is prominently facilitated through the integration of MX headsets and the advent of innovative three-dimensional interactive displays. The confluence of these technological strides not only enhances the overall quality of user experiences but also propels the boundaries of immersive computing to new horizons.

The ascendancy of MX headsets, equipped with advanced sensors and sophisticated optics, orchestrates an intricate dance between the physical and virtual worlds. Users, adorned with these headsets, find themselves seamlessly immersed in environments where digital elements coalesce with real-world surroundings, creating a truly blended and interactive experience. Simultaneously, the emergence of novel three-dimensional interactive displays

adds an extra layer of dynamism to this narrative, offering users intuitive interfaces that respond to gestures and spatial cues, thereby enriching the human-computer interaction paradigm.

The swift evolution of AR technologies has catalysed their wide application across diverse fields such as restoration, education, archaeology, art, tourism, commerce, healthcare, and maintenance [172]. In the realm of restoration, AR technologies contribute to the meticulous preservation and revitalisation of cultural heritage sites and artifacts [124]. By overlaying digital reconstructions or historical information onto physical structures, AR aids in visualising and understanding the past, fostering a deeper appreciation for cultural heritage. In education, AR breathes new life into traditional learning experiences by superimposing educational content onto real-world scenarios [118]. This dynamic approach enhances student engagement, making complex concepts more tangible and accessible through interactive and immersive experiences. Archaeology benefits from AR's ability to digitally augment excavation sites, enabling archaeologists to visualise and interpret historical layers in real time. AR assists in reconstructing ancient landscapes, artifacts, and structures, providing valuable insights into civilisations long past [108]. Art embraces AR as a medium for interactive installations, where digital elements merge seamlessly with physical artworks, creating dynamic and evolving masterpieces [61]. This fusion of the virtual and the tangible introduces novel avenues for artistic expression and audience engagement.

In tourism, AR transforms the way people explore and experience new destinations [187]. Virtual guides, historical overlays, and contextual information enhance the tourist's understanding of landmarks and attractions, offering a more enriching travel experience. Commerce leverages AR to enhance customer engagement through virtual try-on experiences, interactive product demonstrations, and immersive shopping environments [174]. AR bridges the gap between online and offline retail, providing consumers with novel and engaging ways to interact with products. Healthcare integrates AR for medical training, surgical navigation, and patient education [80]. AR overlays medical information onto the real-world

view, aiding healthcare professionals in decision-making and improving patient outcomes. This multifaceted integration of AR into various domains signifies a paradigm shift, where the augmentation of reality becomes a versatile tool with the potential to revolutionise how we engage with information, environments, and each other. These immersive technologies rely on the analysis of the surrounding environment to extract context information. For instance, in the field of autonomous vehicles, scene analysis and understanding (e.g., vehicle detection, traffic signs and light recognition, and pedestrian detection) is a key component for decision-making tasks and end-to-end control [121] so that the augmented environment can be seamlessly visualised on the car display.

In the last decades, advances in computer vision have fostered the design and implementation of object recognition methods, increasing computational performance and lowering process time [190]. As a result, current AR technologies based on object detection use complex computer vision techniques to detect and track objects in the real world. Examples of such technologies include the You Only Look Once model [7], homomorphic filtering and Haar markers [58] and the Single Shot Detector [35]. The use of Convolutional Neural Networks and Deep Learning led to faster and more accurate detection processes [184]. However, there are instances where CNNs deliver poor performance, e.g., due to low resolution of the image taken by camera sensor or when the objects to recognise appear small on the image or far away. For example, during AR maintenance procedures the camera sensor might detect hazards earlier as the AR maintenance personnel approaches the equipment. Providing information about the hazards could help prevent accidents and allow the AR personnel more time to react. To conclude, AR scene analysis has a noticeable impact on scene understanding and the overall AR experience.

The aim of this chapter is to provide a novel integrated end-to-end solution that improves performance in such conditions by introducing SR mechanisms. Not only have Generative Adversarial Networks been used for new data generation and to study adversarial samples, but in the recent past they have also been investigated to perform SR tasks [22][63]. Inspired

by this, the proposed approach is based on a cascade of two connected networks. The first network is a super resolution network that takes as input transformed images. More specifically, a 3D representation is used where the z-axis represents the colour channel of the image. The second network is based on the YOLO series' architecture, which was designed to improve performance at a low computational cost. The key contributions of this chapter are: a) the end-to-end design and training of the two connected networks, allowing automatic minimisation of the SR reconstruction error and maximisation of the detection and classification accuracy with a single novel optimisation function; b) a complete comparative study under a variety of environmental conditions that are known to affect the overall performance of AR devices; and c) a new dataset composed of synthetic objects created under different conditions, which allows unbiased performance evaluation under different sensor and environmental parameters. The aforementioned solutions could be integrated into the AR applications as a remote cloud service for better scene understanding or, perhaps, as an offline solution.

The chapter is organised as follows: Section 3.1 introduces the problem and relevant technologies; Section 3.2 reviews the data and data generation process required for the methodologies; Section 3.3 covers the methodologies used in the comparison with the proposed architecture; Section 3.4 describes the proposed end-to-end architecture; Section 3.5 presents results obtained using both a real image dataset (VisDrone) and a novel synthetic image dataset against the baseline methodologies; and Section 3.6 draws the final conclusions.

3.2 Data & Data Generation

The methodologies covered in this work are trained on a labelled data, i.e. supervised learning [28]. In case of CNNs, the data is the images and the images are annotated with a class and a number of bounding boxes. The modern object detection training is dependant on massive volumes of data because commonly more data leads to better results. The training could be performed on a real data or on a synthetic data. The real data means data that con-

tains pictures of real world taken by a camera. The synthetic data means data that contains images artificially generated by a software. The artificially generated images try to represent, in the best way possible, the real world. The real data has better representation of the real world. However, it is not always guaranteed to have access to the real data due to data scarcity, data unavailability, or sometimes security.

Real data helps AR systems accurately interpret and interact with the physical environment. For example, in AR applications like navigation or maintenance, real data from cameras, sensors, or GPS enables precise object recognition, positioning, and context-awareness. It addresses challenges like ensuring correct object placement, detecting hazards, or providing accurate, real-time overlays on physical environments. Synthetic data is especially useful in training AR systems, particularly when real data is scarce or difficult to acquire. For example, AR applications can use synthetic data to simulate diverse environments, lighting conditions, or user interactions that the system may encounter, thus improving the robustness and accuracy of object recognition and environmental understanding.

3.2.1 Real Dataset

The real dataset utilised in this study is the VisDrone dataset [39], a publicly available resource comprising imagery collected from a diverse array of drones featuring various types and models, view figure 3.1. These images were captured under a wide spectrum of conditions, capturing variations in lighting, density, location, environment, and objects. Curated by the AISKYEYE team at the Lab of Machine Learning and Data Mining, Tianjin University, China, the VisDrone dataset consists of 8,599 images categorised into 6,471 for training, 548 for validation, and 1,580 for testing purposes. Each image is meticulously annotated, providing comprehensive details such as object bounding boxes, object categories, occlusion, truncation ratios, and more. [39] Stored in JPEG format, the VisDrone images exhibit resolutions ranging from 960x540 to 2000x1500 pixels, reflecting the diverse array of drone models and cameras utilised during image acquisition, in stark contrast to the Synthetic dataset.

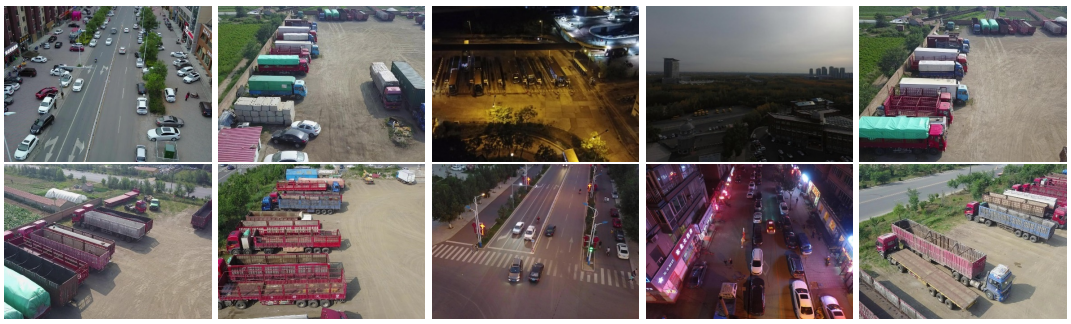


Figure 3.1: Sample images highlighting the variety of the VisDrone dataset.

The instances have been categorised into four main groups based on the size of their associated bounding box pixel areas. These categories include "very small," "small," "medium," and "large." Specifically, the "very small" category comprises bounding boxes with sizes smaller than 16x16 pixels, while the "small" category encompasses sizes ranging from 16x16 to 32x32 pixels. Bounding boxes falling within the range of 32x32 to 64x64 pixels are classified as "medium," while those larger than 64x64 pixels are designated as "large." The tables presented above (3.3, 3.2, and 3.1) provide a comprehensive breakdown of instances within each category for the train, validation, and test subsets.

The dataset images were carefully annotated with bounding boxes to precisely delineate the objects' positions. Furthermore, each object is accompanied by a label denoting its category or class, corresponding to the respective bounding box. In total, the dataset encompasses approximately 220,000 individual instances across the four main categories: 167,000 for training 3.3, 15,000 for validation 3.2, and 35,000 for testing 3.1. On average, each image contains 26 object instances, ranging from a minimum of 1 to a maximum of 267 instances per image. The VisDrone dataset encompasses 10 object classes, including pedestrian, person, car, van, bus, truck, motor, bicycle, awning-tricycle, and tricycle. However, for the purposes of this thesis, only the van, bus, truck, and car classes were pertinent to the experiments, which led to the exclusion of the other classes from the analyses.

Category	Total	Very Small	Small	Medium	Large
Bus	2565	140	449	934	1042
Car	25439	4859	8871	8302	3407
Truck	2311	188	512	743	868
Van	5365	667	2029	2015	654
Total	35680	5854	11861	11994	5971
% Total	100.0%	16.4%	33.2%	33.6%	16.7%

Table 3.1: Object composition in the VisDrone Dataset used for testing.

Category	Total	Very Small	Small	Medium	Large
Bus	237	37	73	74	53
Car	12850	2507	3717	4411	2215
Truck	684	86	179	226	193
Van	1810	311	535	655	309
Total	15581	2941	4504	5366	2770
% Total	100.0%	18.9%	28.9%	34.4%	17.8%

Table 3.2: Object composition in the VisDrone Dataset used for validation.

Category	Total	Very Small	Small	Medium	Large
Bus	5483	404	1104	1995	1980
Car	129409	22943	38344	40034	28088
Truck	11877	994	2739	4096	4048
Van	22867	2712	7002	7437	5716
Total	169636	27053	49189	53562	39832
% Total	100.0%	15.9%	29.0%	31.6%	23.5%

Table 3.3: Object composition in the VisDrone Dataset used for training.

3.2.2 Synthetic Dataset

Public datasets are often unbalanced as it could be observed in Tables 3.3, 3.2, and 3.1. The number of class instances is not uniform. In an attempt to address the issue, a synthetic dataset was generated. A software tool named the Data Generation Tool (DGT) was designed and developed. The software tool is capable of creating images of various objects of interest under user-defined requirements using a parametric approach. The parametric approach provides a way to balance the generated synthetic dataset by adjusting camera angles, distances from the camera, offsets from the camera, as well as offsets from the origin of the 3D world. Furthermore, the light conditions could be tuned to mimic any time of the day, like morning or late evening. Given the extensibility of the software, it could generate images under any

Figure 3.2: Data Generation Tool - Sample of Images from Synthetic Dataset. The first row demonstrates regular RGB images of objects. The middle row represents the objects' segmentation map, each colour belongs to a specific category. The bottom row illustrates depth maps of the objects, where the brighter the colour is the further away from camera the object is.

3D environment, including rare hazardous environments that are hard to collect in the real world or special types of camera sensors like night vision or heatmap. As there are many situations where images are affected by weather conditions, the software can also generate visual effects like fog and rain. Eventually, the designed tool that is capable of generating a well-balanced synthetic dataset that is able to capture many scenarios representing the real world.

The dataset generated using the DGT software was named Synthetic dataset. It contains images based on 32 models, each model correspond to a class, listed in table 3.4. Each model is rendered in a unique scene with a total number of five scenes. The rendering of each model and scene is affected by light parameters that simulate sun light as controlled by the user. Two gradations of light intensity representing day and night are used to generate the dataset. The data is captured in a variety of weather conditions such as rain, rain & wind, and/or fog. The models/objects are viewed from broad range of angles and distances. Furthermore, the images of the dataset are saved at a fixed resolution. Finally, some images are displayed using special visual effects like heatmap or night vision. Overall, the full dataset contains

Table 3.4: The following table lists all 32 categories that are present throughout the entire Synthetic dataset.

32 categories of the Synthetic dataset				
Ground	**Car**	**Sport car**	**Van**	**Bus**
	Audi	Dodge	Ambulance	City bus
	BMW	Ferrari	Kangoo	London bus
	Mini Cooper	McLaren	Combi	School bus
	Golf	Porsche	Minivan	Tourist bus
Air	**Helicopter**	**Jet**	**Aircraft**	**Small aircraft**
	AH-64D	B2 Spirit	A380	Breguet
	Huey	Euro fighter	B747	Cessna
	Ka-50	F-35	A757	Hawker
	Mi-24	Mig-29	Concord	Learjet

around 100 million, i.e., 99,532,800, images including the *real*, ground truth, depth, night, and thermal images. The *real* images in this case mean the images that simulate the real photos. The dataset averages to around 80 TB of uncompressed data. A sample of images from the dataset can be observed in figure 3.2.

The Synthetic dataset comprises five distinct scenes: city, forest, desert, grass, and empty scenes, as illustrated in figure 3.3. The desert and grass scenes are straightforward representations of an open sandy area and an open grass valley, respectively. Both the city and forest scenes incorporate 3D geometry, enhancing the realism of the environments. Conversely, the empty scene lacks any 3D geometry and features a plain black background.

The dataset generation process encompassed the manipulation of various environmental factors to simulate distinct times of the day. This was achieved by employing four different light intensities, each corresponding to a specific time of day. Additionally, the illumination angles were varied, with light provided from four different elevations and azimuths. Furthermore, environmental conditions were further altered by the incorporation of rain and wind parameters. Two scenarios were considered: one devoid of rain and another subjected to heavy rainfall. The heavy rain scene could include two levels of wind intensity and three different wind directions. Moreover, the dataset featured scenes enveloped in fog to introduce additional atmospheric effects. Notably, all scenes were generated at a resolution

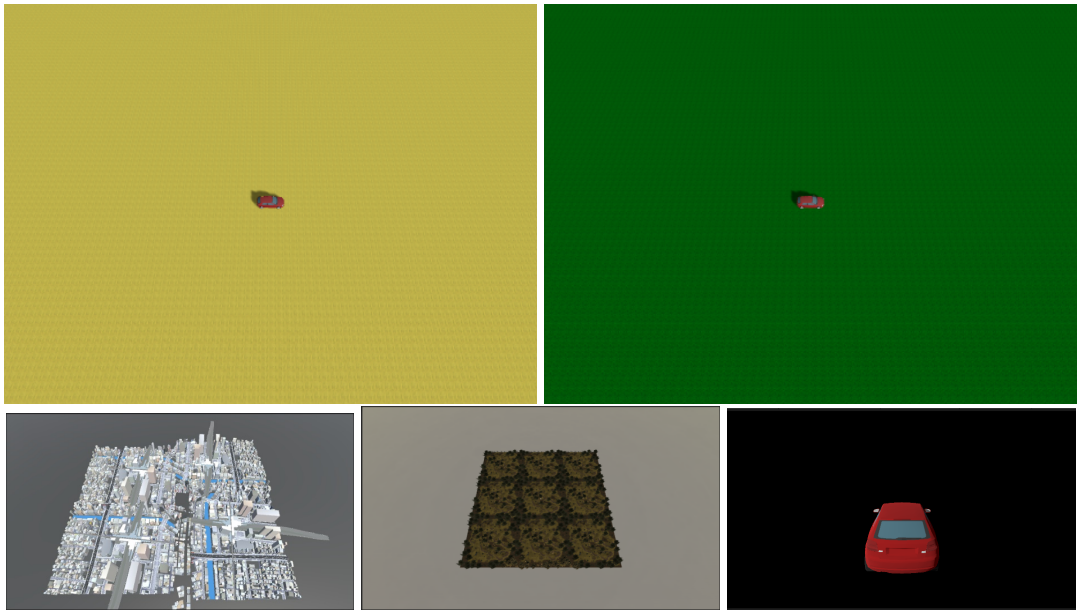


Figure 3.3: Data Generation Tool – Available Scenes: Desert, Grass, City, Forest, Empty. (top-left to bottom right)

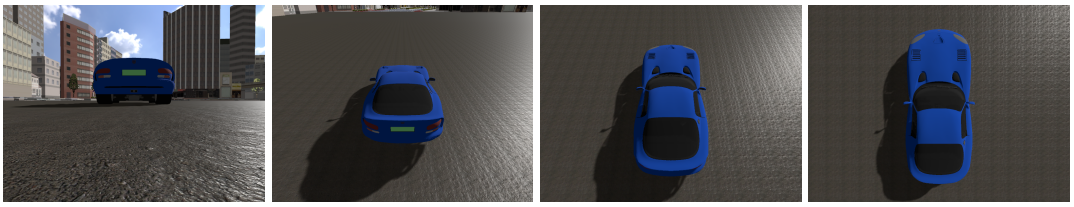


Figure 3.4: Data Generation Tool - Camera Elevations (0 - 90).

of 800x600 pixels, ensuring consistent visual fidelity across the dataset.

The generated dataset showcases objects from diverse viewpoints and orientations, accomplished through meticulous adjustment of camera settings. Specifically, the dataset was produced at four discrete distances, ensuring the camera consistently targeted the centre of each scene. By implementing the upper hemisphere model, where the midpoint of the hemisphere's base aligns with the scene's centre, and the hemisphere's radius is dictated by the distance to the camera, position of camera was tailored across four elevations and azimuths. For instance, as illustrated in figures 3.4 and 3.5, elevation values varied from 0 to 90 degrees in 30-degree increments, while azimuth values ranged from 0 to 270 degrees at intervals of 90 degrees, encompassing crucial viewing angles for comprehensive scene coverage.

Figure 3.5: Data Generation Tool - Camera Azimuths (0 - 360).

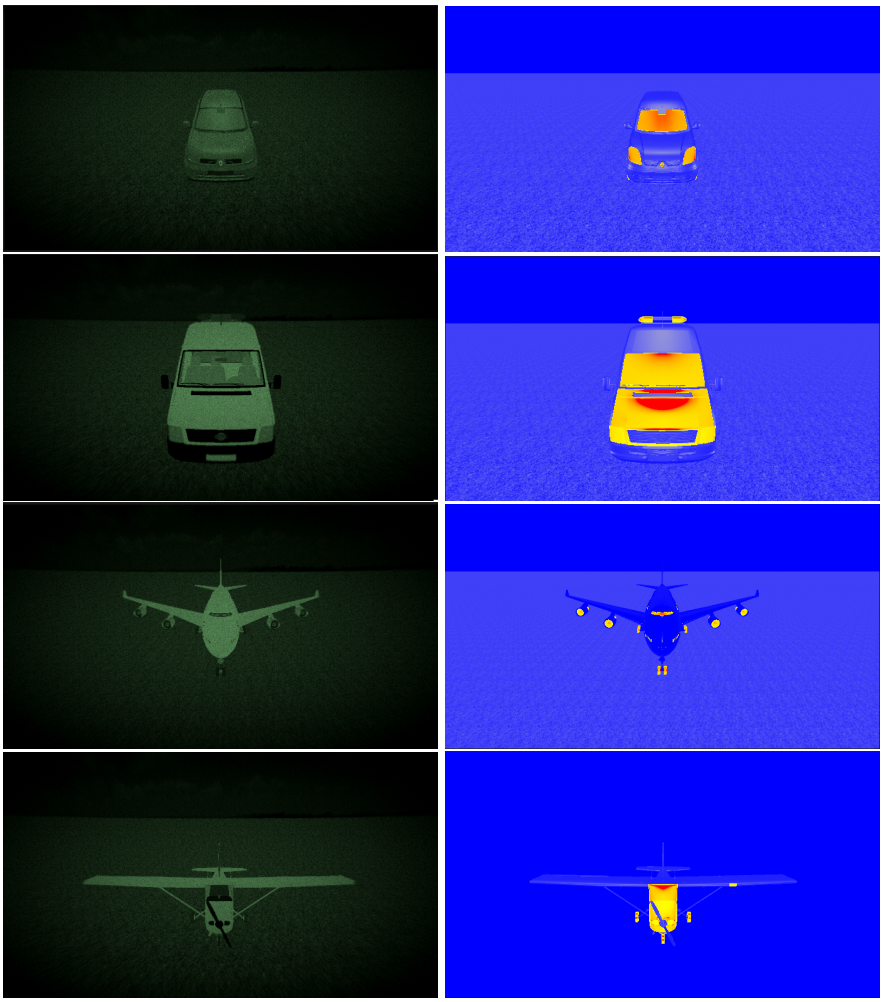


Figure 3.6: Data Generation Tool - Night Vision & Thermal Vision Examples.

The Synthetic dataset depicts objects observed through diverse camera sensors, including night vision goggles and heat-camera vision, as depicted in figure 3.6. This dataset was generated utilising both visual effects: night vision, enabling visibility in low-light conditions, and thermal vision, which generates a heatmap representation of the objects.

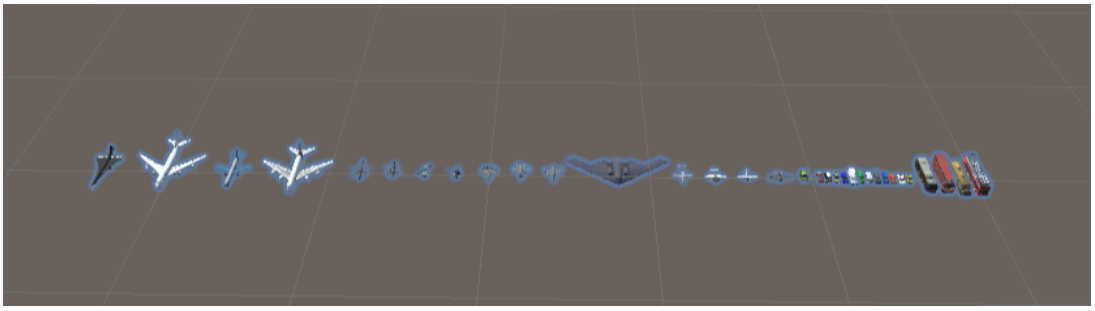


Figure 3.7: Data Generation Tool - All 32 Models.

