Scene and crowd analysis using synthetic data generation with 3D quality improvements and deep network architectures

Anish R. Khadka

Kingston University

Scene and crowd analysis using synthetic data generation with 3D quality improvements and deep network architectures
Doctor of Philosophy

February 2021

Abstract

In this thesis, a scene analysis mainly focusing on vision-based techniques have been explored. The vision-based scene analysis techniques have a wide range of applications from surveillance, security to agriculture. A vision sensor can provide rich information about the environment such as colour, depth, shape, size and much more. This information can be further processed to have an in-depth knowledge of the scene such as type of environment, objects and distances. Hence, this thesis covers initially the background on human detection in particular pedestrian and crowd detection methods and introduces various vision-based techniques used in human detection. Followed by a detailed analysis of the use of synthetic data to improve the performance of state-of-the-art Deep Learning techniques and a multi-purpose synthetic data generation tool is proposed. The tool is a real-time graphics simulator which generates multiple types of synthetic data applicable for pedestrian detection, crowd density estimation, image segmentation, depth estimation, and 3D pose estimation. In the second part of the thesis, a novel technique has been proposed to improve the quality of the synthetic data. The inter-reflection also known as global illumination is a naturally occurring phenomena and is a major problem for 3D scene generation from an image. Thus, the proposed methods utilised a reverted ray-tracing technique to reduce the effect of inter-reflection problem and increased the quality of generated data. In addition, a method to improve the quality of the density map is discussed in the following chapter. The density map is the most commonly used technique to estimate crowds. However, the current procedure used to generate the map is not content-aware i.e., density map does not highlight the humans' heads according to their size in the image. Thus, a novel method to generate a content-aware density map was proposed and demonstrated that the use of such maps can elevate the performance of an existing Deep Learning architecture. In the final part, a Deep Learning architecture has been proposed to estimate the crowd in the wild. The architecture tackled the challenging aspect such as perspective distortion by implementing several techniques like pyramid style inputs, scale aggregation method and self-attention mechanism to estimate a crowd density map and achieved state-of-the-art results at the time.

Scene and crowd analysis using synthetic data generation with 3D quality improvements and deep network architectures

Anish R. Khadka

Kingston University

Scene and crowd analysis using synthetic data generation with 3D quality improvements and deep network architectures

Doctor of Philosophy

February 2021

Acknowledgements

There is saying that the essential aspect of the research career is to have the right people, such as mentors and little doing on own. I have been extraordinarily fortunate in the people I have had in this role. While working on this thesis was challenging, it was also exciting and rewarding at the same time. None of which would have been possible without the support and guidance of my supervising team.

First and foremost, I want to express my gratitude toward my supervisor Prof. Vasilis Argyriou for his continuous encouragement, patience and motivation for my PhD. His advice and guidance helped me throughout the research and writing of this thesis. I could not have imagined having a better advisor and mentor for my PhD study. Thank you for the excellent opportunities you have opened for me. I would also like to extend my gratitude to my second supervisor Prof. Paolo Remagnino for his encouragement and guidance during the research.

Besides my advisers, I am indebted to projects such as MIDAS, WITNESS, and MONICA to provide financial support for this research at Kingston University. I want to thank the Robot Vision Team (RoViT) at Kingston University who helped me throughout the research phase. I was fortunate to learn from and collaborate to produce the research in this thesis. In alphabetical order: Hamideh Keregari, Jiri Fajti, Mahdi Maktabbar Oghaz, Manzoor Razaak and Robert Dupre.

Most importantly, I would like to thank my family, mother and father, and my friends, who kept the conversation going and motivating. In particular, a big thank you to my wife, Shriya, for constant support throughout the journey.

Dedication

Declaration

I hereby declare that the content of this thesis is the product of my own research work. The main parts of this thesis have been published in the following list of my publications. I made the major contribution in design, implementation and experiments for these work under the guidance of Prof. Vasileios Argyriou and some others which have been acknowledged appropriately in text. The information, codes, and former ideas guided from other sources have been cited in a list of references is in the bibliography.