The dataset encompasses 32 objects based on 32 models, illustrated in figure 3.7 and listed in table 3.4, generated across three distinct elevation and azimuth angles. Elevation values ranged from 0 to 45 degrees, while azimuth values spanned from 0 to 90 degrees. Each object was precisely positioned at the centre of the scene, with an offset from the centre set to zero. By selecting specific angles, positions, and offset parameters, the tool meticulously generates individual images, each featuring a single object, facilitating precise analysis and evaluation of object detection algorithms under varied environmental conditions.

3.3 Baseline Detectors

Authors of [144] describe Object Detection as a computer vision task comprised of the Classification and Localisation sub-tasks. The Classification sub-task involves the prediction of the label of an object in an image. The Localisation sub-task also involves predicting the bounding boxes of an object in an image. Subsequently, the Detection task combines both the prediction of the labels and the bounding boxes. However, in addition, the Detection task performs prediction of multiple objects in an image.

For the purpose of this thesis, to facilitate the evaluation and analysis of the proposed methods, this chapter will introduce the baseline detectors. These detectors represent the state-of-the-art methodologies established at the beginning of the research work, serving as the baseline against which other methodologies can be compared. The baseline detectors comprise FasteR R-CNN [134], RetinaNet [99], and YOLOv3 [133]. The aforementioned

methods were chosen because they represent the SOTA methods of the time when the work has started. FasteR R-CNN is the SOTA representative of a two-stage detector that was the closest to real-time performance as stated by authors in [134]. RetinaNet has introduced a novel loss function and demonstrated a balanced architecture between speed and accuracy achieving competitive results on COCO benchmarks [100]. YOLOv3 was a SOTA methodology that claimed to achieve real-time performance [133] , such as 15-30 fps, but at the expense of neglecting accuracy.

FasteR R-CNN, or Fast Region-based Convolution Network Network with Region Proposal Network, is an advancement over Fast R-CNN, originally developed by Girshick et al. in 2015 [56]. FasteR R-CNN utilises the convolutional feature maps employed by region-based methods like Fast R-CNN to construct a RPN, aiding in the regression of region bounds and objectness scores. FasteR R-CNN can be viewed as two distinct networks operating in tandem to predict bounding boxes and labels. It is considered a two-stage detector that first performs region proposal and then proceeds to object detection. The overall architecture of the FasteR R-CNN methodology is depicted in figure 3.8.

The RetinaNet methodology integrates the most successful components from single-stage object detection methods prevalent at the time. Notably, RetinaNet introduces a novel loss function known as Focal Loss. This innovation addresses the Class Imbalance problem encountered in detectors processing between 10^4 to 10^5 candidate locations per image, while only a few locations actually contain objects [99]. Focal Loss strategically down-weights easy examples with minor errors, ensuring that their contribution to the total loss remains minimal despite their large numbers. RetinaNet leverages the ResNet [70] with FPN [98] backbone, along with two sub-networks for classification and bounding box regression, as illustrated in figure 3.9. The ResNet with FPN backbone constitutes a transform-invariant residual network, incorporating skip-connections within residual blocks to preserve information across layers in deep neural networks [70]. The FPN serves as the "neck" of the model, handling the ResNet network's output and providing a multi-scale top-down pyramid to en-

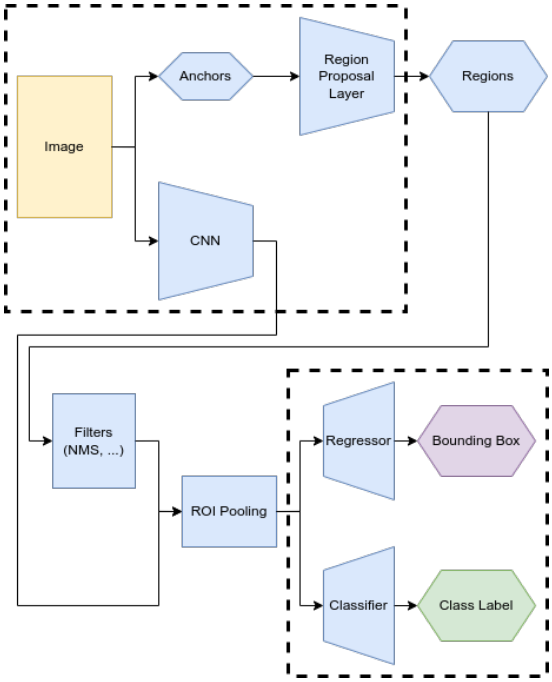


Figure 3.8: FasteR R-CNN Architecture.

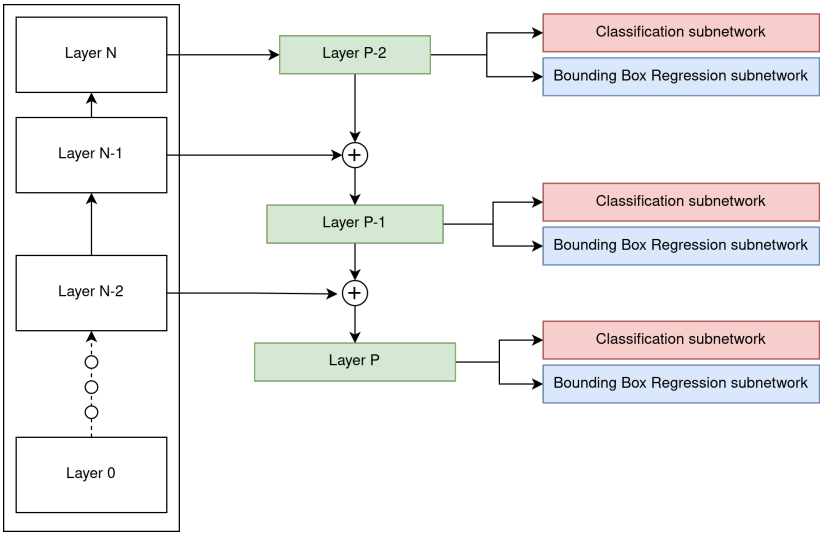


Figure 3.9: RetinaNet Architecture.

sure scale invariance. Each scale of the pyramid corresponds to a pair of sub-networks for classification and bounding box regression. RetinaNet employs an anchor-based approach, utilising a predefined set of anchor boxes at each scale of the pyramid network.

The YOLOv3 methodology, as outlined in [133], constitutes a single-stage detector engin-

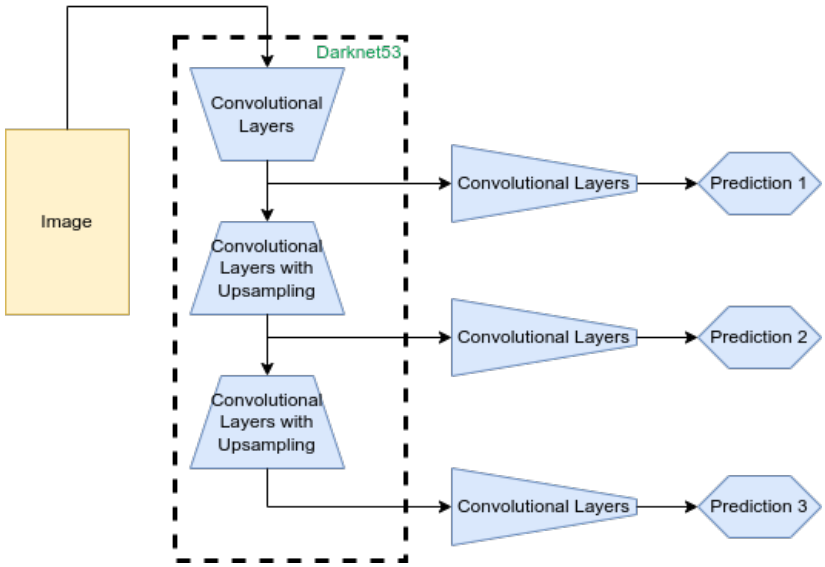


Figure 3.10: YOLOv3 Architecture.

eered to enhance detection speed with minimal sacrifice in accuracy. It represents a progression from YOLO9000 [132] and YOLO [131], embodying a distinctive approach by processing the entire image through partitioning it into a grid of dimensions $S \times S$. YOLOv3 operates on the feature map produced by the Darknet53 backbone, a backbone architecture akin to ResNet but comprising 53 layers. Notably, YOLOv3 predicts boxes at three different scales, as could be observed in figure 3.10, mirroring the functionality of FPN. The final output matrix adopts a shape of $N \times N \times [3 \cdot (4 + 1 + C)]$, where N represents the size of the final layer, C denotes the number of classes, and the numbers 4 and 1 represent the four bounding box offsets and one objectness score, respectively.

3.4 End-to-End Super Resolution Object Detection Method

In this chapter, we propose an end-to-end framework for scene understanding that combines super resolution, object detection, and classification architectures. Figure 3.11 shows an overview of the proposed methodology, where the two main components take as input an image (or a video) and are trained in an end-to-end manner. Details of these two processing blocks are described in the following subsections.

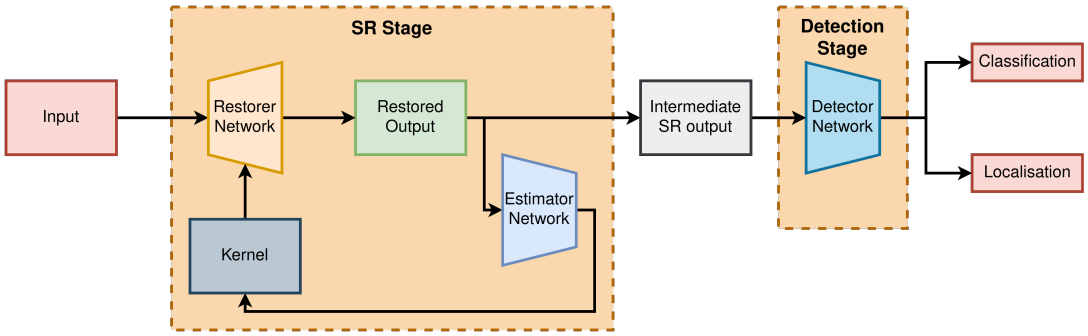


Figure 3.11: Overview of the proposed novel framework trained end-to-end. For the SR and detector models any state-of-the-art solutions can be used without affecting the overall pipeline and the proposed modular architecture.

3.4.1 Super Resolution Method

Super resolution involves generating high-resolution images (HR) from corresponding low-resolution (LR) ones. HR images offer superior reconstruction quality, benefiting various real-world applications like satellite and medical imaging [76], media content, face recognition [114], and security. Additionally, many researchers suggest its potential in enhancing computer vision tasks such as object detection and recognition [29], [67]. Various methods tackle SR, ranging from traditional prediction and edge-based approaches [82], [41] to modern deep learning techniques, which have achieved state-of-the-art performance. Despite advancements, SR remains a formidable challenge and an ongoing research pursuit. Incorporating super resolution into computer vision pipelines as a preprocessing step is common practice. Typically, SR models are trained unsupervised on diverse datasets, while target classification or detection models are trained solely on task-specific datasets. This approach infuses SR images with additional information not present in the target labelled dataset [75].

SR models are trained under the assumption that the low-scale images, provided as input, result from a low-pass filter application, such as Gaussian blur or a point spread function. The training process involves initially down-sampling high-resolution images using such a kernel and then optimising the model to reconstruct these high-resolution images. Ideally, the kernel function should align with the actual blurring process induced by the camera employed in the target application. However, as this information is often unknown, 'standard'

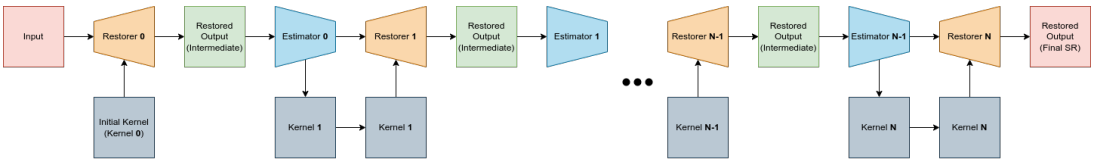


Figure 3.12: Training setup for the DAT SR deep network.

kernels are typically utilised. Regrettably, these standard kernels inadequately capture the specific optics and sensors of the actual cameras used to capture the images of interest, resulting in diminished performance in real-world scenarios. To mitigate this limitation, methods have been proposed to learn the blur kernel, referred to as Blind Super Resolution [110].

The usage of advanced methodologies, exemplified by the cutting-edge Deep Alternating Network (DAT) [75], prominently relies on deep learning architectures. The selection of DAT for our pipeline was motivated by its adoption of an unsupervised learning paradigm and its capacity to execute rapid computations, rendering it well-suited for deployment across mobile platforms and low-specification desktop systems. Indeed, empirical evidence provided by the authors indicates that DAT achieves an impressive average processing speed of 0.75 seconds per image, significantly outperforming its counterparts such as KernelGAN [13] + ZSSR [145], and IKC [60] by more than 500-fold and 5-fold, respectively. These notable processing speeds are regarded as expedient within the domain of SR. The schematic depiction in figure 3.12 elucidates the training architecture of DAT, comprising two principal networks: the Restorer and the Estimator. The Restorer is tasked with generating the SR image, while the Estimator furnishes an estimate of the blur kernel predicated on the restored image. These networks are iteratively engaged, with each Restorer-Estimator iteration aimed at augmenting the quality of the SR image and refining the precision of the estimated kernel. The optimisation of the Restorer-Estimator sequence is achieved via an end-to-end optimisation strategy utilising a stochastic back-propagation algorithm.

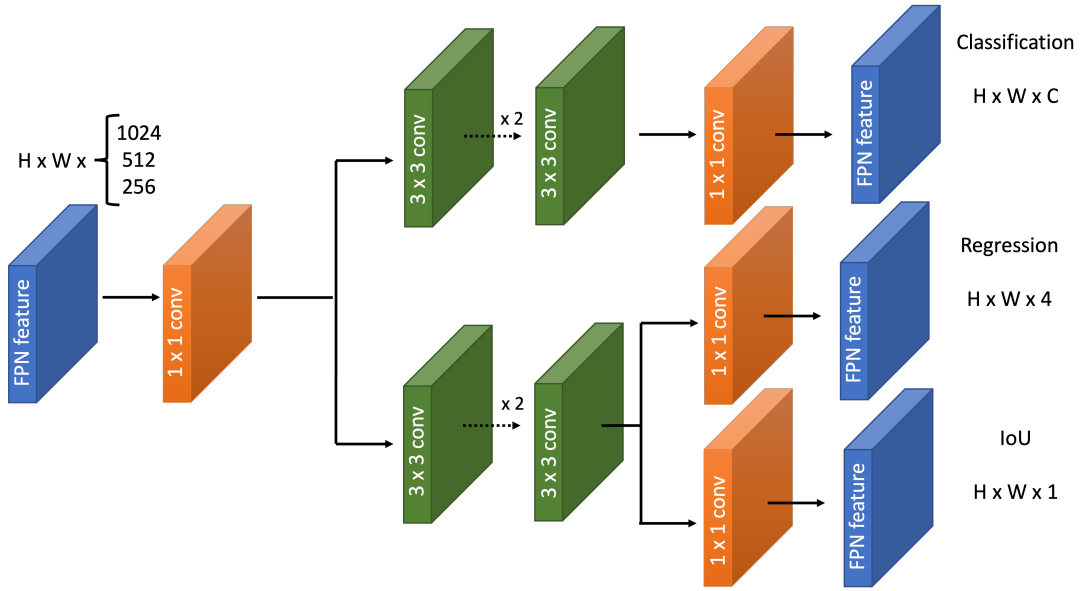


Figure 3.13: YOLOX architecture relying on a decoupled head

3.4.2 Object Detection Method

For augmented reality applications, object detection models are required to achieve high accuracy in real-time. In the proposed framework, the anchor-free model YOLOX [53] presents an optimal solution. YOLOX boasts a simple, powerful, and computationally efficient architecture, which is built upon one of the most widely used detectors in the industry, YOLOv3 [133]. YOLOv3 is renowned for its limited computational cost and excellent software support. However, a significant enhancement of YOLOX over previous architectures in the YOLO series is the employment of a decoupled head, which enhances convergence speed. Figure 3.13 provides an overview of the YOLOX architecture, which includes two parallel branches with $3x3$ convolutional layers following a $1x1$ convolutional layer aimed at reducing the number of channels. Additionally, YOLOX incorporates an Intersection over Union aware branch within the regression branch, distinguishing it from the baseline YOLOv3 model.

Another enhancement of YOLOX is, unlike the past versions of YOLO detectors (except for YOLOv1), the usage of an anchor-free model. Anchors are candidate bounding boxes with predefined dimensions that the detector selects during the detection process and for which it predicts the delta values for their centres and dimensions. Obviously, these additional predic-

tions require extra processing during both the training and inference stages, which impacts the overall computational time. On the other hand, when using an anchor-free approach, bounding boxes are predicted directly, which reduces the number of design parameters. As such an approach requires advanced data augmentation to match the performance of anchor-based models, state-of-the-art data augmentation approaches, i.e., Mosaic and MixUp, were exploited [53]. Indeed, they are known to bring stability and reduce overfitting during the training process. Finally, it is important to specify that YOLOX leverages a high-performance CNN front-end, CSPNet [162], which is followed by FPN [133].

3.4.3 End-to-end Framework

The methods described earlier were integrated into an end-to-end framework, [94]. Thus, the framework comprises two main components, i.e. SR and Detector. Equation (3.1) illustrates the proposed end-to-end architecture where x is the input low-resolution image, y is the image generated by the super resolution function $S(\cdot)$, and z is the output of the detection function $D(\cdot)$.

$$\begin{cases} \text{y=S(x)} \\ \text{z=D(y)} \end{cases} \rightarrow z = D(S(x)) \tag{3.1}$$

In this framework, an input image is first handled by the SR component, which produces a super-resolved output image. Then, this image is passed to the detector component, which recognises and locates objects. Through this process, the detector learns from images enhanced by the SR component. The input images are super-resolved using kernels. There are many types of kernels, such as the common bicubic kernel or linear kernel. These kernels are well-studied and don't require an AI network to calculate them.

However, in the case of the SR task, real-world images don't contain information about the kernel, making it challenging to successfully restore them. Consequently, an estimator is used to infer the kernel during the training process. This estimated kernel is then passed to

the restorer to generate images. As a result, the restored images contain features that are the product of the kernel. These features can be picked up by the detector during the training process, creating a symbiotic relationship between the SR and detector components, leading to improved performance.

To monitor and evaluate the training of the framework, several state-of-the-art loss functions were selected. The detector is trained using Varifocal Loss [182] as the classification loss function and SIoU [55] (Scylla Intersection over Union) as the box regression loss function. Moreover, the training process was facilitated with SimOTA, a simplification of OTA [52] (Optimal Transport Assignment), for dynamic label assignment [53].

The Varifocal Loss is particularly efficient because it considers both classification and localisation scores when ranking candidates using IoU. Similarly, the SIoU loss function addresses direction mismatch between expected and predicted bounding boxes by exploiting angle, distance, shape, and IoU costs.

Finally, the value of SimOTA is to view the task of bounding box assignment as an optimal transport problem, where the unit transportation cost between an anchor-point and ground truth is expressed as a weighted sum of their classification and regression losses to find the best assignment solution.

3.4.4 Parameters

The end-to-end framework was fine-tuned by running 10 epochs, as stated by the author in [70] the model should converge after several epochs, therefore an arbitrary small number based on the transfer learning practices was chosen [178], with a batch size of 3 on both real and synthetic data under four different categories. The selection of batch size as motivated by the hardware restrictions, specifically, the size of VRAM and number of GPUs. The learning rate was set to 0.0001 for the SR component, it was set to 0.0032 with SGD (Stochastic Gradient Descent) optimisation for the detector. For both, the parameters were chosen following the default parameters suggested by the community and authors of [75],

[53]. Additionally, as mentioned earlier, training was enhanced using Mosaic [14] and MixUp [181] as data augmentation strategies.

3.5 Results

3.5.1 Baseline Results

This subsection of the thesis presents the findings derived from the baseline detectors. The primary objective was to establish the baseline performance metrics. The datasets underwent testing across all three supported types of object detection and recognition models: FasteR-RCNN, RetinaNet, and YOLOv3. Testing was conducted with consistent confidence threshold and IoU parameters set to 0.3 and 0.5, respectively. The parameters are set to common values found in the works of authors in [56], [133], [99], [134], [131], [45]. The original dataset comprises diverse images captured under varying environmental, lighting, and angle conditions, featuring real photographs of different vehicle types at considerable distances. Each image contains multiple objects of distinct types positioned at various locations. This dataset served as the reference point for comparing against the results of subsequent experiments. Discrepancies in these outcomes were leveraged as a metric to assess performance enhancements or deteriorations.

The categories encompass the most prevalent vehicle types pertinent to this project. The third row in table 3.5 (Original Data) illustrates the performance metrics derived from the original dataset, i.e., before applying any SR processes, serving as the baseline for all subsequent experiments. The results indicate RetinaNet as the most successful model, achieving a weighted mAP of 48.15%. Notably, the experiment was executed twice, yielding consistent outcomes on both occasions. Overall, the findings suggest the potential of the SR component to enhance model performance, albeit necessitating additional training and fine-tuning for validation. The reason SR improves the performance could be due to a sharper edges of the features produced as the result of the SR process. Also, depending on the SR scale, in case of FasteR R-CNN, the size of an object in the image could match the expected size and

VisDrone Dataset	Training Samples: 1080		
	Testing Samples: 3456		
weighted mAPs	RetinaNet	YOLOv3	FasteR R-CNN
Original Data	48.15%	44.57%	40.40%
SR Data (original size)	47.67%	44.28%	**40.50%**
SR Data (scaled by 2)	44.86%	42.91%	39.93%
SR Data (original size, retrained with SR samples)	**52.61%**	**44.76%**	38.41%

Table 3.5: Results of experiments on VisDrone dataset using RetinaNet, YOLOv3, and FasteR R-CNN models.

thus lead to difference in propagation. Furthermore, if information regarding the anticipated range of object dimensions is available, adjustments to the parameters of the backbone network may enhance the efficacy of object detection and recognition tasks. Concerning FasteR R-CNN, being one of the earliest models in the domain of object detection and recognition, it exhibits heightened sensitivity to the expected object dimensions. The capability of the backbone network to process objects falling outside the anticipated range is comparatively limited, while its generalisation capacity for new feature variations, such as the number of identifying characteristics or object sizes, is less pronounced.

The proposed method has been applied to object recognition and scene understanding. Its evaluation was performed using the VisDrone dataset [188] and the synthetic dataset as defined in the section 3.2. In this evaluation process, using the VisDrone dataset, table 3.6 presents the performance in terms of mAP of the proposed framework compared to other approaches discussed in the literature review. Our framework surpasses all competing methods. Furthermore, the additional value of the super resolution component is clearly evident as it exceeds the mAP of YOLO by more than 20%. The confusion matrix depicted in figure 3.14 further illustrates the model's performance. Particularly, it demonstrates high accuracy in predicting objects belonging to the predominant "car" category. However, it is noteworthy that the "van" category is frequently misclassified as the "car" category, likely due to the visual resemblance between images of these two classes.

Further evaluation has been conducted using the Synthetic dataset that we created using

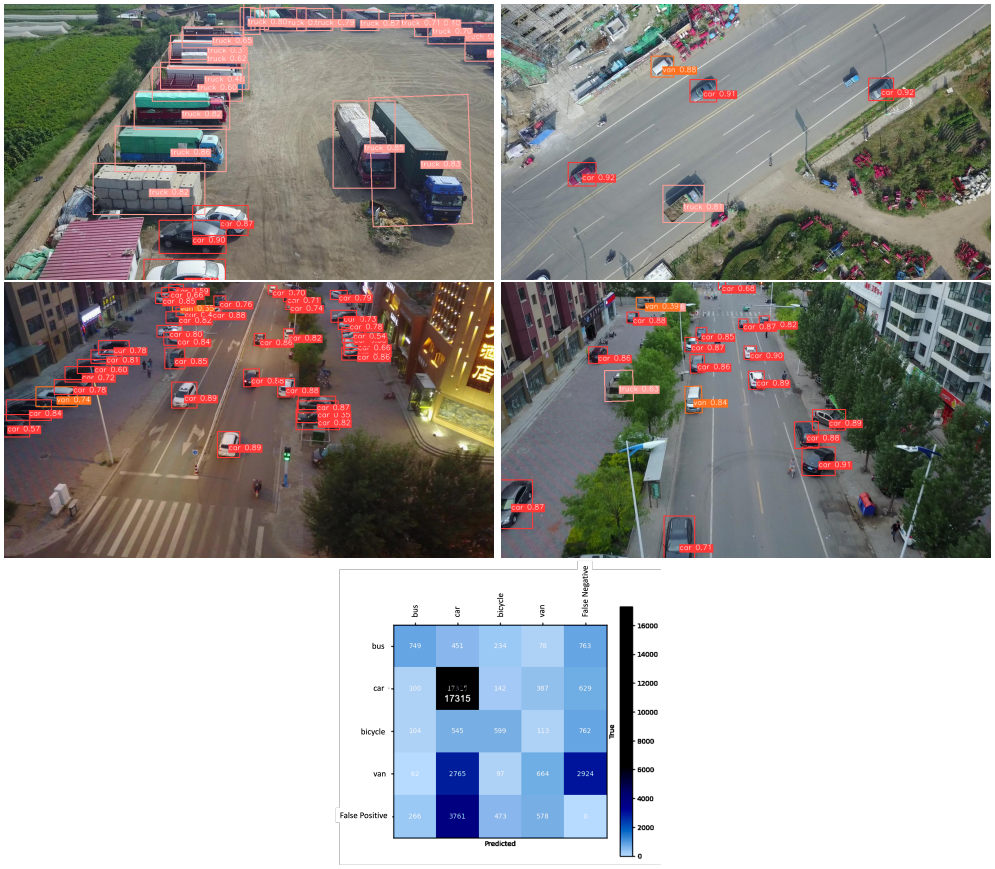
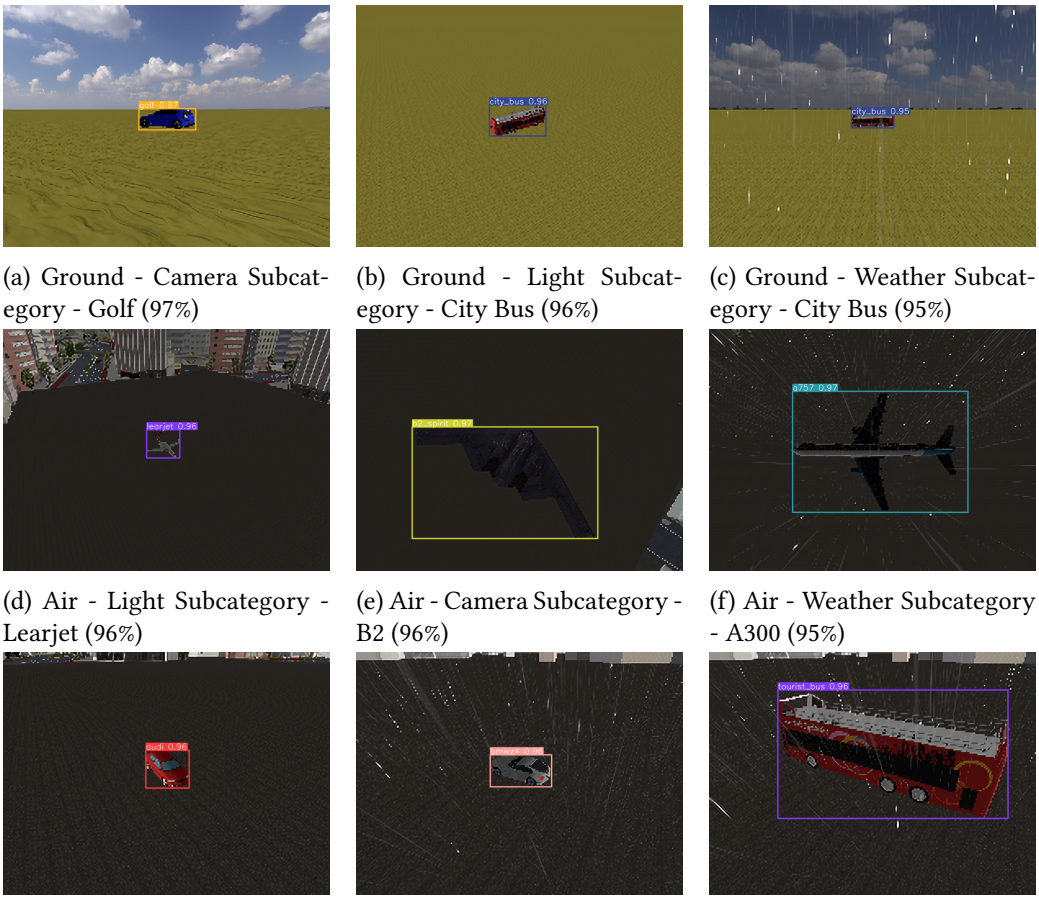


Figure 3.14: An example of predictions and confusion matrix (the white colour of the image was levelled up to see the numbers).

Table 3.6: Performance results of the proposed method in comparison with baseline methods on the VisDrone dataset.

RetinaNet	YOLOv3	FasteR R-CNN	YOLOX	Proposed
52.61%	44.76%	40.50%	46.99%	**67.09%**

Data Generation Tool as described in section 3.2. The performance of the proposed framework for these three categories in terms of mAP is shown in table 3.7. Additionally, the confusion matrices can be observed in figure 3.16. As could be observed, the results demonstrate high confidence in predicting the buses and cars categories. The sport cars and cars categories are quite often mistaken what is reasonable because both categories are very similar. The van category on the other hand sometimes has been mistaken with the sport cars category what indicates close similarity of the samples in the dataset. Whereas the bus category samples seem to be distinct enough to avoid confusion with the other categories. The



(a) Ground - Camera Subcategory - Golf (97%)

(b) Ground - Light Subcategory - City Bus (96%)

(c) Ground - Weather Subcategory - City Bus (95%)

(d) Air - Light Subcategory - Learjet (96%)

(e) Air - Camera Subcategory - B2 (96%)

(f) Air - Weather Subcategory - A300 (95%)

(g) Ground - Camera Subcategory - Golf (96%)

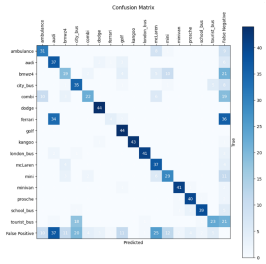
(h) Ground - Weather Subcategory - BMW (96%)

(i) Ground - Weather Subcategory - Tourist Bus (96%)

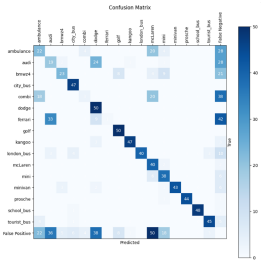
Figure 3.15: Examples of the generated synthetic data. The top and bottom rows represent examples from the Ground category. The middle row represents examples from Air category.

Table 3.7: Performance (mAP, %) of the End-to-End Super Resolution Object Detection framework on the Synthetic dataset in the three categories, where PM is the proposed framework, compared with the baseline models.

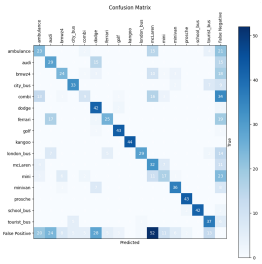
Category	Sub-category	FasteR R-CNN	YOLOv3	RetinaNet	YOLOX	PM
Air	Camera	35.24%	44.82%	44.79%	47.06%	**60.52%**
	Light	40.66%	63.58%	61.25%	66.76%	**81.25%**
	Weather	35.35%	39.00%	45.57%	40.95%	**66.98%**
Ground	Camera	37.95%	76.02%	78.81%	79.12%	**79.22%**
	Light	32.54%	38.52%	77.12%	40.45%	**78.21%**
	Weather	37.07%	66.32%	**76.09%**	69.64%	75.71%



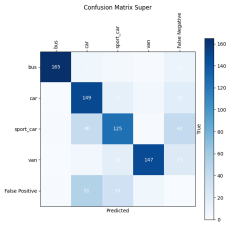
(a) Ground Category - Camera Subcategory - Confusion Matrix



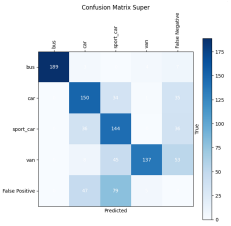
(b) Ground Category - Light Subcategory - Confusion Matrix



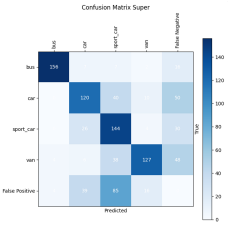
(c) Ground Category - Weather Subcategory - Confusion Matrix



(d) Ground Category - Camera Subcategory - Super Confusion Matrix



(e) Ground Category - Light Subcategory - Super Confusion Matrix



(f) Ground Category - Weather Subcategory - Super Confusion Matrix

Figure 3.16: Confusion Matrices of the Ground category of the generated synthetic data. From the left, the first column shows the Camera sub-category, the second column shows the Light sub-category, the third columns displays the Weather sub-category.

overall results of the proposed method exceed the baseline demonstrating an improvement in most of the categories due to novel SR techniques along with the more robust and lightweight object detection algorithm.

3.6 Conclusion

The research outlined in this chapter presents a comprehensive solution for object detection and recognition tailored for augmented reality devices. Its modular architecture facilitates the seamless integration of diverse super resolution and detection models within a unified pipeline. The work meticulously surveys existing methodologies and approaches in both super resolution techniques and scene analysis methods, with a particular emphasis on their applicability in immersive environments.

The proposed architecture underwent rigorous testing on both real-world and synthetic datasets, juxtaposed against other cutting-edge approaches in the field. The results gleaned from these experiments unveil a notable enhancement, particularly in discerning low-resolution or distant objects. Moreover, the framework underwent extensive evaluation and analysis across a spectrum of environmental conditions and camera sensor configurations.

In tandem with the assessment on real datasets, we introduced a meticulously curated synthetic dataset. This dataset encompasses annotated data spanning multiple objects and environmental scenarios, thereby facilitating comprehensive assessment and experimentation for future endeavours.

Chapter 4

Evaluation of Environmental Conditions on Object Detection Using Oriented Bounding Boxes for AR Applications

Contents

4.1 Introduction

AR glasses, in general, are used as wearables [111]. The AR glasses project digital information onto surrounding real environment. The scene data captured by the sensors of the AR glasses is not always axis-aligned, in fact, the real environment is often chaotic and objects

are transformed arbitrary. The proposed methods mentioned in the previous chapters up until now were designed to predict axis-aligned bounding boxes. Consequently, speculating on the concept that oriented bounding boxes provide tighter and more accurate localisation information [142], the AR glasses could benefit from such technique to improve scene analysis and, naturally, positively affect digital information projection onto surrounding real environment.

The goal of this work was to suggest a novel method that improves item recognition and prediction by using oriented bounding boxes instead of traditional axis-aligned bounding boxes. The network's YOLO series architecture was used in its construction with the goal of enhancing performance while lowering computational expense. The study's main accomplishments included: a) a method to deal with the problem of strong deviation angle loss and a quicker method for multi-scale feature fusion; b) a thorough comparison of performance under different environmental conditions that affect AR devices; and c) the development of a new dataset containing synthetic objects under various conditions, which was used to evaluate performance impartially across various sensor and environmental parameters.

When the suggested method is compared to the standard two-stage and single-stage procedures, the results consistently show a positive trend, suggesting improved performance. Furthermore, during a more in-depth fine-tuning process, the model produced even better results across the majority of the experiments performed. This shows that the new method has the ability to successfully solve the shortcomings of prior approaches while also providing more accurate and reliable item identification and recognition in a variety of applications. The model's capacity to adapt to varied settings and tackle complicated tasks accounts for the improved performance, making it a potential alternative for future research and real-world applications.

The chapter is structured as follows: In section 4.1, the problem and relevant technologies are introduced. The proposed end-to-end architecture is detailed in section 4.2. Section 4.3 showcases the results using both a real image dataset (VisDrone) and our new synthetic im-

age dataset. An ablation study is described in section 4.4. Finally, the conclusion is presented in section 4.5.

4.2 Oriented Bounding Boxes Object Detection Method

This section provides an in-depth examination of the methodology behind the proposed model [93], which integrates the oriented bounding box feature to bolster its object detection capabilities, see figure 4.1. The conversation centres on the YOLOv5 model's architecture, accentuating crucial distinctions between YOLOv3 and YOLOv5. These differences include the adoption of CSPDarknet53 as the backbone, the enhanced neck utilising the Path Aggregation Network (PANet), and the inclusion of the oriented bounding boxes module to boost the model's overall performance.

The YOLO models are built upon a custom backbone architecture that is based on GoogLeNet, a popular convolutional neural network architecture. YOLO proposed its own backbone network as a faster alternative to the VGG-16 classifier. While VGG-16 is known for its accuracy in object classification, it requires a significant amount of computational power. Specifically, VGG-16 requires 30.69 billion floating point operations, making it slower when compared to YOLO classifier. The YOLO alternative is Darknet-19, which requires only 8.52 billion operations. Despite its lower computational cost, Darknet-19 is still able to achieve relatively high accuracy [132]. Another important aspect, the YOLO architecture was designed to infer information straight from pixels to features in contrast to its competitor which utilised the sliding window technique for the object detection task.

Darknet-53 [133] is an improvement upon Darknet-19 [132] that integrates residual connections. Residual connections are a type of shortcut that allow information to bypass certain layers of a neural network. By integrating these connections, Darknet-53 is able to improve its performance on complex object detection tasks. Overall, YOLO models are a promising approach to object detection that balance accuracy and speed, and their custom backbone architecture allows for flexibility in adapting to different use cases. The advanced version

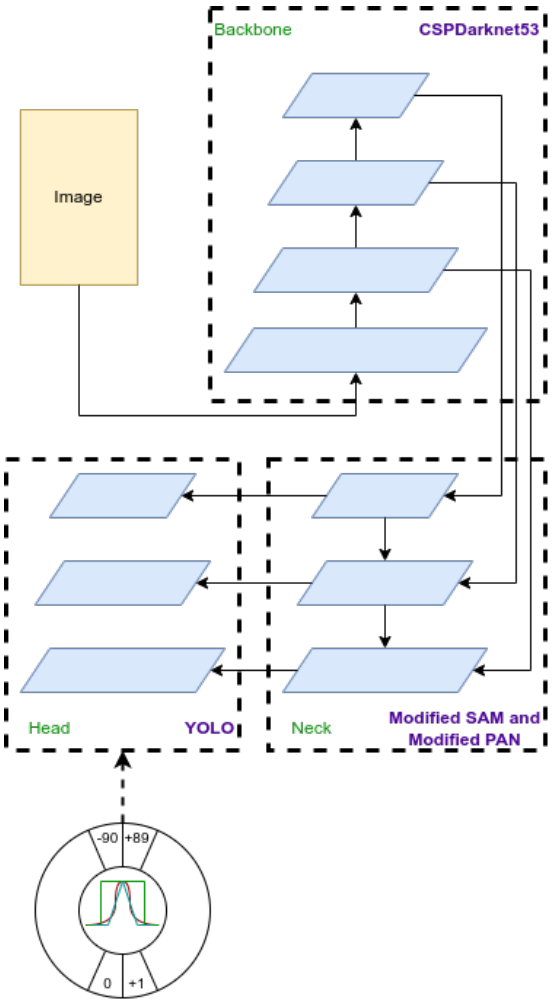


Figure 4.1: An abstract representation of the YOLOv5 with OBB architecture.

of the single-stage object detection model, You Only Look Once (YOLO), employs a more sophisticated backbone known as CSPDarknet53. This innovative backbone is built on the foundation of the CSPNet strategy [53], which works by partitioning the feature map of the base layer into separate components. Following this division, the parts are combined through a cross-stage hierarchical process, enabling more efficient feature extraction.

In addition to the improved backbone, YOLO incorporates the PANet as its "neck." In our case, the "neck" is an intermediate component connecting the backbone and the detection head. While the backbone extracts feature maps from input images, the detection head carries out object detection and classification. The neck acts as a bridge, aggregating and fusing

features from the backbone before forwarding them to the detection head.

This network serves as a feature pyramid network containing a series of bottom-up and top-down layers that contribute to more effective object detection. The PANet's primary function is to aggregate and fuse features at various scales, which significantly enhances the model's overall performance. The architecture of this enhanced YOLO model establishes a streamlined pipeline where CSPDarknet53 is responsible for extracting feature maps, PANet performs feature fusion across multiple scales, and the final layer typically handles the prediction. This pipeline arrangement ensures a more efficient and accurate object detection process. To further boost performance and adapt the model to detect distant objects more effectively, an oriented bounding box module has been integrated. This specialised module allows the model to estimate the orientation of objects in the scene, providing additional context and improving detection accuracy for objects situated far away or partially occluded.

In regression-based methods for object detection, various representations of oriented bounding boxes can be employed. One such representation uses a (x, y, w, h, θ) format, where x, y denote the centre of a prediction, (w, h) represent the width and height, and (θ) is an angle that falls within the range $0 \leq (\theta) \leq 90$. Another similar format also utilises (x, y, w, h, θ), but in this case, (θ) is an angle in the range $0 \leq (\theta) \leq 180$. A more distinct representation employs an eight-parameter format, $(x_0, y_0, x_1, y_1, x_2, y_2, x_3, y_3)$. In this format, each pair of coordinates, x_n, y_n, corresponds to a corner of the oriented bounding box, with n ranging from 0 to 3, such that $0 \leq (\theta) \leq 3$. This alternative representation offers a different way to define oriented bounding boxes in the context of regression-based object detection methods.

In the proposed architecture, as demonstrated in figure 4.1, the model has been modified to approach the prediction of oriented bounding boxes as a classification problem rather than employing the conventional regression methods discussed earlier. To facilitate easier integration, this mechanism has been implemented as a standalone module. By representing each angle of object rotation as a distinct class or category, the model is able to treat the problem as a classification task. However, when using the (x, y, w, h, θ) format, where (θ) falls

within the range $0 \leq (\theta) \leq 90$, the categories corresponding to the edge angles (1 and 90 degrees) tend to converge their losses, which is undesirable from a detection standpoint. To address this issue and incorporate angle-aware context into the regular classification loss, the Circular Smooth Label (CSL) technique is employed. This approach provides essential information for the model, enhancing the accuracy of angle predictions and ultimately improving the overall performance of the oriented bounding box detection.

In conventional object detection models, angle estimation can be problematic due to the periodic nature of angles, where the difference between two angle values might not be accurately represented using a linear scale. The CSL loss function overcomes this issue by employing a circular representation of angles, which preserves the true angular difference between predictions and ground truth labels. The CSL loss function not only enables better angle estimation but also helps in mitigating the problem of abrupt changes in gradient updates. The smoothness of the loss function ensures that the model receives continuous gradient updates during backpropagation, leading to a more stable and efficient learning process. This, in turn, results in improved detection accuracy for objects with varying orientations, especially those situated at a distance or partially occluded. Generally, the integration of the oriented bounding box module as an additional classification branch and the implementation of the Circular Smooth Label loss function significantly enhance the model's ability to account for object orientations. By providing a better representation of angles and ensuring smooth gradient updates, the CSL loss function contributes to a more robust and accurate object detection process.

As mentioned in the section earlier, the model was fine-tuned running 10 epochs following the best practices outline by the authors in [178] with a batch size of 3, due to the hardware limitations enforced by VRAM and number of GPUs, on a real data as well as synthetic data under four different categories in the fist run, and 100 epochs, the number of chosen arbitrary to evaluate the effect of increasing the number of epochs on the training process, with the same batch size and under the same four different categories in the second run. For

detector component, learning rate was set to 0.0032 with the SGD (Stochastic Gradient Descent) optimisation. The learning rate parameter was chosen following the default parameter suggested by the community and authors of [126].

The model was subjected to a fine-tuning process, which involved training on both real and synthetic data to ensure robust performance. This fine-tuning process was conducted in two separate runs, each with a distinct number of epochs and under four different categories to diversify the training data and improve generalisation. In the first run, the model was trained for a relatively short duration, encompassing only 10 epochs with a batch size of 3. This initial training phase aimed to test the model with the real and synthetic data, allowing it to learn basic patterns and features across the four categories. The limited number of epochs in the first run ensured that the model would not overfit to the training data, providing a solid foundation for further fine-tuning.

For the second run, the training process was extended to 100 epochs, with the batch size maintained at 3 with regards to the hardware constraints. This longer training period allowed the model to delve deeper into the nuances of the real and synthetic data, learning more complex features and relationships across the four categories. The extended duration of the second run enabled the model to refine its predictions, thereby enhancing its overall performance and accuracy. Regarding the detector component, a learning rate of 0.0032 was chosen to strike a balance between the speed of convergence and the stability of the training process. To optimise the model's weights, the Stochastic Gradient Descent algorithm was employed, a popular optimisation technique known for its effectiveness in deep learning applications. By using SGD, the model was able to navigate the complex optimisation landscape efficiently, ultimately converging to a solution that achieved a high degree of detection accuracy.

Table 4.1: Results of the proposed Oriented Bounding Boxes Object Detection YOLOv5 model on the Synthetic dataset per 10 & 100 epochs.

Category	Air (10 epochs)	Air (100 epochs)	Ground (10 epochs)	Ground (100 epochs)
Camera	28.13	**74.35**	76.06	**88.12**
Light	82.00	**89.25**	81.85	**82.84**
Weather	60.73	**71.92**	51.88	**83.66**

Table 4.2: Results of the proposed Oriented Bounding Boxes Object Detection YOLOv5 model on the VisDrone dataset per 10 & 100 epochs.