Journal Papers

On review

1. Khadka, Anish R., Remagnino, P., and Argyriou, V. (2020). "Adaptive Scale Variance Network, a method for accurate and effective crowd counting". In: *Computer Vision and Image Understanding.*

Conference Papers

1. Khadka, Anish R., Remagnino, P., and Argyriou, V. (2018). "Object 3D Reconstruction Based on Photometric Stereo and Inverted Rendering". In: *2018 14th International Conference on Signal-Image Technology Internet-Based Systems (SITIS).* 2018 14th International Conference on Signal-Image Technology Internet-Based Systems (SITIS), pp. 208–215.

2. Khadka, Anish R., Oghaz, M. M., Matta, W., Cosentino, M., Remagnino, P. and Argyriou, V. (2019). "Learning how to analyse crowd behaviour using synthetic data". In: *Proceedings of the 32nd International Conference on Computer Animation and Social Agents.* CASA '19. New York, NY, USA: Association for Computing Machinery, pp. 11–14.

3. Khadka, Anish R., Remagnino, P. and Argyriou, V. (2020)."Synthetic Crowd and Pedestrian Generator for Deep Learning Problems". In: *ICASSP 2020*

- *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). ISSN:2379-190X, pp. 4052–4056.

4. Khadka, Anish R., Vasileios, A., and Remagnino, P. (2020). "Accurate Deep Net Crowd Counting for Smart IoT Video acquisition devices". In: *2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. 2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS). ISSN:2325-2944, pp. 260–264.

5. Oghaz, Mahdi Maktabdar and Khadka, Anish R., Argyriou, V. and Remagnino, P., (2019). "Content-aware Density Map for Crowd Counting and Density Estimation". In: *Proceedings of the 32nd International Conference on Computer Animation and Social Agents*. CASA '19. New York, NY, USA: Association for Computing Machinery, pp. 11–14.

Abstract

In this thesis, a scene analysis mainly focusing on vision-based techniques have been explored. The vision-based scene analysis techniques have a wide range of applications from surveillance, security to agriculture. A vision sensor can provide rich information about the environment such as colour, depth, shape, size and much more. This information can be further processed to have an in-depth knowledge of the scene such as type of environment, objects and distances. Hence, this thesis covers initially the background on human detection in particular pedestrian and crowd detection methods and introduces various vision-based techniques used in human detection. Followed by a detailed analysis of the use of synthetic data to improve the performance of state-of-the-art Deep Learning techniques and a multi-purpose synthetic data generation tool is proposed. The tool is a real-time graphics simulator which generates multiple types of synthetic data applicable for pedestrian detection, crowd density estimation, image segmentation, depth estimation, and 3D pose estimation. In the second part of the thesis, a novel technique has been proposed to improve the quality of the synthetic data. The inter-reflection also known as global illumination is a naturally occurring phenomena and is a major problem for 3D scene generation from an image. Thus, the proposed methods utilised a reverted ray-tracing technique to reduce the effect of inter-reflection problem and increased the quality of generated data. In addition, a method to improve the quality of the density map is discussed in the following chapter. The density map is the most commonly used technique to estimate crowds. However, the current procedure used to generate the map is not content-aware i.e., density map does not highlight the humans' heads according to their size in the image. Thus, a novel method to generate a content-aware density map was proposed and demonstrated that the use of such maps can elevate the performance of an existing Deep Learning architecture. In the final part, a Deep Learning architecture has been proposed to estimate the crowd in the wild. The architecture tackled the challenging aspect such as perspective distortion by implementing several techniques like pyramid style inputs, scale aggregation method and self-attention mechanism to estimate a crowd density map and achieved state-of-the-art results at the time.

Contents

List of Figures

List of Tables

List of Abbreviations

2D . . . . . Two Dimension

3D . . . . . Third Dimension

AI . . . . . Artificial Intelligence

ANN . . . . . Artificial Neural Network

BCE . . . . . Binary Cross Entropy

CV . . . . . Computer Vision

CNN . . . . . Convolutional Neural Network

DL . . . . . Deep Learning

DNN . . . . . Deep Neural Networks

HOG . . . . . Histograms of Oriented Gradients

HSV . . . . . Hue, Saturation and Value

ML . . . . . Machine Learning

MSE . . . . . Mean Square Error

MAE . . . . . Mean Absolute Error

PSNR . . . . . Peak Signal-to-Noise Ratio

RGB . . . . . Red, Green and Blue

SVR . . . . . Support Vector Regression

SVM . . . . . Support Vector Machines

SIFT . . . . . Scale Invariant Feature Transform

SSIM . . . . . Structural Similarity Index Measure

TML . . . . . Traditional Machine Learning

UAV . . . . . Unmanned Aerial Vehicles

Go confidently in the direction of your dreams. Live the life you have imagined.

— Henry David Thoreau

1 Introduction to Scene Analysis

Contents

As humans, we experience the world with the number of sensory organs; however, the essential sensor indubitable is a vision sensor. The vision provides rich information about the scene such as colour, depth, shape, size and much more. In this thesis, we define scene analysis in terms of vision explicitly. In general, scene analysis means to examine the content of a given scene or scenario and describe it in meaningful ways. Scene analysis can be as simple task as describing the content and as complex as deducing detailed information about it. Here, the term *"simple"* is described as relative to the human experience. For example, the human can easily describe the core elements of a figure 1.1 (i.e., people raising hands)

Figure 1.1: A photograph of people raising hands by Quintero (2019)

without any difficulty. However, the same is not right about Artificial Intelligence (AI). Nevertheless, due to considerable progress made in the past decade, it is now possible for machines to describe a simple picture (Vinyals et al. 2015).

1.1 Impact and Application of Scene analysis

The scene analysis has a broad range of applications and is already applied in several scenarios. Following are some examples that demonstrate the benefit of scene analysis:

- **Surveillance and Security:** Scene analysis is a crucial part of surveillance and security. It provides useful information within a short period of time such as crowd, pedestrian (S. Zhang et al. 2018), face (M. Zhu et al. 2019; Leo et al. 2020) and eye detection (Ancheta et al. 2018), human actions recognition (Bloom, Argyriou, et al. 2017; Bilinski et al. 2016; Bloom, Makris, et al. 2014), first person scene understanding (Rodin et al. 2020), fall detection (Z. Huang et al. 2018; Fang et al. 2018) and abnormal behaviour (Xie et al. 2019).

- **Autonomous vehicles:** For the purpose of autonomous vehicles, diverse scene analysis techniques are employed. For example, Automatic parking system (Heimberger et al. 2017), object detection (Takumi et al. 2017), road lane detection (Hoang et al. 2016), traffic sign detection (Z. Zhu et al. 2016), and much more.

- **Road and Traffic Safety:** Another useful application of scene analysis can be seen in road and traffic sign management. The techniques can be applied to improve traffic flow (Lira et al. 2016), detection of roadside occupant (Ho et al. 2019), automatic licence plate recognition (Laroca et al. 2018), and road weather condition estimation (Ozcan et al. 2020). Other applications can be detecting road cracks, anomaly detection (Santhosh et al. 2020) and traffic counting (J.-P. Lin et al. 2018).

- **Entertainment:** Sports in the entertainment field is one of the most analysed topics. Hence, automatic scene analysis in sports has seen increased research in recent days. Some core technology such as multi camera tracking (R. Zhang et al. 2020), player detection and tracking (Y. Yang et al. 2017), ball tracking (Kamble et al. 2019), sport specific action recognition (Cust et al. 2019) applies scene analysis techniques to extract information. In addition, the scene analysis technique is also extensively utilised in the computer game such as action recognition using Kinect (Gang Li et al. 2020), cheat detection in games (Witschel et al. 2020) and much more (M. Li, G. Xu, et al. 2018).

- **Agriculture:** Another field that can benefit from scene analysis is an agriculture field. In several situation the techniques can be employed for better agriculture such as smart agricultural farming (Chang et al. 2018; Patrício et al. 2018), food quality grading system (Arakeri et al. 2016), precision agricultural using UAV (Unmanned Aerial Vehicles) (Alsalam et al. 2017), plant disease detection using leaves (Kuricheti et al. 2019), smart farming (Guo et al. 2020) and sustainable agriculture (Tombe 2020) using satellite images.