Category	10 epochs	100 epochs
VisDrone	70.28	**76.51**

4.3 Experimental results

In this chapter, a novel approach was proposed to tackle the task of object recognition and scene analysis. The effectiveness of this method was evaluated through a comparison with state-of-the-art approaches using the VisDrone dataset [39] and the synthetic dataset described in the previous chapter. These images depict distant objects positioned at random locations and rotations. Additionally, the VisDrone dataset is annotated in a unique format that includes information about all four vertices of a bounding box, making it ideal for oriented-bounding box prediction, unlike more common formats that only provide corner positions along with width and height. The dataset is composed primarily of RGB images and features categories such as helicopters, small vehicles, and large vehicles, among others.

Table 4.2 presents the results of the model on the VisDrone dataset, comparing the performance after 10 epochs and 100 epochs of training. The table highlights the improvement in detection accuracy as a result of the extended training duration. For the VisDrone dataset, the model achieved a detection accuracy of 70.28% after 10 epochs of training. This initial result indicates that the model was able to learn basic patterns and features within the data during the first run of the fine-tuning process. However, when the training was extended to 100 epochs, the model's detection accuracy increased to 76.51%. This improvement demonstrates the benefits of the longer training period, as the model was able to learn more complex

Table 4.3: Performance (mAP, %) of the Oriented Bounding Boxes Object Detection YOLOv5 method on the Synthetic dataset in the three categories, where PM is the proposed framework, compared with the baseline models.

Category	Sub-category	FasteR R-CNN	YOLOv3	RetinaNet	YOLOX	PM
Air	Camera	35.24%	44.82%	44.79%	47.06%	**81.23%**
	Light	40.66%	63.58%	61.25%	66.76%	**86.05%**
	Weather	35.35%	39.00%	45.57%	40.95%	**77.79%**
Ground	Camera	37.95%	76.02%	78.81%	79.12%	**89.51%**
	Light	32.54%	38.52%	77.12%	40.45%	**85.71%**
	Weather	37.07%	66.32%	76.09%	69.64%	**83.66%**

features and relationships across the dataset. The result also suggests that the fine-tuning process was successful in enhancing the model's performance on the VisDrone dataset.

Further evaluation had been carried out using the Synthetic dataset. Subsets of images were selected from the main dataset obtaining four different categories organised in a way to evaluate the model on specific properties: a) Camera, b) Light, c) Weather d) Sensor. The "Camera" category represented images generated with different camera angles and distances from the objects. The "Light" category contained images generated using variable balanced lighting parameters. The "Weather" category represented images generated using different balanced weather parameters, including varying rain and wind conditions. The "Sensor" category defined different night and thermal vision. The performance of the proposed framework in terms of mAP is shown in table 4.1 for all the four categories.

In conclusion, the results presented in table 4.2 show that the model's performance on the VisDrone dataset improved considerably with the extended training duration. The increase in detection accuracy from 70.28% to 76.51% highlights the effectiveness of the fine-tuning process in refining the model's predictions and overall performance. Table 4.1 presents the results of the model on the Synthetic dataset, showcasing the performance after 10 epochs and 100 epochs of training for both air and ground categories. The table highlights the improvement in detection accuracy for most categories as a result of the extended training duration.

Table 4.4: Performance (mAP, %) of the Oriented Bounding Boxes Object Detection YOLOv5 method on the VisDrone dataset in the three categories, where PM is the proposed framework, compared with the baseline models.

RetinaNet	YOLOv3	Faster R-CNN	YOLOX	PM
52.61%	44.76%	40.50%	46.99%	**76.51%**

For the air category, the model demonstrated significant improvements in detection accuracy across all subcategories when comparing the results after 10 epochs and 100 epochs of training. The detection accuracy for the camera subcategory increased from 28.13% to 74.35%, for the light subcategory from 82.00% to 89.25%, for the sensor subcategory from 60.73% to 71.92%, and for the weather subcategory from 23.57% to 44.11%. These improvements indicate that the model was able to learn more intricate features and relationships within the Synthetic dataset during the extended training period, enhancing its overall performance.

Similarly, in the ground category, the model exhibited improved detection accuracy for three out of four subcategories after 100 epochs of training. The detection accuracy for the camera subcategory increased from 76.06% to 88.12%, for the light subcategory from 81.85% to 82.84%, and for the sensor subcategory from 51.88% to 83.66%. However, the weather subcategory experienced a slight decrease in detection accuracy, dropping from 26.21% after 10 epochs to 20.37% after 100 epochs. This decline might indicate overfitting or the presence of challenging samples in the weather subcategory.

The results presented in table 4.3 demonstrate that the model's performance on the Synthetic dataset improved considerably for most subcategories with the extended training duration. While the weather subcategory in the ground category exhibited a decline in detection accuracy, the overall trend suggests that the fine-tuning process was successful in refining the model's predictions and enhancing its performance across the Synthetic dataset. As could be grasped from the aforementioned results, the Ground dataset didn't produce a dramatic increase in performance when comparing to the Air dataset. However, the Ground dataset had higher results in the initial training. The reason could be the original dataset used during the training which would commonly be biased towards cars.

Table 4.5: Comparison of inference and training times for various object detection methods, showcasing their computational efficiency and suitability for different real-time and training scenarios, where Training Time is measured for 100 epochs in hours (h) and Inference Time represents time it takes to process single image in milliseconds (ms).

Method	Inference Time (ms)	Training Time (h)
Faster R-CNN	200	16.67
YOLOv3	33	2.75
RetinaNet	67	5.58
YOLOX	25	2.08
PM	25	2.08

As depicted in the tables 4.3 and 4.4, the new method presents some performance improvement, in comparison with the previous state-of-the-art solutions. Judging from the results, oriented bounding boxes together with the newer more robust and lightweight architecture played a noticeable role in obtaining better performance. The backbone of the proposed methodology implements a much deeper neural network thus the network could *memorise* a greater amount of feature representations than the baseline models, like YOLOv3 that implements Darknet-19 or RetineNet that implements ResNet50. Furthermore, overall trend points out that even after further fine-tuning the model still kept improving and outperformed itself in the most of the experiments.

The results in the table 4.5 illustrate the trade-offs between inference time and training time across various object detection methods. YOLOX and the method marked as PM demonstrate the fastest inference times at 25 ms, making them particularly well-suited for real-time applications. YOLOv3 follows closely with an inference time of 33 ms, while Faster R-CNN lags significantly behind at 200 ms, highlighting its computational intensity for real-time scenarios. Regarding training time, YOLOX and PM also exhibit the shortest training duration of approximately 2.08 hours, making them efficient for rapid deployment. In contrast, Faster R-CNN requires the longest training time at 16.67 hours, reflecting its complex architecture and feature extraction process. RetinaNet strikes a middle ground with inference and training times of 67 ms and 5.58 hours, respectively, offering a balance between speed and performance. These findings emphasize the importance of selecting a method tailored

Table 4.6: Results with and without Weather rain in the Synthetic dataset using the YOLOv5 with Oriented Bounding Boxes models trained for 10 & 100 epochs.

Epochs	Subset	rain	no rain
10	Air	48.84%	58.11%
100	Air	**80.72%**	**77.77%**
10	Ground	56.28%	43.73%
100	Ground	**85.97%**	**85.33%**

Table 4.7: Results for four different Camera distances in the Air Synthetic dataset using the YOLOv5 with Oriented Bounding Boxes models trained for 10 & 100 epochs.

Epochs	70m	163m	256m	350m
10	36.82%	35.98%	48.39%	22.81%
100	**86.35%**	**81.45%**	**81.55%**	**81.87%**

Table 4.8: Results for four different camera distances in the Ground Synthetic dataset using the YOLOv5 with Oriented Bounding Boxes models trained for 10 & 100 epochs.

Epochs	15m	35m	55m	75m
10	31.36%	66.26%	82.19%	73.84%
100	**97.49%**	**92.17%**	**87.57%**	**83.65%**

to specific use cases, prioritising either real-time efficiency or robust training depending on the application.

4.4 Ablation Study

Table 4.6 presents the results of an ablation study performed on the Synthetic dataset, comparing the model's performance when trained for 10 and 100 epochs under two different conditions: rain and no rain. The table illustrates the impact of weather conditions on the model's detection accuracy for both air and ground subsets.

In the air subset, the model's detection accuracy was higher in the presence of rain compared to no rain for both 10 and 100 epochs of training with an exception of the top row, table 4.6. Where after 10 epochs, the detection accuracy was 48.84% with rain and 58.11% without rain. However, after 100 epochs, the model exhibited improved performance and the trend has flipped, with the detection accuracy increasing to 80.72% with rain and 77.77% without rain. This suggests that the extended training duration allowed the model to bet-

ter adapt to the challenges posed by the rain condition. Similarly, for the ground subset, the model showed higher detection accuracy with rain (56.28%) compared to without rain (43.73%) after 10 epochs of training. After 100 epochs, the detection accuracy improved substantially, reaching 85.97% with rain and 85.33% without rain. This indicates that the model was able to learn more complex features related to weather conditions and achieve better performance in the presence of rain after extended training. Moreover, the rain and no rain results seemed to reach a near parity.

Overall, table 4.6 demonstrates that the model's performance on the Synthetic dataset is affected by weather conditions, particularly rain. Furthermore, absence of rain doesn't always guarantee better performance because there are cases where the model performs better with the rain condition. The results show that extending the training duration from 10 to 100 epochs led to a significant improvement in detection accuracy for both air and ground subsets under both rain and no rain conditions. This suggests that the fine-tuning process was successful in enabling the model to adapt to varying conditions, ultimately enhancing its overall performance.

Continuing to the Synthetic dataset, table 4.7 presents the results of an ablation study performed on the Air Synthetic dataset, comparing the model's performance when trained for 10 and 100 epochs at four different camera distances: $70m$, $163m$, $256m$, and $350m$. The table illustrates the impact of camera distance on the model's detection accuracy. After 10 epochs of training, the model's detection accuracy varied across the different camera distances. The highest detection accuracy was observed at a camera distance of $256m$ (48.39%), while the lowest was at $350m$ (22.81%). The detection accuracy for camera distances of $70m$ and $163m$ was 36.82% and 35.98%, respectively. These results indicate that the model's initial performance is affected by the camera distance, naturally, with the model struggling to achieve consistent detection accuracy across all the distances. However, after 100 epochs of training, the model exhibited a substantial improvement in detection accuracy for all camera distances. The detection accuracy increased to 86.35% for $70m$, 81.45% for $163m$, 81.55% for

$256m$, and 81.87% for $350m$. This demonstrates that the extended training duration allowed the model to learn more complex features related to camera distance and achieve better performance across all tested distances. In addition, the detection performance became more stable hence more predictable.

Generally, table 4.7 shows that the model's performance on the Air Synthetic dataset is influenced by camera distance. Extending the training duration from 10 to 100 epochs led to significant improvements in detection accuracy at all camera distances, suggesting that the fine-tuning process was successful in enabling the model to adapt to varying camera distances and enhance its overall performance. Next, table 4.8 presents the results of an ablation study performed on the Ground Synthetic dataset, comparing the model's performance when trained for 10 and 100 epochs at four different camera distances: $15m$, $35m$, $55m$, and $75m$. The table illustrates the impact of camera distance on the model's detection accuracy.

After 10 epochs of training, the model's detection accuracy varied across the different camera distances. The highest detection accuracy was observed at a camera distance of $55m$ (82.19%), while the lowest was at $15m$ (31.36%). The detection accuracy for camera distances of $35m$ and $75m$ was 66.26% and 73.84%, respectively. These results indicate that the model's initial performance is affected by the camera distance, with the model showing better performance at larger distances. However, after 100 epochs of training, the model exhibited a substantial improvement in detection accuracy for all camera distances. The detection accuracy increased to 97.49% for $15m$, 92.17% for $35m$, 87.57% for $55m$, and 83.65% for $75m$. This demonstrates that the extended training duration allowed the model to learn more complex features related to camera distance and achieve better performance across all tested distances.

Table 4.8 shows that the model's performance on the Ground Synthetic dataset is influenced by camera distance. Extending the training duration from 10 to 100 epochs led to significant improvements in detection accuracy at all camera distances, suggesting that the fine-tuning process was successful in enabling the model to adapt to varying camera distances and enhance its overall performance. However, in comparison to the Air subset, the

results are less stable and follow a more "natural" trend because it is commonly expected for the model to drop performance with increasing distance.

The emphasis in the results highlights the significant improvement in detection accuracy when training for 100 epochs compared to 10 epochs, particularly across varying conditions such as weather and camera distance. For instance, in the Synthetic dataset, the model's detection accuracy increased substantially after 100 epochs, showing improved adaptation to both rain conditions and different camera distances. The accuracy for the air and ground subsets improved markedly, with rain and camera distance having a more stable and predictable impact. This suggests that 100 epochs allowed the model to better learn complex features and adapt to challenging conditions, while 10 epochs resulted in less stable and less accurate performance. These results provide strong evidence that training for 100 epochs enables the model to generalize more effectively and achieve optimal performance across various use cases.

4.5 Conclusions

This chapter explores scene analysis techniques in augmented reality and proposes a modified YOLO architecture with oriented bounding boxes for object detection and recognition. The solution detects objects at large distances and odd angles while maintaining low processing times, suitable for real-time AR applications.

Existing methods are categorised into two-stage and single-stage detectors, with the proposed model evaluated using real and synthetic datasets. The evaluation results show that the proposed solution improves object detection and recognition, particularly for distant objects, compared to state-of-the-art approaches.

In conclusion, this chapter presents an innovative solution designed to address the limitations inherent in current approaches. It offers a comprehensive analysis of AR scene analysis and object detection methods, proposing a novel methodology that addresses existing limit-

ations and demonstrates a potential for real-world augmented reality applications.

Chapter 5

Enhancing Few-Shot Object Detection Through Data Augmentation Techniques

Contents

5.1 Introduction

In the domain of AR maintenance, the maintenance processes require work with multiple types of equipment. Furthermore, unlike more common objects such as food, clothes, people, etc., the maintenance, as part of the industry, is often involved with proprietary rare or novel equipment the data of which is not readily available and scarce. In situations like these, FSL techniques could be beneficial. For instance, DA artificially expands a scarce dataset helping AI models such as the one mentioned in the previous chapters. A more detailed

overview of the DA techniques could be viewed in the earlier part of the thesis, chapter 2.4. Applying these DA techniques for FSL benefits the AR maintenance [34], for example, during the maintenance processes, an AR personnel, as part of the work, could install a new equipment. However, the new equipment might be absent in the database of images for object detection. To address this issue, the AR personnel would be required to take a couple of images of the new equipment, instead of large number of images as what the contemporary AI community tries to address ([180], [171], [153]), and upload to the database of images for object detection. Thus, the AR personnel could collect the data onsite whilst wearing the AR glasses, communicating with the offsite experts, and sharing new information to facilitate AI technology in real-time by lifting off data collection requirements thus speeding up the maintenance processes.

Object detection stands as a fundamental task within the domain of computer vision, serving as a cornerstone in various applications ranging from autonomous driving to surveillance and medical imaging. However, traditional object detection algorithms often rely on large labelled datasets for training, posing limitations in scenarios where labelled data is scarce or expensive to acquire. Few-shot learning, a subset of machine learning, addresses this challenge by enabling models to learn from a limited number of labelled examples.

In recent years, the burgeoning interest in few-shot learning approaches has extended to the field of object detection, offering promising prospects for generalising to unseen object categories with minimal labelled data. A pivotal strategy for enhancing the performance of few-shot learning lies in data augmentation. This practice involves generating synthetic data samples from existing labelled data to augment the training dataset. By employing data augmentation techniques, object detection models can potentially improve their generalisation capabilities, particularly in scenarios with limited labelled data.

According to authors [168], a problem few-shot learning aims to address is the problem of training a machine learning algorithm given only a limited number of samples of a new class. Commonly, few-shot learning is treated as an approach that facilitates an N-way K-shot

learning scheme with support and query sets [151]. The few-shot learning approach learns a mapping of a similarity function between the support and query sets. The similarity function typically outputs a probability value of the similarity by comparing generated embeddings. The principles of the aforementioned approach are exploited by authors in [138] introducing a variant of Siamise Network. Another example that follows the approach is presented in [148] as Prototypical Network. Lastly, Matching Networks are well-known representatives of few-shot learning approach [159]. In extreme cases, zero-shot learning approach could be used to address the problem of training a machine learning algorithm given none samples of a new class. In situations such as like this, an auxiliary medium is used, for example, textual description to support the training process [137]. In this chapter, the proposed method addresses the same problem as few-shot learning using a novel approach different from the few-shot learning approach mentioned here.

The focus of this chapter revolves around the exploration of data augmentation strategies tailored to address the problem of few-shot learning mentioned in the previous paragraph using a different approach. By augmenting the training dataset with synthetic data samples, the aim is to bolster the robustness and generalisation capacities of object detection models, empowering them to glean insights effectively from a small number of labelled examples. The research endeavours to tackle several key objectives: a) Investigating the efficacy of various data augmentation techniques, including geometric transformations, colour jittering, and synthetic data generation, in evaluating the few-shot learning performance in object detection; b) Evaluating the ramifications of data augmentation on the generalisation abilities and robustness of object detection models across diverse datasets and application scenarios; c) Providing nuanced insights into the trade-offs among data augmentation strategies, computational efficiency, and detection accuracy, thereby equipping practitioners with the knowledge to make informed decisions when designing and training object detection models for few-shot learning tasks.

Through these objectives, the aspiration is to contribute to the advancement of few-shot

learning techniques in object detection, facilitating their integration into real-world applications where labelled data remains limited or cost-prohibitive. This research endeavours to bridge the chasm between traditional object detection methodologies and the emerging concept of few-shot learning, offering novel insights and methodologies to enhance the capabilities of object detection systems in challenging and data-constrained environments.

5.2 Data Augmentation combined with YOLOv8 method for Few-Shot Learning

In the following section, the proposed methodology is covered in detail. The methodology is based on the recent version of YOLOv8 model. The base model is integrated with additional DA techniques as depicted in the figure 5.1. The DA techniques are used to better address the data scarcity issue introduced by the FSL problem. FSL focuses on learning from a small set of data, consequently the data scarcity is an inherited characteristic. The methodology is designed to work with a broad set of DA techniques. The first set of DA techniques is utilised to generate extra data from the input data thus improving the data scarcity issue. The second set of DA techniques is utilised to address the overfitting issue. The overfitting issue comes from fact that the newly generated data could produce samples that are too similar. Logically, the training on a dataset that contains copies of similar data might result in a biased model. To avoid such outcome, the second set of DA techniques is applied with a certain probability per image.

As mentioned earlier, to address the Few-Shot Learning problem, the first set of the DA techniques involve techniques such as Sharpen, Solarise, and Superpixels. The Sharpen DA technique is an image manipulation technique, its purpose is to enhance the details and edges of an image making them more pronounced. The enhancement is achieved by applying a smoothing filter such as Gaussian blur. Then, by subtracting the smoothed image from an original image and amplifying the difference by the sharpening factor. The result is added to the original image and become the final output of the Sharpen DA technique. The Solarise DA

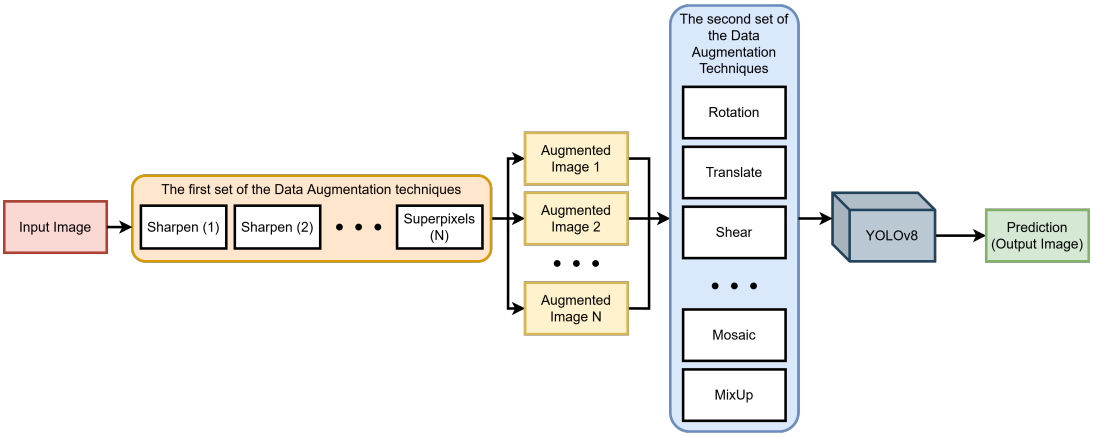


Figure 5.1: The proposed methodology utilising the Data Augmentation techniques and YOLOv8 architecture.

technique is another image manipulation technique. It reverses the tone of the of the input image generating a unique visual effect. At its core, the Solarise DA technique involves inverting the values of the image above a certain threshold while the value below remain unchanged. Finally, the result is blended with the original image, e.g. additive blending.

The Superpixels DA technique is yet another image manipulation technique that aims to group pixels by similarity, such as colour or texture. This technique introduces structural variation by partitioning the image into segments and reducing overall complexity of the image. The segmentation is achieved using an algorithm introduced by author in [2], Simple Linear Iterative Clustering (SLIC). In its nature, the algorithm samples the image at regular intervals creating a grid-like structure where each sample becomes a centre of a cluster. Each centre point would then be updated by calculating the mean colour and spatial location of all pixels in the cluster. The process is repeated until convergence. Once convergence is achieved, some randomness is introduced into the image to distort it, i.e. rotation, scaling, translation, and colour shifting, within each of the pixel groups. Finally, the result is blended with the original image.

The second set of DA techniques is comprised out of more common DA techniques. They are Rotation, Translation, Scaling, Shearing, Perspective changing, Flipping, Hue-Saturation-Value (HSV) manipulation, Mosaic, and MixUp. Rotation rotates the image by a given angle

in degrees. Translation offsets the image by a certain number of pixels along the X- and Y-axis. Scaling transforms the size of an image along the axes. Shearing skews the image by offsetting the pixels along the axes simulating "*dragging*" effect. The Perspective technique is similar to Shearing, but transforms the entire image in a way that replicates the change of a viewing angle. Flipping mirrors the image either horizontally or vertically, this is a simple yet effective trick that helps with overfitting. Another common approach is the manipulation of Hue-Saturation-Value. Adjusting some or all of the HSV parameters could introduce some variation mimicking certain environments in the image, like the change of the time of the day.

Finally, the methodology utilises the Mosaic and MixUp DA techniques. The Mosaic technique is a conceptually simple idea that was introduced to the series of YOLO models quite recently. It combines multiple images, typically four images, into a single image with the corresponding ground-truth annotations. The images are pooled from the same batch during the training process. The MixUp technique is another variation of a data augmentation technique that uses multiple images. MixUp attempts to blend multiple images as is with a certain level of transparency, commonly 50% of transparency. The result is considered to improve the overall generalisation performance of the training model [181].

The YOLOv8 model is an anchor-free object detection AI model. It is a natural evolution over the YOLOv5 model aimed at improving overall prediction performance as well as the complexity of the model in terms of computing power. The model was also implemented with the data scarcity and energy efficiency in mind therefore making it a perfect candidate as the basis of the proposed methodology.

Any AI model consists out of neural networks that are combined into layers, e.g. a layer of convolutional neural network. To better understand the architecture of the YOLOv8 model, the layers are grouped into blocks as depicted in figure 5.2. The blocks constitute the main building components of the YOLOv8 architecture including the Conv block, the C2f block, and the SPPF block. Additionally, there is the Bottleneck block that acts as the inner com-

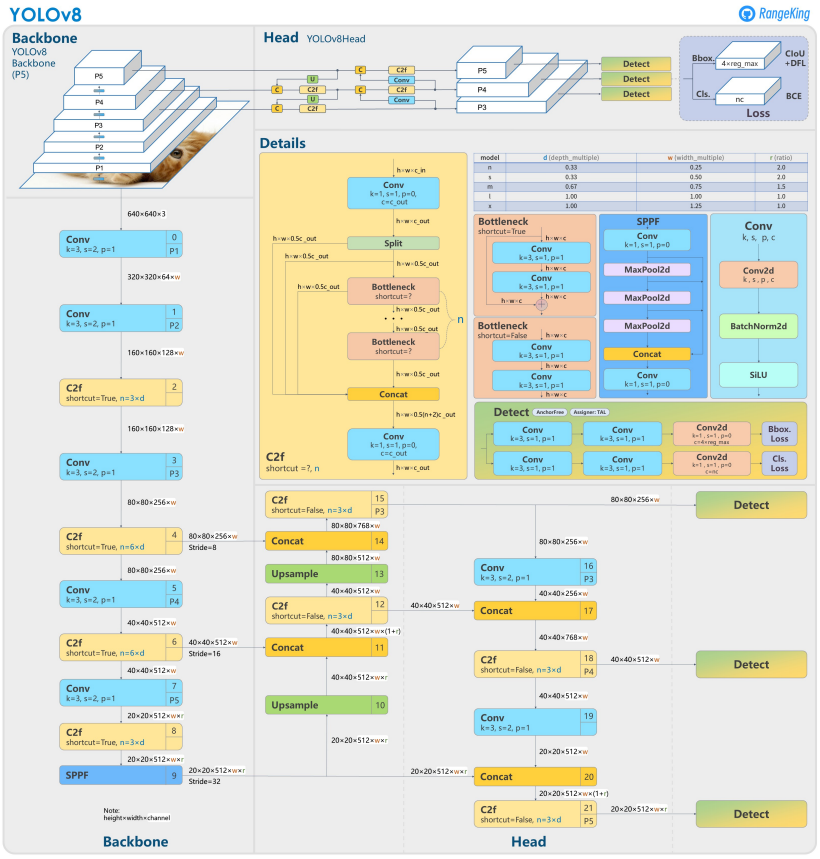


Figure 5.2: A diagram depicting the architecture of the YOLOv8 architecture [130].

ponent of the Conv block. Finally, the Detect blocks are responsible for final bounding-box regression and classification. The blocks are covered in greater detail in the subsequent paragraphs.

The Conv block is a block responsible for the performing convolutional operations using convolutional neural network layers. This is a basic building block of the YOLOv8 model representing a common triplet of Convolutional layer, Batch Normalisation, and Activation Layer. The Convolution layer is a neural network that expects three main parameters: kernel size, stride, and padding. The Batch Normalisation is a regulatory technique that helps address the overfitting. The activation layer is function that translate the arbitrary result of the convolutional neural network into a more manageable range of values used for prediction. In the instance of the YOLOv8 model, instead of more common ReLU activate function [90],

a Sigmoid-Weighted Linear Unit (SiLU) activation layer [43] is used.

The C2f block is an iterative improvement over the C3 block found in YOLOv5 [126]. The input feature map is processed by the first convolutional layer where the output is halved. First half uses a *shortcut* and concatenated back to at the end of the block right before the final convolutional layer. The second half is processed via a series of the Bottleneck blocks where the output is halved again and the first half is sent to the end and the second is passed along the series of the Bottleneck blocks. The Bottleneck block by design is similar to a more common residual block found in the models, such as RetinaNet [99]. The difference is that the Bottleneck block provides the control of whether to use the shortcut or not. The shortcut allows the network to preserve some of the residual features helping the model to better generalise. Nonetheless, after a series of the Bottleneck blocks the final output is concatenated with the rest of the feature maps divided earlier and passed along to the final convolutional layer.

The Spatial Pyramid Pooling Fast (SPPF) block is a scale invariant variation of Feature Pyramid Network (FPN) designed for improved speed performance and high accuracy. Starting from the Conv block, the SPPF block introduces a set of Max Pooling layers are combined together to better address variable size of object in the image whilst preserving the spatial information essential for localisation of the object in the image. The concatenated result is sent to the final Conv block and outputted to the neck of the model.

The neck is an intermediate area between the backbone and the head. It connects the last layers of the backbone with the detectors of the head in such a way to provide object detection at three main scales: small, medium, and large. This approach allows the detection of objects of variable sizes. As depicted in figure 5.2, the neck combines the last layers by upsampling and concatenating the outputs starting from the last layer of the backbone. Finally, the head performs anchor-free regression of the bounding boxes as well as the classification of the predicted objects.

The proposed methodology combines the aforementioned solutions into a single framework. The combination of the DA techniques from the both sets has the potential to help address Few-Shot Learning problem as well as some of the energy efficiency aspects [156]. The first set of DA techniques help address the data scarcity issue, whereas the second set of the DA techniques regularises and avoids the overfitting, whilst the overall architecture reduces model complexity thus reduces training time and energy consumption. A more detailed of the results is provided in the following section. The proposed methodology is capable of applying a list of augmentations on a given input image in a sequential manner, where each augmentation is applied one after another on the same input image. There are no limitations to what DA techniques could be applied to the first set of the DA techniques, as depicted in 5.1. However, the number of DA techniques should be picked with great care as the resulting number of images could explode leading to the overfitting. Therefore, a suggested number of DA techniques is three. The resulting augmented images, see figure 5.1, are passed along to the second set of the DA techniques that are applied together, *in parallel*, with a certain probability. Adjusting the probabilities will affect the performance of the model, furthermore, this allows automatic search of the best probabilities akin to RandAugment [27].

5.3 Experimental Results

An overview of results is provided in the following section. The results were obtained using two distinct datasets depicting various environments such as urban, forest, desert, and grass valley: VisDrone, and Synthetic. To facilitate the experiments, the datasets were split into subsets. For example, 5 percent experiment contains 5 percent of total number of images of an entire dataset, except for 1 sample experiment because it contains only a single image from a dataset. The results were conducted with three methods which are baseline model (BASE), a fine-tuned model (FT), and a fine-tuned model with data augmentation (FT+DA). The baseline model is a model generalised on a well-generalised dataset such as COCO dataset followed by a transfer learning to the target task using VisDrone and Synthetic datasets. The fine-tuned model is a model that was fine-tuned on the target dataset with a prior selection of a

Table 5.1: Performance comparison of Data Augmentation combined with YOLOv8 model on the VisDrone dataset. FT stands for a Fine-Tuned method, and FT+DA stands for the Fine-Tuned with Data Augmentation method.

VisDrone		
METHOD	**FT**	**FT+DA**
Data %	**RESULT (mAP)**	
100%	73.40%	96.40%
75%	73.30%	84.70%
50%	69.60%	95.10%
25%	57.80%	80.10%
20%	58.80%	92.40%
5%	43.20%	36.80%
1 sample	7.87%	10.20%

pretrained model. The fine-tuned model with data augmentation is a model fine-tuned on the target dataset using data augmentation on a well-generalised pretrained weights. Tables 5.1 and 5.2 demonstrate the performance results of the model measured using the mean average precision, i.e. mAP. The subsequent tables 5.3 and 5.4, display the total amount of emission of CO2, in grams, the methods have generated during the training process. Tables 5.5 and 5.6 list the total amount of energy that was consumed during the training process in watts per hour. To calculate the emissions and energy consumption, specific software was utilised [23] that monitored the experiments directly at the computer at set intervals of time, e.g., every 15 seconds. Carbon emissions of the consumed electricity are calculated as a weighted average of the emissions from the different energy sources that are used to generate electricity, including fossil fuels and renewables based on each kilowatt-hour consumed by the computer. Finally, the duration of each experiment, in seconds, is listed in tables 5.7 and 5.8.

The performance results listed in the table 5.1 is comparing an original fine-tuned model against a fine-tuned model with our data augmentation using the VisDrone dataset. The Visdrone is a dataset of pictures that represents an industrial environment with vehicles such as cars, trucks, vans, and buses. To address the Few-Shot Learning problem, the dataset was split into subsets. The subsets are 1 sample of total number of images from the dataset, 5% of total number of images from the dataset, 20% of images, 25% of images, 50% of images,

Table 5.2: Performance comparison of Data Augmentation combined with YOLOv8 model on the Synthetic dataset. FT stands for a Fine-Tuned method, and FT+DA stands for the Fine-Tuned with Data Augmentation method.

Synthetic		
METHOD	**FT**	**FT+DA**
Data %	**RESULT (mAP)**	
100%	53.50%	87.50%
75%	46.00%	71.00%
50%	39.90%	58.60%
25%	35.60%	48.50%
20%	41.20%	51.00%
5%	25.30%	40.70%
1 sample	1.97%	2.33%

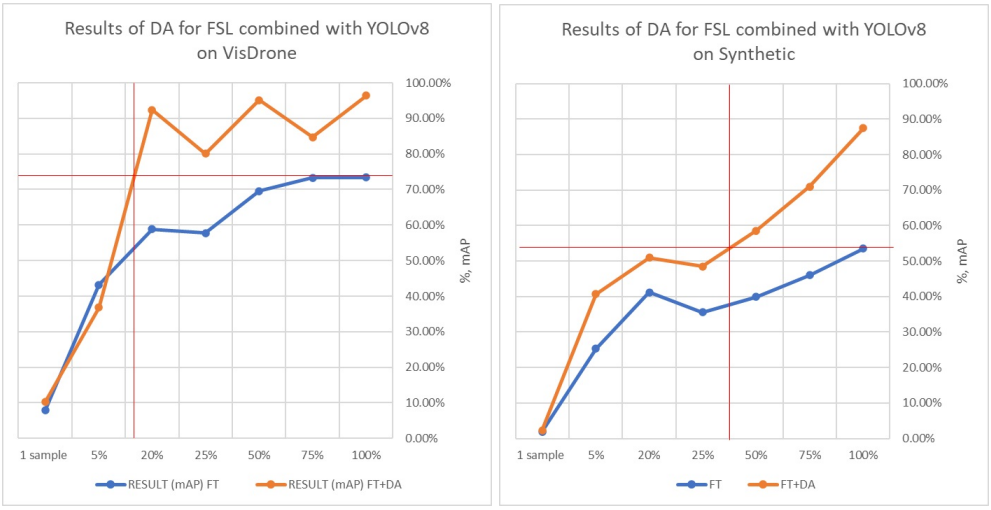


Figure 5.3: Graphs visualising the performance of the Data Augmentation for Few-Shot Learning combined with YOLOv8 method on VisDrone and Synthetic datasets.

75% of images, and 100% of images, i.e. the entire dataset. The results indicate that the model struggles when given a single sample for the training, however, the addition of the data augmentation noticeably improves the performance of the prediction. Unexpectedly, at the 5%, as provided in the table 5.1, the original fine-tuned model prevails over the data augmentation model. A similar trend could be observed at 25% and 75%. However, the data augmentation model outperforms the original model with much less data, i.e. at 20% mark. Consequently, earlier convergence of the data augmentation model suggests that the model has reached it's plateau with fewer samples.

Table 5.3: Total Emissions of Data Augmentation for Few-Shot Learning Object Detection model on Visdrone. BASE is the model trained from scratch, FT is the fine-tuned model, and FT+DA is the fine-tuning with data augmentation techniques applied.

Visdrone			
METHOD	**TOTAL EMISSION (g, CO2)**		
Data %	**BASE**	**FT**	**FT+DA**
100%	3.38×10^{0}	1.19×10^{-2}	1.65×10^{-2}
75%	3.38×10^{0}	9.68×10^{-3}	1.04×10^{-2}
50%	3.38×10^{0}	6.75×10^{-3}	6.55×10^{-3}
25%	3.38×10^{0}	6.36×10^{-3}	5.35×10^{-3}
20%	3.38×10^{0}	7.28×10^{-3}	9.82×10^{-3}
5%	3.38×10^{0}	5.34×10^{-3}	2.83×10^{-3}
1 sample	3.37×10^{0}	2.96×10^{-4}	2.22×10^{-4}

Table 5.4: Total Emissions of Data Augmentation for Few-Shot Learning Object Detection model on Synthetic. BASE is the model trained from scratch, FT is the fine-tuned model, and FT+DA is the fine-tuning with data augmentation techniques applied.

Synthetic			
METHOD	**TOTAL EMISSION (g, CO2)**		
Data %	**BASE**	**FT**	**FT+DA**
100%	3.42×10^{0}	4.34×10^{-2}	1.71×10^{-1}
75%	3.41×10^{0}	3.55×10^{-2}	1.84×10^{-1}
50%	3.39×10^{0}	1.57×10^{-2}	5.84×10^{-2}
25%	3.38×10^{0}	8.57×10^{-3}	2.87×10^{-2}
20%	3.38×10^{0}	9.51×10^{-3}	9.35×10^{-2}
5%	3.38×10^{0}	9.37×10^{-3}	3.04×10^{-2}
1 sample	3.37×10^{0}	0.24×10^{-3}	1.71×10^{-3}

The table 5.2 provides detailed overview of the performance results of the original fine-tuned model and the fine-tuned model with our data augmentation on the Synthetic dataset and figure 5.3 visualises the performance of the models over different subsets of data. The Synthetic dataset contains pictures of hazardous environments involving fire in urban environment. In the same manner as the VisDrone dataset, the Synthetic dataset was split into subsets. The performance results of the original fine-tuned model and the data augmentation fine-tuned model follow a similar pattern. Slight drop could be observed at 25% and 75% marks in both results of the methods what suggests that it originates from data variability. Notably, the fine-tuning with data augmentation allows the new model, at 50% mark, achieve better results that the original model, at 100% mark.

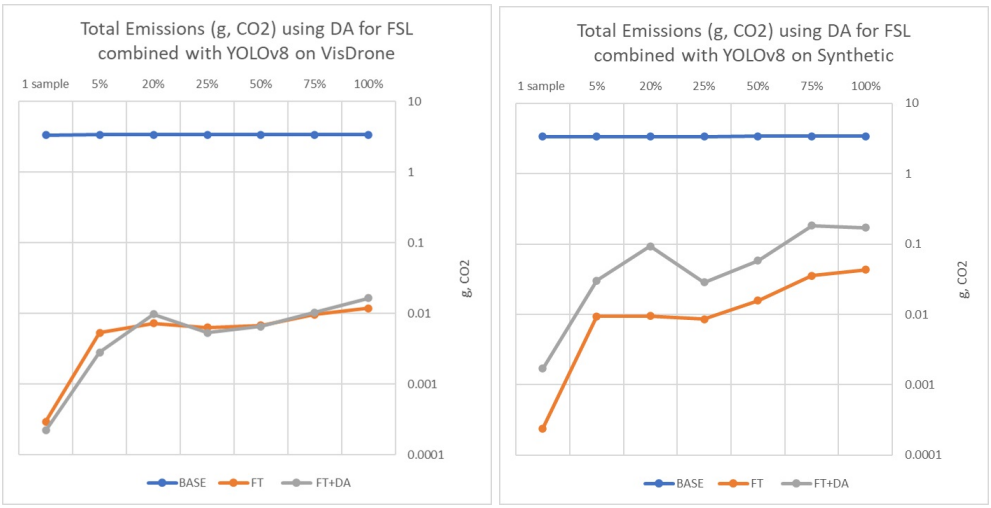


Figure 5.4: Graphs visualising the total emission (g, CO2) of the Data Augmentation for Few-Shot Learning combined with YOLOv8 method on VisDrone and Synthetic datasets.

The total emissions of the model over the duration of each experiment coinciding with the experiments in the performance results. Each experiment was ran at least three times and an error is calculated and provided alongside with the measurement. The total amount of emission is calculated in grams of CO2. The results are provided for both datasets: VisDrone and Synthetic. The results of the total emissions are structured in the same way as the performance results. As demonstrated in tables 5.3 and 5.4, usage of a pretrained model greatly contributes to reduction of the total amount of emissions generated during the training process. The fine-tuning process utilises an already pretrained model thus it avoids generating extra emissions by picking a well-generalised pretrained model, see figure 5.4. Observing the results, tables 5.3 and 5.4, the fine-tuning is superior to training-from-scratch. Although Data Augmentation incurs a modest overhead through the generation of additional augmented data for training purposes, comparative analysis of outcomes reveals competitive performance, with instances demonstrating superior results even with reduced input data. In cases, when compared with the training-from-scratch, it is noticeable that fine-tuning with data augmentation consumes far less energy.

The tables 5.5, 5.6 show the total amount of energy consumption for each experiment. The energy consumed summarises the consumed energy of CPU, GPU, and RAM. A visualisation

Table 5.5: Total Energy Consumption of Data Augmentation for Few-Shot Learning Object Detection model on the VisDrone dataset. BASE is the model trained from scratch, FT is the fine-tuned model, and FT+DA is the fine-tuning with data augmentation techniques applied.

VisDrone			
METHOD	**TOTAL ENERGY (Wh)**		
Data %	**BASE**	**FT**	**FT+DA**
100%	12.61×10^{0}	4.44×10^{-2}	6.16×10^{-2}
75%	12.60×10^{0}	3.61×10^{-2}	3.85×10^{-2}
50%	12.59×10^{0}	2.52×10^{-2}	2.44×10^{-2}
25%	12.59×10^{0}	2.37×10^{-2}	2.05×10^{-2}
20%	12.59×10^{0}	2.72×10^{-2}	3.65×10^{-2}
5%	12.58×10^{0}	1.99×10^{-2}	1.06×10^{-2}
1 sample	12.56×10^{0}	1.10×10^{-3}	8.20×10^{-4}

Table 5.6: Total Energy Consumption of Data Augmentation for Few-Shot Learning Object Detection model on the Synthetic dataset. BASE is the model trained from scratch, FT is the fine-tuned model, and FT+DA is the fine-tuning with data augmentation techniques applied.

Synthetic			
METHOD	**TOTAL ENERGY (Wh)**		
Data %	**BASE**	**FT**	**FT+DA**
100%	12.73×10^{0}	1.61×10^{-1}	6.41×10^{-1}
75%	12.70×10^{0}	1.34×10^{-1}	6.88×10^{-1}
50%	12.63×10^{0}	6.17×10^{-2}	3.53×10^{-1}
25%	12.60×10^{0}	2.97×10^{-2}	1.75×10^{-1}
20%	12.60×10^{0}	4.43×10^{-2}	2.21×10^{-1}
5%	12.60×10^{0}	4.56×10^{-2}	1.13×10^{-1}
1 sample	12.57×10^{0}	1.86×10^{-3}	5.73×10^{-3}

of the detailed overview could be observed in figure 5.5. The results demonstrate that the original fine-tuning model and the fine-tuned model with data augmentation require noticeably less energy than the model trained-from-scratch. A strategically well chosen pretrained weights provide with equal or greater performance and save more energy. The general trend is as expected shows that with the increase size of the training subset the energy consumption increases, partially also demonstrated later in the processing time tables 5.7, 5.8. Looking from the perspective of the original model and the data augmentation model, the data augmentation model required twice as much energy to surpass the best original model in table 5.6 at 50% mark. The VisDrone results demonstrate that the data augmentation model required 17.7% less energy, at 20% mark, than the best original model (at 100%).

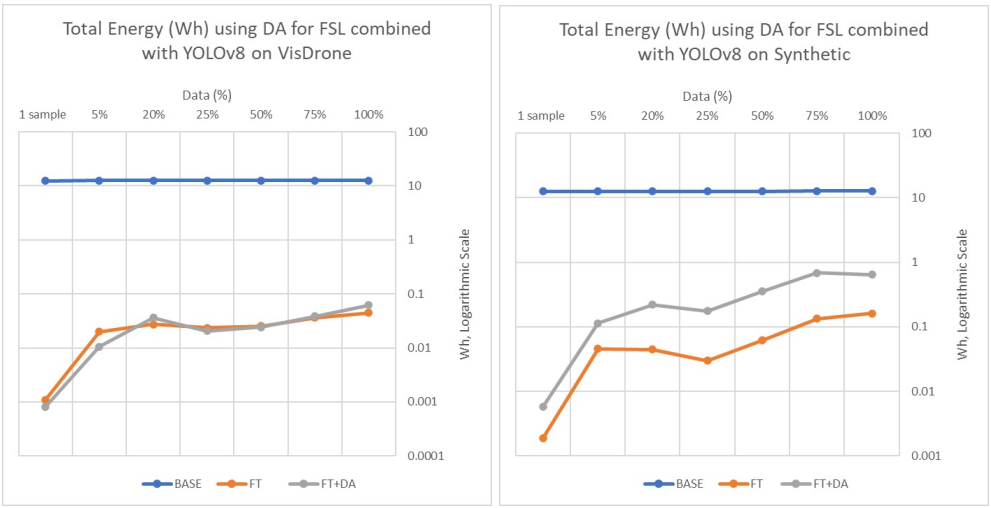


Figure 5.5: Graphs visualising the total energy consumption (Wh) of the Data Augmentation for Few-Shot Learning combined with YOLOv8 method on VisDrone and Synthetic datasets.

Table 5.7: Processing Time of Visdrone dataset using the Data Augmentation for Few-Shot Learning Object Detection combined with YOLOv8 method. BASE is the model trained from scratch, FT is the fine-tuned model, and FT+DA is the fine-tuning with data augmentation techniques applied.

Visdrone			
METHOD	**PROCESSING TIME (s)**		
Data %	**BASE**	**FT**	**FT+DA**
100%	168,613	211	934
75%	168,508	167	595
50%	168,372	132	368
25%	168,370	210	301
20%	168,431	231	528
5%	168,359	90	193
1 sample	168,048	21	22

The processing time results of the VisDrone and Synthetic datasets comparing three methods, tables 5.7 and 5.8 and an illustration is provided by figure 5.6. The first method is train-from-scratch following with fine-tuning on the task specific dataset, the second method is selecting a generalised pretrained model and fine-tuning on the task specific dataset, and the final method is the same as the second method but differ by introducing a list of data augmentation techniques to further facilitate the Few-Shot Learning problem. The processing time was logged on the platform with the following specification:

Table 5.8: Processing Time of Synthetic dataset using the Data Augmentation for Few-Shot Learning Object Detection combined with YOLOv8 method. BASE is the model trained from scratch, FT is the fine-tuned model, and FT+DA is the fine-tuning with data augmentation techniques applied.