- **Natural emergencies:** Scene analysis methods can be applied in many ways as an early warning system in the circumstances such as flood (Bhola et al. 2019) and tsunami detection. In another case, for rescue and search operation (Gotovac et al. 2016), crowd disaster avoidance system (Yogameena et al.

2017), and after disaster analysis and monitoring (Kamilaris et al. 2018) the scene analysis can be a vital tool.

- **Other:** There are countless other examples where scene analysis can be useful. Such as, in the medical field (Kuvaev et al. 2020), Smart library book sorting (X. Shi et al. 2020), 3D Conceptual design (Z. Yang et al. 2020), and much more.

This thesis will carry out a scene analysis using Machine Learning (ML) techniques known as Deep Learning (DL). Nevertheless, before we jump into details on how scene analysis can be achieved, we must first distinguish what we mean by Artificial Intelligence, Machine Learning and Deep Learning.

1.2 Introduction to Artificial Intelligence

Artificial Intelligence (AI) has been around for decades. The general concept of AI is to achieve a truly intelligent machine. Here, the term *truly intelligent* can be understood as a system with characteristics of intelligence, like, we observe in human behaviour (Russell 1997). The field can be defined as *"the effort to automate the intellectual task normally performed by humans"* (Chollet 2018).

In the early phase of AI, numerous experts believed that human-level intelligence can be handcrafted with a large set of rules and was known as *symbolic AI* (Chollet 2018). Although *symbolic AI* worked well for logical problems such as chess, its limitation on defining rules for complex problems in image, speech and language led to the invention of the Machine Learning field. The figure 1.2 shows the hierarchical relationship between AI, ML and DL fields. We can observe that AI is a general field that incorporates Machine learning, Deep Learning, and many other techniques that do not involve learning, such as *symbolic AI*.

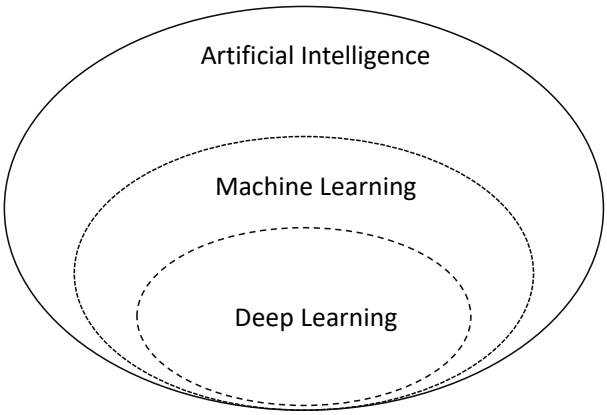


Figure 1.2: Artificial intelligence and its relationship with Machine learning and Deep Learning.

1.2.1 Machine Learning (ML)

Machine Learning is a subfield within AI. However, it also interacts with other fields such as statistics and computer science and is also known as *predictive analytics* or *statistical learning*. The core concept of ML is about extracting knowledge from data (Müller et al. 2018). Unlike *symbolic AI* which requires explicit programming, the ML system is *trained* to produce appropriate responses. When presented with a considerable number of relevant examples to the task, it discovers statistical representation[1] of them and produces a set of rules to automate such tasks (Chollet 2018).

In recent years, Machine Learning has gained popularity within the research community. Subsequently, wide ranges of fields have benefited from such research. In fact, we are already using Machine Learning daily from an automatic recommendation of which product to buy, to which film to watch and recognition of friends and family in the picture you have just uploaded on the websites. Numerous widely popular websites use Machine Learning as their core user experience[2]. The applications such as image detection and classification (Lan et al. 2018), video

[1] *Representation* is defined as a way of observing data differently. A simplistic example would be to look at image data; It can be viewed as Red, Green and Blue (RGB) or in Hue, Saturation and Value (HSV) format, either way, it is the same data but different Representation.