Synthetic			
METHOD	**PROCESSING TIME (s)**		
Data %	**BASE**	**FT**	**FT+DA**
100%	170,088	2,068	10,074
75%	169,713	1,693	8,565
50%	168,785	765	2,761
25%	168,456	436	1,368
20%	168,509	489	4,435
5%	168,561	541	1,529
1 sample	168,042	22	128

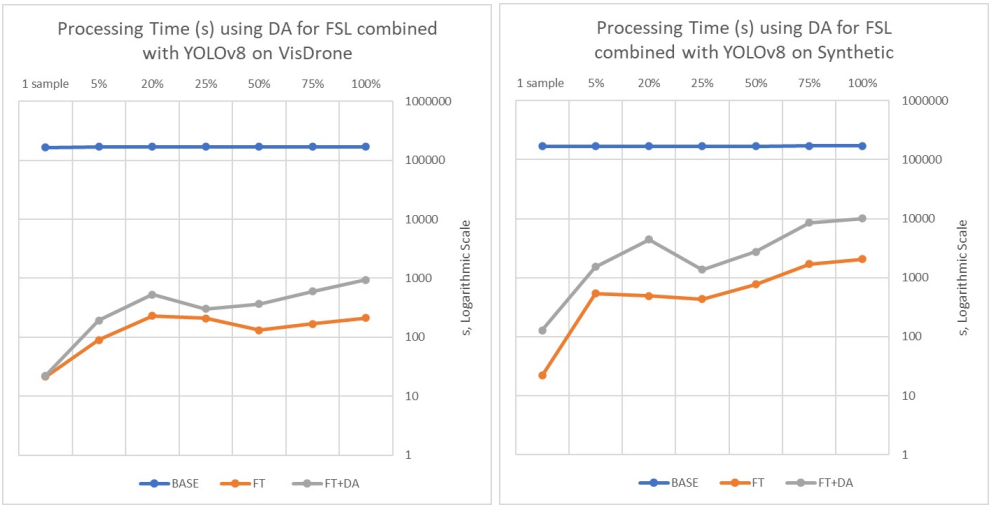


Figure 5.6: Graphs visualising the total processing time (s) of the Data Augmentation for Few-Shot Learning combined with YOLOv8 method on VisDrone and Synthetic datasets.

- Ubuntu 20.04
- Intel(R) Xeon(R) Gold 5122 CPU @ 3.60GHz
- 64 GB RAM
- 1 x NVIDIA GeForce RTX 2080 Ti

Observing the processing time results, see figure 5.6, the train-from-scratch method is the most time-consuming and, inherently, the most energy demanding and it emits the most

amount of CO2. In addition, the train-from-scratch method doesn't introduce any evident performance improvements. The original fine-tuned model has lower processing time due to the fewer number of images in the dataset. The data augmented fine-tuned model takes from 2 to 4 times longer to complete, when compared with the similar sized experiments, i.e. 5% FT against 5% FT+DA, as demonstrated in tables 5.7 and 5.8. However, if compared with the best original model results, at 100% of the dataset, the data augmentation model of VisDrone surpassed the fine-tuned model, abbreviated as FT in figure 5.6, using only 20% of the dataset but it took twice the time, i.e. our approach could achieve competitive performance using less data. The same comparison for the Synthetic dataset, the data augmentation model surpassed the original model using 50% of the dataset and it took 33% more time. Still, comparing the data augmentation with the train-from-scratch, the added time is insignificant, but allows to reach comparative performance results using less data.

5.4 Conclusion

In conclusion, this chapter has explored the potential of data augmentation as a means to bolster few-shot learning performance in object detection tasks. By augmenting the training dataset with synthetic data samples, the aimed was to enhance the robustness and generalisation capabilities of object detection models, particularly in scenarios with limited labelled data. Through a comprehensive investigation of various data augmentation techniques, including geometric transformations, colour jittering, and synthetic data generation, as an outcome, insights into their efficacy and trade-offs in improving few-shot learning performance were gained.

The results of the research could be applied towards the AR maintenance. The DA techniques for FSL could either improve the performance of the object detection models or achieve competitive performance given fewer samples to train on in comparison with the models that were trained on a full dataset. It highlights the potential for the use cases mentioned earlier in this chapter where the AR personnel could utilise such methodology to achieve quicker

maintenance processes and retaining the benefits obtained by utilising AR such as better user experience.

Our research findings underscored the importance of data augmentation in mitigating the challenges posed by data scarcity in object detection tasks. By expanding the training dataset through synthetic data generation, notable improvements in the generalisation abilities and robustness of object detection models were demonstrated, enabling them to effectively learn from a small number of labelled examples. Moreover, our evaluation across diverse datasets and application scenarios shed light on the applicability and limitations of different data augmentation strategies, providing valuable guidance for practitioners in designing and training object detection models for few-shot learning tasks.

Looking ahead, there remain several avenues for further exploration and refinement of data augmentation techniques in the context of few-shot learning object detection. Future research could delve deeper into the development of novel augmentation strategies tailored specifically for object detection tasks, considering the intricacies of object localisation and classification. Additionally, investigating the interplay between data augmentation and other factors such as model architecture, optimisation techniques, and domain-specific considerations could yield further insights into enhancing few-shot learning performance.

Ultimately, the integration of data augmentation into the few-shot learning pipeline represents a promising approach for addressing the challenges of data scarcity in object detection tasks. By harnessing the power of synthetic data generation and augmentation techniques, new possibilities for advancing the capabilities of object detection systems in real-world applications were unlocked, where labelled data remains limited or difficult to obtain. Through continued research and innovation in this area, we can pave the way for more robust, adaptive, and efficient object detection solutions, driving forward the frontier of computer vision and machine learning.

Chapter 6

Analysis of the 3D object detection for Augmented Reality using a synthetic dataset

Contents

6.1 Introduction

The work presented in this thesis has covered various methods addressing scene analysis via object detection using SOTA approaches. However, the methodologies up until this point were designed to predict 2D bounding boxes, either axis-aligned or oriented. These methods have their own merits, although, as could be observed throughout the chapters, the AR and AR applications closely interact with the real environment. However, to fully grasp the scene in a real 3D world, 2D recognition and detection results alone are no longer sufficient. [38] Observing the real environment, all things are in 3D. Thus, the AR glasses, applications,

and, specifically, AR maintenance need to interact with the 3D environment. 3D information provides spatial information that could guide the AR user, or the AR maintenance personnel, to accomplish tasks that require spatial interaction. For example, in the case of the AR maintenance personnel, 3D instructions with animations could be superimposed onto the real hardware. Furthermore, remote assistance could connect to the AR maintenance application and communicate changes to the surrounding 3D environment inferred by the 3D object detection model, in turn, facilitating real-time support.

In the last decades, advances in computer vision have fostered the design and implementation of object recognition methods, increasing computational performance and lowering process time [190]. As a result, current AR technologies based on object recognition use complex computer vision techniques to detect and track objects in the real world. Examples of such technologies include the You Only Look Once model [7], homomorphic filtering and Haar markers [58] and the Single Shot Detector [35]. The use of Convolutional Neural Networks and Deep Learning led to faster and more accurate detection processes [184]. However, the augmented reality experience could be improved by projecting 3D objects into the augmented reality space surrounding the user inferred from the real environment.

The aim of this study is to analyse a novel 3D solution that evaluates performance of the 3D bounding box prediction in various conditions. The proposed architecture consists out of smaller components. The first component predicts a 2D bounding box common to the standard object detection task. It achieves it by generating and processing Heatmaps, Embeddings, and Offsets [89]. The three outputs are then further processed in Cascade Corner Pooling and Centre Pooling [40] components which infer the final positions of the 2D bounding box. The next component is then estimates depth using a Multi-Scale Deep Network [42], in two-stages, the first stage collects global information, and, the second stage refines the global information to produce more precise prediction. The next component estimates the 3D dimensions and orientation of the object using CNNs. The 3D dimensions are regressed directly from feature map outputs using fully-connected layers, however, the orientation is formu-

lated as a classification task and regressed accordingly using the hybrid discrete-continuous MultiBin loss [113]. The final 3D bounding box is constrained by the predicted 2D bounding box with an assumption that the 2D bounding box has been trained to match the position of the 3D bounding box.

The chapter is organised as follows: Section 6.1 introduces the problem and relevant technologies; Section 6.2 describes the proposed architecture; Section 6.3 presents results obtained using a novel synthetic image dataset; and Section 6.4 draws the final conclusions.

6.2 Anchor-free 3D Object Detection Method

This section delves into the intricacies of the proposed methodology designed for the task of 3D object detection. The approach leverages cutting-edge anchor-free techniques and harnesses the power of machine learning algorithms to meticulously calculate essential spatial attributes of predicted objects, encompassing parameters such as depth, dimensions, and orientation, see 6.3. The exposition begins by elucidating the anchor-free techniques, a cornerstone of the methodology, which is instrumental in generating *heatmaps* for the predicted key-pairs representing the corners of the bounding boxes.

Within this framework, particular emphasis is placed on the techniques employed for the mapping of the predicted corners to the corresponding object categories. This step in the process is achieved through the application of Embedding methods, which facilitate the establishment of meaningful associations between the predicted corners and the expected object classes. Moreover, the methodology delves into the strategies for refining the positions of the predicted corners. This is accomplished by integrating Offsets, which serve as crucial adjustments to enhance the precision and accuracy of the predicted corner locations.

Central to the methodology are the architectural underpinnings that significantly contribute to its effectiveness. The discussion elaborates on the architectural components that form the bedrock of this approach. This includes detailed insights into techniques such as Cascade

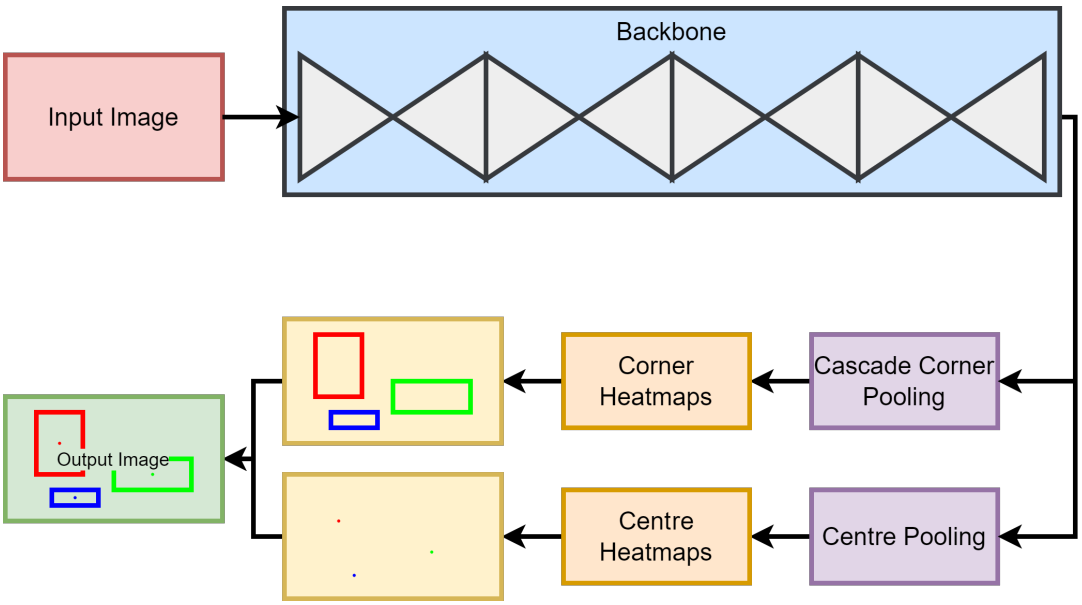


Figure 6.1: An abstract representation of the CenterNet architecture.

Corner Pooling, an innovative approach that plays a pivotal role in the spatial localization of object corners. Moreover, the methodology highlights the significance of Centre Pooling, a technique that further refines the positioning of the predicted corners and enhances their spatial accuracy.

Building on this foundation, the methodology proceeds to elucidate the process of 3D bounding box regression. This multifaceted process encompasses the estimation of depth, 3D dimensions, and object orientation attributes. The section expounds on the intricate techniques and machine learning algorithms employed to predict and refine these attributes, which are critical for achieving comprehensive and accurate 3D object detection.

The methodology presented in this study, as visually depicted in Figure 6.1, is fundamentally grounded in a anchor-free approach. Within this framework, keypoint descriptors play a fundamental role in representing crucial elements of object detection, encompassing essential entities such as the top-left, bottom-right, and centre points. This method places a strong emphasis on the precise determination of both corner and centre keypoints, a task of utmost significance, as it significantly influences the accuracy and reliability of object localisation

and recognition.

The crux of this approach lies in the meticulous discernment and characterisation of keypoint descriptors, which serve as distinctive markers of object structure and spatial attributes. These descriptors capture not only the spatial coordinates but also the semantic information associated with the objects in question. This careful attention to detail contributes to the heightened accuracy and reliability of tasks related to object localisation and recognition. These capabilities are of particular importance in various domains, including autonomous systems, surveillance, and computer vision research, where the ability to accurately and consistently identify and locate objects is a central requirement.

To ascertain the corner keypoints of the bounding box, the methodology incorporates an AI model equipped with a Cascade Corner Pooling module. This module plays a central role in the computational process, as it is responsible for calculating the maximum summed response along the boundaries of the feature map. Moreover, it extends its scope of analysis to encompass the internal directions within the feature map. This approach is a testament to the stability and robustness of the methodology, particularly in the face of feature-level noises and variations. The Cascade Corner Pooling module represents a sophisticated architectural innovation, optimised to provide a comprehensive and noise-tolerant mechanism for the precise localisation of corner keypoints.

Concomitantly, the methodology employs a dedicated AI model component for the determination of centre keypoints. This task is facilitated by the incorporation of a Centre Pooling module, a crucial element in the object detection process. The Centre Pooling module assumes the role of calculating the maximum summed response along both horizontal and vertical directions within the feature maps. This bidirectional analysis is instrumental in pinpointing the central keypoints of objects, a task that was deemed to be fundamental for more accurate object localisation and recognition.

The conceptual underpinnings of this methodology draw upon the work of Duan et al.

[40], whose innovative approach to object detection has informed and inspired this comprehensive framework. The utilisation of keypoint descriptors, coupled with the meticulous and data-driven methodologies for determining corner and centre keypoints, positions this methodology at the forefront of advancements in object detection techniques. It is characterised by its robustness, accuracy, and adaptability, making it well-suited for a diverse range of applications, including but not limited to autonomous systems, robotics, and computer vision research. The subsequent sections will provide a more in-depth exploration of the components, algorithms, and techniques that collectively drive the efficacy of this advanced methodology.

The determination of keypoints within this framework is achieved through a systematic process that entails the utilisation of Heatmaps derived from a set of feature maps generated by the Cascade Corner Pooling and Centre Pooling modules. These Heatmaps serve as a representation of the approximate positions of keypoint entities, which are classified into three distinct categories: top-left, bottom-right, and centre points. Notably, the Heatmaps are configured with a dimensionality of C channels, where C corresponds to the number of object classes under consideration. Additionally, they possess dimensions mirroring those of the input image, denoted as $H \times W$, with H denoting the image's height and W representing its width.

The critical facet in this methodology is the generation of Associative Embeddings, which serve as a means to establish a link between individual keypoints and their respective object classes. These Embeddings are computed by the AI network and obviate the need for a "ground-truth" number. Instead, their significance lies in the relative differences, which facilitate the grouping of object detections based on their associated Embeddings. Each detection generated by the AI network is accompanied by a numerical value, denoted as a "tag," which plays an instrumental role in the subsequent grouping of detections. The premise is that detections with similar tags should be effectively clustered together.

The output of the model necessitates certain adjustments to optimize the fit of the pre-

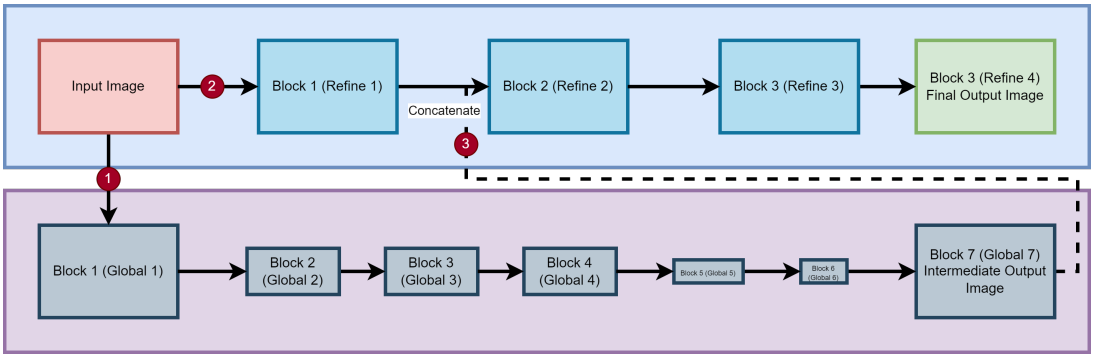


Figure 6.2: An abstract representation of the Multi-Scale Deep Network

dicted object to the actual object within the image. In response to this requirement, Offsets are introduced to facilitate these adjustments. Each Offset map serves as a spatial mapping of keypoint locations within the feature map space to their corresponding positions on the input image, conveyed in pixel coordinates. The application of Offsets represents an essential mechanism for fine-tuning the localisation of keypoints and enhancing the overall precision of object detection within the framework.

To achieve the successful detection of 3D objects, the methodology entails the prediction of centre keypoints, incorporating additional information encompassing depth, 3D dimensions, and orientation, as visually depicted in Figure 6.3. The depth component is a transformed output, derived from Eigen et al.'s approach [42], and it is presented as an additional scalar value associated with each centre keypoint.

The depth prediction mechanism comprises two principal modules, as illustrated in Figure 6.2. The initial component, the Global Coarse-Scale Network [42], takes the input image and endeavours to predict the depth of the entire scene at a global scale. This global-level prediction serves as the foundation for subsequent refinements. The refinement process is carried out by the Local Fine-Scale Network [42], which receives the output from the Global Coarse-Scale Network and fine-tunes the initial coarse predictions. This fine-tuning process is vital for aligning the depth predictions with local details, including the edges of objects and walls.

In assessing the quality of the depth predictions, the methodology employs a Scale Invariant Error metric. This metric computes the per-pixel differences between the predicted depth map and the ground truth. Notably, the calculations are performed in a logarithmic space, a choice made to address the potential issue of the average scale of the scene influencing the error measurements. This approach ensures a more robust and scale-invariant assessment of the quality of the depth predictions, ultimately contributing to the accuracy of the 3D object detection process.

The representation of a 3D bounding box in this context relies on three primary parameters, specifically the bounding box centre, dimensions, and orientation. The centre of the 3D bounding box is characterised by a set of three 3D-coordinates, denoted as x, y, and z. The dimensions of the 3D bounding box are governed by an additional triad of attributes, namely width, height, and length, measured in metres. These dimensions are directly regressed to their respective attributes through the application of a straightforward loss function. The orientation of the 3D bounding box is defined by another set of three attributes, encompassing azimuth, elevation, and roll angles in degrees.

The derivation of the three 3D-coordinates is realised through the utilisation of the MultiBin architecture [113]. In this architecture, each angle is treated as a distinct class, effectively framing the orientation prediction as a classification task. To account for the angular relationships between classes, the MultiBin architecture incorporates the computation of small offsets using trigonometric functions, specifically sine and cosine, applied to the angles. The outcome of this module encompasses three values for each class: the confidence associated with the class, the cosine difference of the angle, and the sine difference of the angle.

For the successful projection of a 3D bounding box onto a 2D image, the calculations necessitate the availability of a camera intrinsic matrix. This matrix plays a critical role in ensuring the accurate alignment of the 3D bounding box with the 2D image. Furthermore, to enhance the precision and reliability of the 3D bounding box, it is constrained through the utilisation of the 2D bounding box. This constraint mechanism contributes to the refinement

of the 3D bounding box and bolsters the accuracy of the overall object detection process.

6.2.1 Parameters

The proposed methodology underwent a series of experiments to comprehensively assess its performance. In the initial experiment, the model was subjected to 10 epochs of training with a batch size of 3, utilising synthetic data that spanned four distinct categories. During this training phase, the learning rate was set to 0.0001, and the optimisation process was facilitated by the Adam optimiser. Importantly, data augmentation techniques were not employed during this phase.

Subsequent to the initial experiments detailed above, additional experiments were carried out to probe the impact of extended training on the performance of the models. These final experiments didn't involve the utilisation of fine-tuned models that had undergone 10 epochs of training but were done from the beginning. The choice of batch size for these experiments varied from 3 to 6, contingent on the memory limitations of the GPU.

Furthermore, to expand the scope and depth of the analysis, extended experiments were conducted. These experiments employed the same fine-tuned models, which had undergone 100 epochs of training, and maintained similar batch size ranges as the final experiments. This extended training duration aimed to provide a more comprehensive assessment of the models' performance, encompassing a broader range of training scenarios and conditions.

6.3 Experimental Results

An extensive evaluation has been conducted using a synthetic dataset that was created using the Data Generation Tool mentioned earlier in the chapter 3.2. This dataset consists of approximately 3000 images per category of vehicles in different environments and weather conditions. An example of the dataset with predictions can be seen in Figure 6.3. The images in this dataset belong to four different categories, each allowing us to assess our model on specific properties: a) Camera, b) Light, c) Weather, d) Sensor. The categories were as-

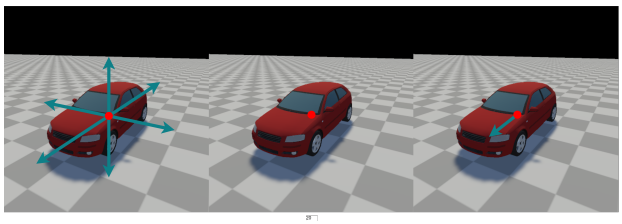


Figure 6.3: The network output for 3D object detection. From left to right: 3D dimensions (metres), depth (metres), orientation (degrees).

sembled in such a way to evaluate the model on specific properties and parameters of the virtual environment. Furthermore, each of the produced categories was split into Air and Ground subcategories where the Air subcategory contains only air vehicles, and the Ground one contains only ground vehicles.

The core underlying principles for evaluating 3D object detection are the same as for 2D object detection. The 2D object detection evaluation metrics rely on intersection-over-union. However, unlike 2D, 3D object detection produces 3D bounding boxes with spatial information. Consequently, intersection-over-union should take into account third dimension [4]. The rest of the evaluation proceed in the same manner as for evaluation of 2D bounding boxes. Furthermore, the proposed methodology produces 2D bounding boxes along with the main 3D bounding boxes where both could be utilised to analyse the results of the methodology.