[2] Websites such as Facebook utilises Machine Learning techniques to achieve highly accurate face detection (Khan et al. 2018)

analysis (Nishani et al. 2017) and recommendation (Gao et al. 2017), security intrusion detection (Yin et al. 2017) and so forth are a fraction of examples where ML has been utilised (Pouyanfar et al. 2018).

1.2.1.1 Traditional Machine Learning (TML)

The traditional Machine Learning techniques are based on conventional methods and cannot process the data in raw form. TML techniques require extensive domain expertise as well as a meticulously engineered system (Pouyanfar et al. 2018). Usually, the architecture involves a feature extractor which transforms the raw data such as image pixel values into appropriate format or feature vector from which a learning subsystem could detect or classify patterns in the input.

The downside of using TML is that the technique relies highly on the input data's Representation. Hence, feature engineering has been the leading research direction in ML for an extended period, where much focus was given in building features and extraction methods from the raw data. A further disadvantage of using TML is that feature extractor does not generalise well in cross-domain problems (i.e., the extractor is limited to their designed domain). There have been numerous attempts in proposing a good feature extractor such as Histogram of Oriented Gradients (HOG) (Dalal et al. 2005), Haar-like features (P. Viola et al. 2001), Scale Invariant Feature Transform (SIFT) (Lowe 1999). More details about feature extractor are discussed under section 2.2

1.2.1.2 Deep Learning (DL)

Among numerous ML algorithms, *Deep Learning (DL)* also known as representation learning (Deng 2014) is used in modern day applications[3]. The main reason behind the Deep Learning trend is due to its performance. In some cases, Hekler et al. (2019) stated that the Deep Learning model surpasses even the human expert performance.

In comparison with TML techniques, Deep Learning methods accomplish the feature extraction in an automated manner and require minimal domain expertise

[3]Lan et al. 2018; Nishani et al. 2017; Gao et al. 2017; Yin et al. 2017.

and human intervention. Deep Learning has three significant advantages over TML techniques, and they are;

- Deep Learning architecture offers a simple solution to any given problem without requiring problem-specific tweaks and tricks.
- Deep Learning methods are easily scalable and can handle large datasets without running into computational problems.
- Deep Learning models once trained in specific data can also be used in other related datasets, and learned features are general enough to handle such situations. For example, in recent years it is widely common to use pre-trained Deep Learning models (e.g. trained on ImageNet dataset (Russakovsky et al. 2015)) in various other context (Weng et al. 2020).

Deep Learning is the outcome of a certain number of core techniques, and novel approaches are added constantly. Some primary enablers of Deep Learning approach are Convolutional Neural Networks (CNN), pooling and activation functions[4].

1.3 Motivation of Thesis

In this thesis, we further narrow down the scene analysis topic to pedestrian and crowd analysis. We focus on human detection in small and large groups. Besides, we also concentrate on the data generation process to elevate the lack of data in large groups detection.

1.3.1 Importance of Human Detection

The world has seen a dramatic increase in the human population in the past century. The growth has made crowd phenomena increasingly common. Large gatherings can be seen in indoor areas such as airports, building halls, shopping centres, and outdoor areas such as parks, riversides, sports events, and public demonstrations. The purpose of such a gathering can provide vital information

[4] see section 2.4 for detailed discussion

to analyse the behaviour and properties of the crowd. As a result, research in the subject has a great interest in the scientific field, such as computer vision, public service, statistical physics, psychology, and behaviour (Grant et al. 2017). Such studies' significance becomes even more apparent when we investigate crowd turbulence, which is the primary reason for crowd disasters resulting in mass-panic, stampede, and overall loss of control (Saleh et al. 2015).

There are plenty of recorded events related to crowd tragedy (J. Wang et al. 2013; Illiyas et al. 2013; Krausz et al. 2012) as such it is crucial to have a rigorous understanding of the crowd to prevent such accidents. Crowd analysis can be applied in various interdisciplinary fields, and a handful of useful applications are discussed below.