The Real dataset was the KITTI dataset that is a widely used benchmark dataset for research in computer vision and autonomous driving. It stands for "Karlsruhe Institute of Technology and Toyota Technological Institute" and was created by researchers from these institutions. This dataset is commonly referenced in academic publications related to tasks such as object detection, tracking, 3D scene understanding, and more.

The primary objective behind its inception is to foster the advancement of algorithms and technologies relevant to autonomous vehicles. The dataset is characterised by a comprehensive collection of diverse data modalities, encompassing high-resolution camera im-

Table 6.1: Performance (mAP, %) of the End-to-End Super Resolution Object Detection framework on the Synthetic dataset in the three categories, where PM is the proposed framework, compared with the baseline models.

Category	Sub-category	FRRCNN	YOLOv3	RETINA	CenterNet
Air	Camera	35.24%	44.82%	44.79%	**61.04%**
	Light	40.66%	63.58%	61.25%	**69.95%**
	Weather	35.35%	39.00%	45.57%	**88.71%**
Ground	Camera	37.95%	76.02%	**78.81%**	74.66%
	Light	32.54%	38.52%	**77.12%**	63.82%
	Weather	37.07%	66.32%	**76.09%**	58.75%

Table 6.2: Performance (mAP, %) of the 3D Bounding Box Object Detection model on the KITTI dataset.

Class	AP, %
Car	87.85%
Pedestrian	60.85%
Cyclist	48.69%
mAP	
All classes	**65.80%**

ages, lidar point clouds, and calibration parameters. This dataset is used for a multitude of tasks, including but not limited to object detection, tracking, 3D scene understanding, and other pertinent applications. An added feature of the KITTI dataset is its provision of annotations for various object types, such as cars, pedestrians, and cyclists, thereby rendering it an invaluable resource for algorithm evaluation in these specific domains. Furthermore, the dataset encompasses a wide spectrum of real-world driving scenarios, variable weather conditions, and different times of day, thereby facilitating a comprehensive assessment of algorithm performance under diverse environmental conditions. It is imperative to note that while the KITTI dataset commands widespread respect in the research community, it does exhibit certain limitations, notably its relatively modest scale and the absence of data pertaining to certain object classes, such as motorcycles. The performance of the proposed framework for the aforementioned four categories using mAP could be seen in the table 6.1 and 6.2. Additionally, the confusion matrices can be observed in Figure 6.5.

The results of the table 6.1 seemed to show competitive results comparing it with the

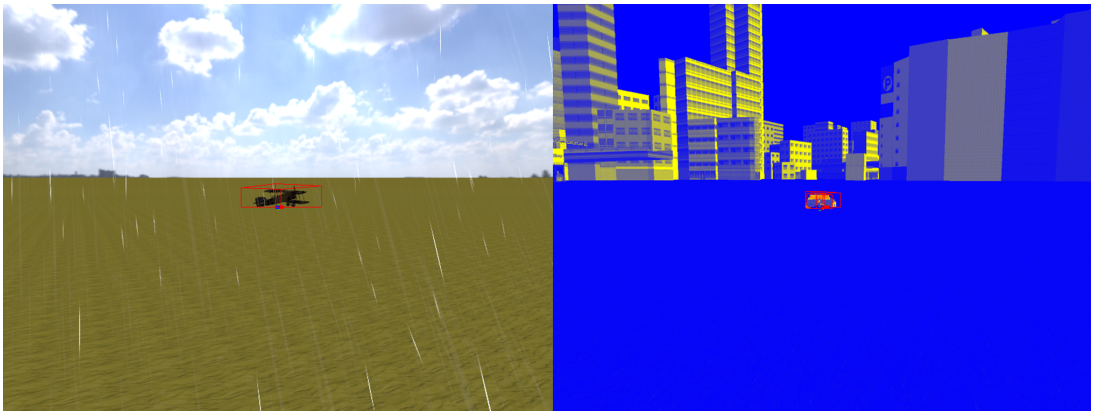


Figure 6.4: Example of 3D bounding box predictions.

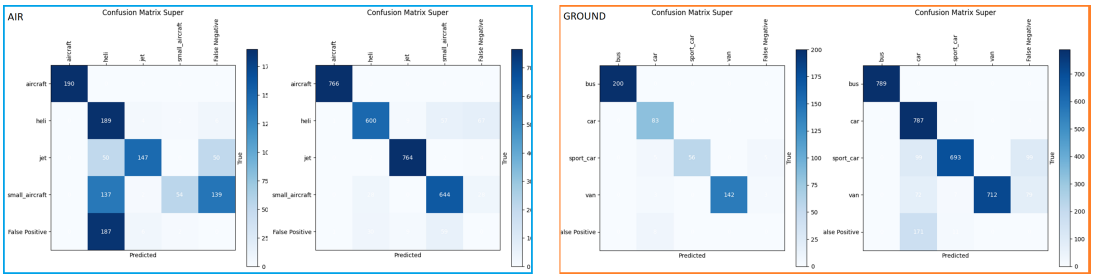


Figure 6.5: Two comparisons between initial fine-tuning and extensive fine-tuning of Air and Ground categories. From left to right: the blue box is grouping Air category (initial and extensive correspondingly), the orange box is grouping Ground category (initial and extensive correspondingly).

results of the table 6.2. Table 6.2 demonstrating state of the art performance with a high precision in the "Car" category. As the real dataset was is unbalanced towards the "Car" category the "Pedestrian" and "Cyclist" categories demonstrate worse results. The "Cyclist" category is the smallest category in the real dataset and consequently performed the worst. Additionally, visual similarity between "Pedestrians" and "Cyclists" somewhat contributes towards decreased results.

Table 6.1 displays the results obtained by CenterNet on the Air and Ground datasets. Figure 6.1 shows the confusion matrices of the CenterNet model between the Air and Ground datasets. The CenterNet model produced competitive results for the Air sub-category. When comparing the Air and Ground results the ground vehicles were detected with significantly higher accuracy. The most likely cause of such behaviour is related with the attributes of the

Air and Ground datasets. The main difference between the subsets is the supported distances between the camera and the object. Furthermore, the objects themselves are more diverse in terms of size and shape.

As the results in table 6.1 show, the category with the highest results was the Weather sub-category under the Air category. Nevertheless, in both sub-categories performance was relatively high taking into consideration the number of epochs used for the training. The Weather sub-category includes images with parameters such as rain and fog, as well as different strengths of wind. The Light sub-category includes images with similar parameters and visual features, for example multiple images of the objects at certain angles. According to table 6.1, the Light subset results ended up around the middle of the output performance table. Comparing Air and Ground results, the model performed well with a general trend of increased accuracy for the Air sub-category. Examples of predictions are illustrated in Figure 6.3 for the CenterNet models. Analysing table 6.1, the results produced by the model are competitive and in half of the cases are better than the baseline.

In the subsequent stage of experimentation, the machine learning architecture underwent fine-tuning, involving a rigorous training regimen spanning 100 epochs. This fine-tuning process was executed separately for each of the primary categories, namely "Air" and "Ground," as well as for each of the sub-categories, namely "Camera," "Light," "Weather," and "Sensor." For the task of extended evaluation, two distinct sets of data were amalgamated from the comprehensive Synthetic dataset, utilising identical parameters. The first dataset encompassed images captured in the "Forest" and "Grass" scenes, while the second dataset comprised images originating from the "City" and "Desert" scenes. Notably, each of these datasets comprised approximately 3000 images for each sub-category.

The allocation of images into training, validation, and testing subsets adhered to a specific protocol. The training subset exclusively consisted of images derived from the "Forest" and "Grass" scenes, encompassing 100% of the images within the first dataset. The validation subset, on the other hand, was composed of a tenth of all images from the "City" and "Desert"

scenes, representing 10% of the images in the second dataset. The remaining images from the "City" and "Desert" scenes were collectively grouped into the testing subset, thereby establishing a structured and well-defined partitioning of the data for the purposes of robust evaluation. The results of the experiments demonstrate the performance of the new machine learning architecture via a mean average precision in percentages. In comparison with the final experiments, the extended experiments, depicted, demonstrated an overall performance increase.

Regarding the "Air" category, a noticeable gain could be observed in the Camera sub-category. On the other hand, the "Light" sub-category has minor deviations. However, the results of the "Light" sub-category were already high, therefore dramatic changes were not expected. In the "Weather" sub-category similar to the "Camera" sub-category, the improvement of the CenterNet model is noticeable. Although, the type of images in the "Sensor" sub-category differs from the types of images in the other sub-categories, all the models reached better performance metrics.

Switching to the "Ground" category, covering the same four sub-categories: "Camera", "Light", "Weather", and "Sensor".The "Camera" sub-category had high results in the final experiments, the extended experiments produced slightly deviating results. The "Light" sub-category followed similar pattern where the final experiments produced moderately better results in the extended experiments. The CenterNet model had some room for improvement therefore the positive change was more noticeable. The "Weather" sub-category continued the trend. The CenterNet model showed considerable increase. The "Sensor" sub-category was akin to the "Sensor" sub-category under the "Air" category where concordant improvement was seen. The results suggest that the CenterNet model benefited from the extensive training and gained an improvement.

6.4 Conclusion

This chapter amalgamates a comprehensive overview of existing methodologies and approaches within the scope of scene analysis, with a specific emphasis on their applicability in immersive environments. The research presented herein delves into an in-depth analysis of a 3D object detection model from the vantage point of the augmented reality domain. The architectural framework comprises a diverse set of components, each meticulously designed to tackle various intricacies related to the estimation of keypoints, the conversion of keypoints to 2D bounding boxes, and the inference of crucial spatial information. This information encompasses depth, 3D dimensions measured in metres, as well as orientation, encompassing azimuth, elevation, and roll angles. The collective contributions of these components culminate in a model that exhibits proficiency in the projection of 3D bounding boxes onto a 2D image. The presented methodology found practical application in the domains of object recognition and scene comprehension. An evaluation of its performance was carried out using a synthetic dataset intentionally designed to introduce diverse environmental conditions that would influence image quality as well as using a dataset consisting out of images from real environments. The results outlined the detailed analysis of the evaluation conducted on the CenterNet model. In summary, the CenterNet model exhibited a consistent level of stability and predictability, denoting its ability to deliver reliable results across varying environmental conditions.

To empirically evaluate the efficacy of the proposed architecture, a comprehensive testing regimen was conducted, utilising a synthetic dataset in a comparative study. The outcomes of this assessment reveal a model that not only delivers competitive performance but also demonstrates stability, particularly when tasked with the detection of distant objects. The evaluation and analysis of the proposed model were undertaken under diverse environmental conditions and with varying camera sensors, thereby establishing its versatility and robustness. Furthermore, to augment the comprehensiveness of the study, a novel and well-balanced synthetic dataset was meticulously curated. This dataset encompasses annotated

data spanning a multitude of objects and environmental scenarios, providing a rich resource for subsequent rounds of assessment, experimentation, and refinement.

Chapter 7

Conclusions

Contents

7.1 Summary of Contributions

In this thesis, a series of contributions aimed at enhancing scene understanding in augmented reality applications were presented. The first contribution introduces a novel modular deep learning framework designed for AR systems. This framework consists out of modular components tailored to handle a notorious aspect of scene analysis such as object detection. By providing a flexible and customisable architecture, our framework enables seamless end-to-end training for the purpose of AR systems, facilitating real-time scene understanding and interaction.

The second contribution delves into an evaluation of environmental conditions on object detection using oriented bounding boxes for AR applications. Here, the research investigates how factors such as lighting conditions, occlusions, and viewpoints influence object detection performance. By leveraging OBBs, the aim of the evaluation was to improve object detection accuracy under challenging environmental conditions. Our extensive experimentation

and analysis provide valuable insights into optimising AR systems for real-world scenarios, shedding light on the intricate interplay between environmental factors and object detection algorithms.

The third contribution focuses on data augmentation techniques purposed to address the main problem of few-shot learning for the object detection task. In scenarios where labelled data is scarce, our approach leverages various augmentation strategies, including geometric transformations, colour jittering, and synthetic data generation, to enhance model generalisation and robustness. Through empirical evaluation and comparative analysis, the results demonstrate the effectiveness of our data augmentation strategy in improving object detection performance, particularly in settings with limited labelled data.

Finally, the fourth contribution presents an in-depth analysis of 3D object detection for AR applications using a synthetic dataset. By synthesising realistic scenes with annotated 3D objects, a benchmark dataset for evaluating the performance of 3D object detection algorithms in AR environments was provided. Through quantitative evaluation and qualitative analysis, the strengths and limitations of existing 3D object detection approaches were assessed, offering valuable insights for future research and development in this domain. Overall, our contributions collectively advance the state-of-the-art in scene understanding for augmented reality applications. By developing a modular deep learning framework, evaluating environmental influences on object detection, exploring data augmentation techniques for few-shot learning, and analysing 3D object detection algorithms using synthetic datasets, a deeper understanding of object detection in AR contexts is contributed, and valuable guidance for optimising AR systems in various environmental conditions and application scenarios is provided.

7.2 Ethical Considerations

Ethical considerations are paramount in the development and implementation of object detection algorithms for scene analysis in smart devices and augmented reality applications.

These considerations extend to various aspects of technology deployment, encompassing privacy [65], fairness [112], security [103], data governance [1], social impacts [169], and human rights [50]. One of the primary ethical concerns revolves around privacy, as object detection algorithms have the potential to infringe upon individuals' privacy rights ([64], [44], [129]). For instance, surveillance systems equipped with object detection capabilities may inadvertently capture and process sensitive personal information, raising concerns about unauthorised monitoring and surveillance without consent. To address these privacy concerns, it is essential to implement privacy-preserving measures such as data unionisation, encryption, and access controls to safeguard individuals' privacy rights. Another critical ethical consideration is bias and fairness in object detection algorithms [112]. Biased algorithms may result in disparities in detection accuracy across different demographic groups, perpetuating societal inequalities and discrimination. This issue is particularly relevant in applications such as surveillance and law enforcement, where biased algorithms can lead to unfair treatment and wrongful judgements [150]. Mitigating bias in dataset collection, algorithm design, and evaluation is crucial to ensure fair and equitable outcomes for all individuals and communities.

Security risks pose another ethical challenge in the deployment of object detection algorithms [103]. Adversarial attacks, where malicious actors manipulate input data to deceive the algorithm or cause erroneous detection, can compromise the reliability and integrity of object detection systems [152]. In critical applications such as autonomous driving and surveillance, adversarial attacks can pose serious safety hazards and security breaches. Robustness testing and adversarial training techniques are essential to enhance the resilience of object detection algorithms against such attacks and mitigate potential security risks. Data governance and ownership are also central ethical considerations in the development of object detection algorithms [1]. The collection, storage, and use of data for training algorithms raise concerns about data privacy, ownership, and consent. Transparent data governance policies and practices are essential to ensure accountability and compliance with data protection regulations. Clear guidelines and consent mechanisms for data usage are necessary to

address ethical concerns related to data governance and ownership and uphold individuals' rights to control their personal data.

Moreover, object detection algorithms have the potential to influence societal norms and behaviours, particularly in public spaces and community settings [169]. The deployment of surveillance systems equipped with facial recognition capabilities, for example, may impact social cohesion and community relations, leading to feelings of surveillance and distrust among individuals. Ethical considerations should guide the responsible deployment and use of object detection technology to minimise negative social impacts and promote public trust [169]. Lastly, the use of object detection algorithms in surveillance and law enforcement contexts raises human rights concerns [50]. Excessive or indiscriminate surveillance may infringe upon individuals' fundamental rights and liberties, including the right to privacy, freedom of expression, and freedom of movement. Clear legal frameworks and oversight mechanisms are necessary to safeguard human rights protections and ensure accountability and transparency in the deployment of object detection technology in sensitive contexts.

Finally, it is crucial to discuss how the proposed framework aligns with the latest regulatory frameworks, such as the UK AI Act and the EU AI Act, which focus on ensuring ethical AI development. Both regulations emphasise key principles such as transparency, fairness, accountability, and the safety of AI systems, especially those that interact with real-world environments. For instance, the EU AI Act specifically categorises AI applications based on their risk levels, imposing stricter requirements on high-risk systems such as those used in augmented reality (AR). By demonstrating how the proposed system adheres to these standards, the work will not only highlight its technical merits but also its commitment to responsible AI deployment, addressing concerns about user safety and ethical implications. These regulations require AI systems to be explainable, traceable, and robust to risks, aspects that are integral when designing AR applications that may directly impact users' perceptions and decisions in dynamic, real-world contexts [37].

Furthermore, aligning with these regulatory frameworks ensures that the proposed AR

system is built with accountability and transparency in mind. For example, ensuring that the system can provide clear information about how data is processed, how decisions are made, and how the model's predictions are derived will be essential to meet the obligations set forth by these acts. Moreover, compliance with ethical guidelines will strengthen the trust of stakeholders, including end-users and regulatory bodies, in the system's safety and fairness. As these regulations increasingly influence AI design and deployment, reflecting their principles in the research ensures that the framework remains relevant not only from a technological standpoint but also from a societal perspective, addressing concerns regarding bias, privacy, and the potential misuse of AI [135]. By demonstrating how the AR framework complies with these evolving legal standards, the research will ensure its readiness for real-world implementation while upholding the values of ethical AI development.

7.3 Directions for Future Research

Looking ahead, the future of object detection algorithms in scene analysis for smart devices and augmented reality applications presents several promising directions. One significant area for advancement involves enhancing the robustness and generalisation capabilities of these algorithms. Future efforts should prioritise developing models capable of effectively handling diverse environmental conditions, occlusions, and variations in scale and viewpoint. Robustness against adversarial attacks and domain shifts will be essential for ensuring reliable performance in real-world scenarios. Additionally, integrating multiple modalities such as visual, depth, and semantic information holds promise for improving object detection accuracy and robustness. Future research should explore innovative approaches for multi-modal fusion, leveraging the complementary strengths of different sensor modalities to enhance scene understanding and object localisation in complex environments.

Efficiency and real-time processing are also critical considerations for future developments in object detection. As the demand for real-time applications grows, there is a need for more efficient and computationally lightweight algorithms. Future research should focus on

developing efficient architectures and optimisation techniques to enable real-time processing on resource-constrained devices such as smartphones and AR glasses without compromising detection accuracy. Continuous learning and adaptation capabilities are another key area for future exploration in object detection algorithms. Algorithms should be capable of adapting to evolving environments and scenarios through incremental learning, domain adaptation, and lifelong learning techniques. This adaptability will enable algorithms to improve over time through exposure to new data and experiences.

Ethical considerations must remain a priority in the future deployment of object detection technology. Researchers and developers should continue to prioritise ethical design principles, fairness, transparency, and accountability in the development and deployment of object detection algorithms. By doing so, potential harms can be minimised, and societal benefits maximised. Tailoring object detection algorithms to specific domains and application contexts is another avenue for future advancement. Customising algorithms to address domain-specific challenges and requirements can significantly enhance their performance in applications such as autonomous driving, healthcare, retail, and industrial automation.

Lastly, fostering interdisciplinary collaboration across various disciplines including computer vision, machine learning, robotics, human-computer interaction, and ethics will be crucial for advancing the field of object detection. By leveraging diverse expertise and perspectives, interdisciplinary collaborations can address complex challenges and drive innovation in scene analysis for smart devices and AR applications. Through continued innovation, collaboration, and ethical considerations, object detection algorithms will play a pivotal role in enabling intelligent and immersive AR experiences, enhancing human-machine interaction, and addressing real-world challenges across diverse domains.

Bibliography

[1] R. Abraham, J. Schneider, and J. Vom Brocke. Data governance: A conceptual framework, structured review, and research agenda. *International journal of information management*, 49:424–438, 2019. 151

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels. *EPFL*, 2010. 119

[3] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 42, 62

[4] M. G. Adam, M. Piccolrovazzi, S. Eger, and E. Steinbach. Bounding box disparity: 3d metrics for object detection with full degree of freedom. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 1491–1495. IEEE, 2022. 142

[5] S. Adam, B. Sergey, B. Matthew, W. Daan, and L. Timothy. One-shot learning with memory-augmented neural networks. In *Proceedings of the 33nd International Conference on Machine Learning, New York, NY, USA*, pages 19–24, 2016. 44

[6] M. A. Al-Antari, M. A. Al-Masni, and T.-S. Kim. Deep learning computer-aided diagnosis for breast lesion in digital mammogram. *Deep Learning in Medical Image Analysis: Challenges and Applications*, pages 59–72, 2020. 25

[7] R. Anderson, J. Toledo, and H. ElAarag. Feasibility study on the utilization of microsoft hololens to increase driving conditions awareness. In *2019 SoutheastCon*, pages 1–8. IEEE, 2019. 75, 134

[8] R. T. Azuma. A survey of augmented reality. *Presence: teleoperators & virtual environments*, 6(4):355–385, 1997. 25

[9] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem. Sod-mtgan: Small object detection via multi-task generative adversarial network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 206–221, 2018. 38

[10] A. Bansal, K. Sikka, G. Sharma, R. Chellappa, and A. Divakaran. Zero-shot object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 384–400, 2018. 43

[11] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 404–417. Springer, 2006. 14

[12] S. Beery, G. Wu, V. Rathod, R. Votel, and J. Huang. Context r-cnn: Long term temporal context for per-camera object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13075–13085, 2020. 23

[13] S. Bell-Kligler, A. Shocher, and M. Irani. Blind super-resolution kernel estimation using an internal-gan. *Advances in Neural Information Processing Systems*, 32, 2019. 89

[14] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 93

[15] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998. 26

[16] J. Cao, H. Cholakkal, R. M. Anwer, F. S. Khan, Y. Pang, and L. Shao. D2det: Towards high quality object detection and instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11485–11494, 2020. 34

[17] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. 38, 44

[18] R. Cassani, M.-A. Moinnereau, and T. H. Falk. A neurophysiological sensor-equipped head-mounted display for instrumental qoe assessment of immersive multimedia. In *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6, 2018. 9

[19] H. Chen, L. Hou, G. K. Zhang, and S. Wu. Using context-guided data augmentation, lightweight cnn, and proximity detection techniques to improve site safety monitoring under occlusion conditions. *Safety science*, 158:105958, 2023. 11, 52, 53

[20] Y. Chen, X. Yuan, R. Wu, J. Wang, Q. Hou, and M.-M. Cheng. Yolo-ms: rethinking multi-scale representation learning for real-time object detection. *arXiv preprint arXiv:2308.05480*, 2023. 42

[21] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. 42

[22] M. Chu, Y. Xie, L. Leal-Taixé, and N. Thuerey. Temporally coherent gans for video super-resolution (tecogan). *arXiv preprint arXiv:1811.09393*, 1(2):3, 2018. 75

[23] B. Courty, V. Schmidt, S. Luccioni, Goyal-Kamal, MarionCoutarel, B. Feld, J. Lecourt, LiamConnell, A. Saboni, Inimaz, supatomic, M. Léval, L. Blanche, A. Cruveiller, ouminasara, F. Zhao, A. Joshi, A. Bogroff, H. de Lavoreille, N. Laskaris, E. Abati, D. Blank, Z. Wang, A. Catovic, M. Alencon, M. Stęchły, C. Bauer, Lucas-Otavio, JPW, and MinervaBooks. mlco2/codecarbon: v2.4.1, May 2024. 124

[24] A. B. Craig. *Understanding augmented reality: Concepts and applications*. Newnes, 2013. 10, 11

[25] C. Creed, M. Al-Kalbani, A. Theil, S. Sarcar, and I. Williams. Inclusive ar/vr: accessibility barriers for immersive technologies. *Universal Access in the Information Society*, 23(1):59–73, 2024. 7

[26] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 113–123, 2019. 11, 45, 53, 57, 58

[27] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 58, 123

[28] P. Cunningham, M. Cord, and S. J. Delany. Supervised learning. In *Machine learning techniques for multimedia: case studies on organization and retrieval*, pages 21–49. Springer, 2008. 76

[29] D. Dai, Y. Wang, Y. Chen, and L. Van Gool. Is image super-resolution helpful for other vision tasks? In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016. 88

[30] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, 29, 2016. 34

[31] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005. 22, 26

[32] M. R. Desselle, R. A. Brown, A. R. James, M. J. Midwinter, S. K. Powell, and M. A. Woodruff. Augmented and virtual reality in surgery. *Computing in Science & Engineering*, 22(3):18–26, 2020. 4

[33] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 45, 47, 48, 51

[34] V. Di Pasquale, V. De Simone, C. Franciosi, P. Morra, and S. Miranda. Augmented and virtual reality to support corrective and preventive actions in maintenance: a framework proposal. *Procedia Computer Science*, 232:1879–1889, 2024. 116

[35] N. Dimitropoulos, T. Togias, G. Michalos, and S. Makris. Operator support in human–robot collaborative environments using ai enhanced wearable devices. *Procedia Cirp*, 97:464–469, 2021. 75, 134

[36] A. Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 43

[37] L. DOWN and I. ACT. Proposal for a regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts. 2021. 152

[38] M. Drobnitzky, J. Friederich, B. Egger, and P. Zschech. Survey and systematization of 3d object detection models and methods. *The Visual Computer*, 40(3):1867–1913, 2024. 133

[39] D. Du, P. Zhu, L. Wen, X. Bian, H. Lin, Q. Hu, T. Peng, J. Zheng, X. Wang, Y. Zhang, et al. Visdrone-det2019: The vision meets drone object detection in image challenge results. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019. 26, 77, 106

[40] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6569–6578, 2019. 134, 138

[41] C. E. Duchon. Lanczos filtering in one and two dimensions. *Journal of Applied Meteorology and Climatology*, 18(8):1016–1022, 1979. 88

[42] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 134, 139

[43] S. Elfwing, E. Uchibe, and K. Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018. 122

[44] V. Eubanks. *Automating inequality: How high-tech tools profile, police, and punish the poor.* St. Martin's Press, 2018. 151

[45] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010. 64, 65, 93

[46] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 36

[47] C.-M. Feng, K. Yu, Y. Liu, S. Khan, and W. Zuo. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2704–2714, 2023. 12, 61, 62

[48] M. Figurnov, M. D. Collins, Y. Zhu, L. Zhang, J. Huang, D. Vetrov, and R. Salakhutdinov. Spatially adaptive computation time for residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1039–1048, 2017. 23

[49] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017. 44, 45

[50] L. Floridi, J. Cowls, M. Beltrametti, R. Chatila, P. Chazerand, V. Dignum, C. Luetge, R. Madelin, U. Pagallo, F. Rossi, et al. Ai4people—an ethical framework for a good ai society: opportunities, risks, principles, and recommendations. *Minds and machines*, 28:689–707, 2018. 151, 152

[51] K. A. Frenkel. An interview with ivan sutherland. *Communications of the ACM*, 32(6):712–714, 1989. 7

[52] Z. Ge, S. Liu, Z. Li, O. Yoshie, and J. Sun. Ota: Optimal transport assignment for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 303–312, 2021. 92

[53] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 90, 91, 92, 93, 102

[54] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 66

[55] Z. Gevorgyan. Siou loss: More powerful learning for bounding box regression. *arXiv preprint arXiv:2205.12740*, 2022. 92

[56] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 22, 23, 27, 32, 34, 85, 93

[57] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 27, 29

[58] D. L. Gomes Jr, A. C. de Paiva, A. C. Silva, G. Braz Jr, J. D. S. de Almeida, A. S. de Araújo, and M. Gattas. Augmented visualization using homomorphic filtering and haar-based natural markers for power systems substations. *Computers in Industry*, 97:67–75, 2018. 75, 134

[59] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 38

[60] J. Gu, H. Lu, W. Zuo, and C. Dong. Blind super-resolution with iterative kernel correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1604–1613, 2019. 89

[61] B. Gui-wei and Z. Guo-bao. Research on the visual impact of digital media art based on augmented reality technology. *Computer Aided Design and Applications (CADANDA)*, 2024. 74

[62] Y. Guo, S. Peeta, S. Agrawal, and I. Benedyk. Impacts of pokémon go on route and mode choice decisions: exploring the potential for integrating augmented reality, gamification, and social components in mobile apps to influence travel decisions. *Transportation*, pages 1–50, 2022. 3

[63] R. Gupta, A. Sharma, and A. Kumar. Super-resolution using gans for medical imaging. *Procedia Computer Science*, 173:28–35, 2020. 75

[64] S. Gürses, C. Troncoso, and C. Diaz. Engineering privacy by design. *Computers, Privacy & Data Protection*, 14(3):25, 2011. 151

[65] T. Hagendorff. The ethics of ai ethics: An evaluation of guidelines. *Minds and machines*, 30(1):99–120, 2020. 151

[66] S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015. 41

[67] M. Haris, G. Shakhnarovich, and N. Ukita. Task-driven super resolution: Object detection in low-resolution images. In *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part V 28*, pages 387–395. Springer, 2021. 88

[68] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 27, 30, 34

[69] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015. 30

[70] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 85, 92

[71] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 558–567, 2019. 51

[72] M. L. Heilig. Stereoscopic-television apparatus for individual use, 10 1960. U.S. Patent. 7

[73] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019. 61

[74] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 23

[75] Y. Huang, S. Li, L. Wang, T. Tan, et al. Unfolding the alternating optimization for blind super resolution. *Advances in Neural Information Processing Systems*, 33:5632–5643, 2020. 88, 89, 92

[76] Y. Huang, L. Shao, and A. F. Frangi. Simultaneous super-resolution and cross-modality synthesis of 3d medical images using weakly-supervised joint convolutional sparse coding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6070–6079, 2017. 88

[77] IKEA. Ikea place app launched to help people virtually place furniture at home. Online, 2017. Accessed on March 25, 2024. 3

[78] A. Inc. Apple visionpro. `https://www.apple.com/`, 2023. 7

[79] P. T. Jackson, A. A. Abarghouei, S. Bonner, T. P. Breckon, and B. Obara. Style augmentation: data augmentation via style randomization. In *CVPR workshops*, volume 6, pages 10–11, 2019. 45, 53, 60, 61

[80] G. Jha, L. s. Sharma, and S. Gupta. Future of augmented reality in healthcare department. In *Proceedings of Second International Conference on Computing, Communications, and Cyber-Security: IC4S 2020*, pages 667–678. Springer, 2021. 74

[81] G. Jocher, A. Chaurasia, and J. Qiu. Ultralytics YOLO, Jan. 2023. 41

[82] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE transactions on acoustics, speech, and signal processing*, 29(6):1153–1160, 1981. 88

[83] J. C. Kim, S. Saguna, C. Åhlund, and K. Mitra. Augmented reality-assisted healthcare system for caregivers in smart regions. In *2021 IEEE International Smart Cities Conference (ISC2)*, pages 1–7. IEEE, 2021. 25

[84] Y. Kim, A. S. Uddin, and S.-H. Bae. Local augment: Utilizing local bias property of convolutional neural networks for data augmentation. *IEEE Access*, 9:15191–15199, 2021. 55

[85] J. Konecnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 8, 2016. 24

[86] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 26, 50

[87] T. Kumar, M. Turab, K. Raj, A. Mileo, R. Brennan, and M. Bendechache. Advanced data augmentation approaches: A comprehensive survey and future directions. *arXiv preprint arXiv:2301.02830*, 2023. 45

[88] F. Lamberti, F. Manuri, A. Sanna, G. Paravati, P. Pezzolla, and P. Montuschi. Challenges, opportunities, and future trends of emerging techniques for augmented reality-based maintenance. *IEEE Transactions on Emerging Topics in Computing*, 2(4):411–421, 2014. 4

[89] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018. 37, 134

[90] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 121

[91] V. Li, G. Amponis, J.-C. Nebel, V. Argyriou, T. Lagkas, S. Ouzounidis, and P. Sarigiannidis. Super resolution for augmented reality applications. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–6. IEEE, 2022. 38

[92] V. Li, G. Amponis, J.-C. Nebel, V. Argyriou, T. Lagkas, and P. Sarigiannidis. Object recognition for augmented reality applications. *Azerbaijan Journal of High Performance Computing*, 4(1):15–28, 2022. 21

[93] V. Li, B. Villarini, J.-C. Nebel, T. Lagkas, P. Sarigiannidis, and V. Argyriou. Evaluation of environmental conditions on object detection using oriented bounding boxes for ar applications. In *2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*, pages 309–316. IEEE, 2023. 101

[94] V. Li, B. Villarini, J.-C. Nebel, and A. Vasileios. A modular deep learning framework for scene understanding in augmented reality applications. In *2023 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pages 45–51. IEEE, 2023. 91

[95] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Light-head r-cnn: In defense of two-stage object detector. *arXiv preprint arXiv:1711.07264*, 2017. 34

[96] W. Liang, Y. Liang, and J. Jia. Miamix: Enhancing image classification through a multi-stage augmented mixed sample data augmentation method. *Processes*, 11(12):3284, 2023. 11, 55, 56

[97] C. Lin, A. Koval, S. Tishchenko, A. Gabdulkhakov, U. Tin, G. P. Solis, and V. L. Katanaev. Double suppression of the gα protein activity by rgs proteins. *Molecular cell*, 53(4):663–671, 2014. 69, 71

[98] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 34, 85

[99] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 36, 69, 84, 85, 93, 122

[100] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 64, 71, 85

[101] T. Lindeberg. Scale invariant feature transform. *Digitala Vetenskapliga Arkivet*, 2012. 14, 22

[102] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016. 36

[103] O. A. Lottu, B. S. Jacks, O. A. Ajala, and E. S. Okafo. Towards a conceptual framework for ethical ai development in it systems. *World Journal of Advanced Research and Reviews*, 21(3):408–415, 2024. 151

[104] R. Masoni, F. Ferrise, M. Bordegoni, M. Gattullo, A. E. Uva, M. Fiorentino, E. Carrabba, and M. Di Donato. Supporting remote maintenance in industry 4.0 through augmented reality. *Procedia manufacturing*, 11:1296–1302, 2017. 25

[105] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943. 22

[106] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021. 71

[107] J. E. Melzer and K. Moffitt. *Head mounted displays*. CRC Press, 1997. 7

[108] J. Meneely, C. Donnelly, C. Lavelle, T. Martin, B. Sloan, and S. Weir. Reconstruction of the ballintaggart court tomb using 3d scanning, 3d printing, and augmented reality (ar). In *3D Imaging of the Environment*, pages 190–199. CRC Press, 2024. 74

[109] Meta. Meta Quest. `https://www.meta.com/quest`, 2024. Meta Quest is an all-in-one standalone VR headset that offers users a wireless and untethered virtual reality experience. Unlike traditional VR headsets, which typically require a powerful gaming PC or console for operation, the Quest is equipped with all the necessary hardware to run VR applications and games independently. 7, 9

[110] T. Michaeli and M. Irani. Nonparametric blind super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–952, 2013. 89

[111] Microsoft. Microsoft HoloLens. `https://www.microsoft.com/en-us/hololens`, 2022. Microsoft HoloLens is a wearable holographic computer that enables users to interact with digital content and holograms overlaid onto the physical world around them. It features sensors, advanced optics, and a processing unit that allows it to track the user's movements and spatially map the environment in real-time. This enables HoloLens to display holographic images and information seamlessly integrated with the user's surroundings. 7, 8, 99

[112] B. D. Mittelstadt, P. Allo, M. Taddeo, S. Wachter, and L. Floridi. The ethics of algorithms: Mapping the debate. *Big Data & Society*, 3(2):2053951716679679, 2016. 151

[113] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka. 3d bounding box estimation using deep learning and geometry. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7074–7082, 2017. 135, 140

[114] S. P. Mudunuri and S. Biswas. Low resolution face recognition across variations in pose and illumination. *IEEE transactions on pattern analysis and machine intelligence*, 38(5):1034–1040, 2015. 88

[115] L. Muñoz-Saavedra, L. Miró-Amarante, and M. Domínguez-Morales. Augmented and virtual reality evolution and future tendency. *Applied sciences*, 10(1):322, 2020. 3

[116] J. Nam, H. Chung, H. Lee, et al. A new terrain in hci: Emotion recognition interface using biometric data for an immersive vr experience. *arXiv preprint arXiv:1912.01177*, 2019. 9

[117] M. Negnevitsky. *Artificial intelligence: a guide to intelligent systems*. Pearson education, 2005. 22

[118] H. Nindiasari, M. F. Pranata, S. Sukirwan, S. Sugiman, M. Fathurrohman, A. Ruhimat, Y. Yuhana, et al. The use of augmented reality to improve students' geometry concept problem-solving skills through the steam approach. *Infinity Journal*, 13(1):119–138, 2024. 74

[119] R. Padilla, S. L. Netto, and E. A. Da Silva. A survey on performance metrics for object-detection algorithms. In *2020 international conference on systems, signals and image processing (IWSSIP)*, pages 237–242. IEEE, 2020. 68

[120] R. Palmarini, J. A. Erkoyuncu, R. Roy, and H. Torabmostaedi. A systematic review of augmented reality applications in maintenance. *Robotics and Computer-Integrated Manufacturing*, 49:215–228, 2018. 4

[121] M. I. Pavel, S. Y. Tan, and A. Abdullah. Vision-based autonomous vehicle systems based on deep learning: A systematic literature review. *Applied Sciences*, 12(14):6831, 2022. 75

[122] C. U. Perez Malla, M. d. C. Valdes Hernandez, M. F. Rachmadi, and T. Komura. Evaluation of enhanced learning techniques for segmenting ischaemic stroke lesions in brain magnetic resonance perfusion images using a convolutional neural network scheme. *Frontiers in neuroinformatics*, 13:33, 2019. 69

[123] J.-M. Perez-Rua, X. Zhu, T. M. Hospedales, and T. Xiang. Incremental few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 38

[124] C. Portalés, J. L. Lerma, and C. Pérez. Photogrammetry and augmented reality for cultural heritage applications. *The Photogrammetric Record*, 24(128):316–331, 2009. 74

[125] P. Purkait, C. Zhao, and C. Zach. Spp-net: Deep absolute pose regression with synthetic views. *arXiv preprint arXiv:1712.03452*, 2017. 32

[126] H. Qing, R. Li, C. Pan, and O. Gao. Remote sensing image object detection based on oriented bounding box and yolov5. In *2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, volume 10, pages 657–661. IEEE, 2022. 105, 122

[127] A. Radford. Improving language understanding by generative pre-training. 2018. 42

[128] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 62

[129] I. D. Raji and J. Buolamwini. Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial ai products. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 429–435, 2019. 151

[130] RangeKing. Brief summary of yolov8 model structure. GitHub, 2023. `https://github.com/RangeKing`. 14, 121

[131] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 22, 23, 27, 34, 35, 69, 87, 93

[132] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 22, 35, 87, 101

[133] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 22, 27, 35, 64, 84, 85, 86, 90, 91, 93, 101

[134] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 23, 33, 84, 85, 93

[135] H. Roberts, A. Babuta, J. Morley, C. Thomas, M. Taddeo, and L. Floridi. Artificial intelligence regulation in the united kingdom: a path to good governance and global leadership? *Internet Policy Review*, 2023. 153

[136] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 62

[137] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *International conference on machine learning*, pages 2152–2161. PMLR, 2015. 117

[138] S. K. Roy, M. Harandi, R. Nock, and R. Hartley. Siamese networks: The tale of two manifolds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3046–3055, 2019. 117

[139] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 64, 66, 71

[140] I. H. Sarker. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2(6):420, 2021. 25

[141] A. Sauer, N. Savinov, and A. Geiger. Conditional affordance learning for driving in urban environments. In *Conference on Robot Learning*, pages 237–252. PMLR, 2018. 24

[142] G. Savathrakis and A. Argyros. An automated method for the creation of oriented bounding boxes in remote sensing ship detection datasets. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 830–839, 2024. 100

[143] M. Sebillo, G. Vitiello, L. Paolino, and A. Ginige. Training emergency responders through augmented reality mobile interfaces. *Multimedia Tools and Applications*, 75:9609–9622, 2016. 5

[144] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 12, 22, 27, 29, 84

[145] A. Shocher, N. Cohen, and M. Irani. "zero-shot" super-resolution using deep internal learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3118–3126, 2018. 89

[146] C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019. 46

[147] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 61

[148] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 117

[149] J. Spitz, J. Wagemans, D. Memmert, A. M. Williams, and W. F. Helsen. Video assistant referees (var): The impact of technology on decision making in association football referees. *Journal of Sports Sciences*, 39(2):147–153, 2021. 3

[150] G. Sreenu and S. Durai. Intelligent video surveillance: a review through deep learning techniques for crowd analysis. *Journal of Big Data*, 6(1):1–27, 2019. 24, 151

[151] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018. 117

[152] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 151

[153] W. Tang, B. Yang, X. Li, Y.-H. Liu, P.-A. Heng, and C.-W. Fu. Prototypical variational autoencoder for 3d few-shot object detection. *Advances in Neural Information Processing Systems*, 36, 2024. 116

[154] H. Tianyu, Z. Quanfu, and D. H. ShenYongjie. Overview of augmented reality technology. *Computer Knowledge and Technology*, 34:194–196, 2017. 73

[155] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE, 2015. 42

[156] G. Tsoumplekas, V. Li, V. Argyriou, A. Lytos, E. Fountoukidis, S. K. Goudos, I. D. Moscholios, and P. Sarigiannidis. Toward green and human-like artificial intelligence: A complete survey on contemporary few-shot learning approaches. *arXiv preprint arXiv:2402.03017*, 2024. 123

[157] Umbrellium. Umbrellium. Online, 2024. Accessed on March 25, 2024. 3

[158] V. N. Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999. 22

[159] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016. 117

[160] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. Ieee, 2001. 10, 26, 27, 29, 30

[161] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023. 38

[162] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh. Cspnet: A new backbone that can enhance learning capability of cnn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 390–391, 2020. 91

[163] C.-Y. Wang, H.-Y. M. Liao, and I.-H. Yeh. Designing network design strategies through gradient path analysis. *arXiv preprint arXiv:2211.04800*, 2022. 42

[164] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao. Yolov9: Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*, 2024. 42

[165] H. Wang, S. Tian, Y. Fu, J. Zhou, J. Liu, and D. Chen. Feature augmentation based on information fusion rectification for few-shot image classification. *Scientific Reports*, 13(1):3607, 2023. 60

[166] J. Wang, H. Zhu, S.-H. Wang, and Y.-D. Zhang. A review of deep learning on medical image analysis. *Mobile Networks and Applications*, 26:351–380, 2021. 25

[167] R. J. Wang, X. Li, and C. X. Ling. Pelee: A real-time object detection system on mobile devices. *Advances in neural information processing systems*, 31, 2018. 71

[168] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020. 44, 45, 116

[169] M. Whittaker, K. Crawford, R. Dobbe, G. Fried, E. Kaziunas, V. Mathur, S. M. West, R. Richardson, J. Schultz, O. Schwartz, et al. *AI now report 2018*. AI Now Institute at New York University New York, 2018. 151, 152

[170] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell. Understanding data augmentation for classification: when to warp? In *2016 international conference on digital image computing: techniques and applications (DICTA)*, pages 1–6. IEEE, 2016. 53, 59

[171] Z. Xin, S. Chen, T. Wu, Y. Shao, W. Ding, and X. You. Few-shot object detection: Research advances and challenges. *Information Fusion*, page 102307, 2024. 116

[172] J. Xiong, E.-L. Hsiang, Z. He, T. Zhan, and S.-T. Wu. Augmented reality and virtual reality displays: emerging technologies and future perspectives. *Light: Science & Applications*, 10(1):216, 2021. 74

[173] C. Xu, C. Liu, X. Sun, S. Yang, Y. Wang, C. Wang, and Y. Fu. Patchmix augmentation to identify causal features in few-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 11, 56, 57

[174] X.-Y. Xu, Q.-D. Jia, and S. M. U. Tayyab. Exploring the stimulating role of augmented reality features in e-commerce: A three-staged hybrid approach. *Journal of Retailing and Consumer Services*, 77:103682, 2024. 74

[175] K. Yan, X. Wang, L. Lu, and R. M. Summers. Deeplesion: Automated deep mining, categorization and detection of significant radiology image findings using large-scale clinical lesion annotations. *arXiv preprint arXiv:1710.01766*, 2017. 25

[176] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 670–685, 2018. 23

[177] S. Yoo and A. Blandford. Augmented reality and surgery: Human factors, challenges, and future steps. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 459–461. IEEE, 2022. 4

[178] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014. 92, 104

[179] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 54, 55, 57

[180] F. Zhang, Y. Shi, Z. Xiong, and X. X. Zhu. Few-shot object detection in remote sensing: Lifting the curse of incompletely annotated novel objects. *IEEE Transactions on Geoscience and Remote Sensing*, 2024. 116

[181] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 45, 52, 54, 55, 93, 120

[182] H. Zhang, Y. Wang, F. Dayoub, and N. Sunderhauf. Varifocalnet: An iou-aware dense object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8514–8523, 2021. 92

[183] Z. Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012. 7

[184] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019. 75, 134

[185] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020. 52

[186] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 37

[187] C. Zhu, M.-U. Io, H. F. B. Ngan, and R. L. Peralta. Interpreting the impact of augmented reality on heritage tourism: two empirical studies from world heritage sites. *Current Issues in Tourism*, pages 1–15, 2024. 74

[188] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu. Vision meets drones: A challenge. *arXiv preprint arXiv:1804.07437*, 2018. 94

[189] B. Zoph, E. D. Cubuk, G. Ghiasi, T.-Y. Lin, J. Shlens, and Q. V. Le. Learning data augmentation strategies for object detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 566–583. Springer, 2020. 45

[190] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 2023. 75, 134