- **Disaster management:** Plenty of scenarios include gathering crowds such as musical events, political rallies, public demonstration, and sports events. These events have a high chance of crowd related tragedy, as mentioned above. Hence, crowd analysis can play a critical role in the effective management of crowds and avoid overcrowding and reduce such risk (Abdelghany et al. 2014).

- **Public space design:** Study of crowds in public space can provide an insight into the design flaw of the public space such as train stations, airports, and other public places (Chow et al. 2008). Considering recent events such as COVID-19, it is apparent that space design is even more crucial than before. The social distancing phenomena highlights that public and private space are not designed well to accommodate it adequately. Hence, scene analysis can provide useful information to improve the design of the architecture and the public space's usability.

- **Information gathering and analysis:** Diverse type of intelligence can be generated from crowd analysis approaches such as total number visitors in a shop at a given time, which can be used to allocate staff numbers efficiently. A similar method can be used to identify the pedestrian flow and manage the signal-wait time (Vishwanath A. Sindagi et al. 2018).

(a) Perspective distortion **(b)** Non-uniform distribution

(c) Occlusion **(d)** Complex Background

Figure 1.3: Some sample images found in the ShanghaiTech dataset (Yingying Zhang et al. 2016) show the challenges that lie in the crowd analysis field. Perspective distortion, non-uniform distribution, Occlusion, complex background are some examples of difficulties in crowd analysis.

- **Virtual crowd modelling:** The crowd analysis provides us with a much more in-depth understanding of the crowd behaviour, in turn, such information can be used to establish an improved mathematical model, thereby improving the virtual crowd modelling. It helps generate realistic crowds in the application such as computer games, films and designing the emergency evacuation schemes (Perez et al. 2016).

- **Cross domain applications:** Due to the versatile nature of the learning in crowd analysis, numerous other domains have extended the crowd counting techniques to nonhuman related fields such as counting microscopic cells (Lempitsky et al. 2010), vehicle counting (Y. Li et al. 2018), environmental survey (French et al. 2015; Zhan et al. 2008) and much more.

1.3.2 Challenges of Crowd Estimation and Counting

Crowd estimation and counting are challenging topics due to severe Occlusion, perspective distortion, varied, complex background and diverse crowd scenario (C. Zhang, K. Kang, et al. 2016). Figure 1.3, shows the problem that exists in the crowd analysis field. Most of the early research on crowd analyses were focused on scene-specific and were not generalised enough to be used in other scenarios. Some research (Antoni B Chan, Liang, et al. 2008; Antoni B. Chan et al. 2008) used for crowd counting required a manual annotation of a handful of frames from the target scenes for training purposes. Though some problems have improved below, we discuss difficulties that still exist in crowd analysis.

- **Complexity present in diverse crowd scenes:** As shown in figure 1.3, it is particularly challenging to design and develop an algorithm which can handle all the complex scenarios where crowds can exist. The figure 1.3 shows that the crowd analysis approaches needs to accommodate complicated background and also need to solve the problem of *occlusion*[5] which hides the useful features to locate heads in the scene; *perspective distortion*[6] where a couple of pixels can represent groups of heads at one place of image and a lot more to represent a single head. In the past, crowd analysis used head or body detection techniques to analyse the crowd, and the diversion from such an approach occurred in the last decade. In recent years, the crowd analysis used direct regression of crowd count rather than counting heads or people's bodies.

- **Inadequate crowd dataset:** One of the main driving forces in the crowd analysis field is the availability of the public crowd dataset (C. Zhang, K. Kang, et al. 2016). While there are range of publicly available dataset such as *Shanghaitech* dataset (Yingying Zhang et al. 2016), *UCF-CC-50* dataset (Idrees, Saleemi, et al. 2013), *Mall* dataset (K. Chen et al. 2012), *Venice*

[5]Occlusion is the constant source of headache in computer vision field (M. Zhu et al. 2019).
[6]Perspective distortion changes the size of head appearance based on the distance with the camera. Head appears larger nearer to the camera and smaller away from it.

dataset (Weizhe Liu et al. 2019), *UCSD* dataset (Antoni B Chan, Liang, et al. 2008) at present. However, due to the nature of Deep Learning[7] which requires enormous amounts of data, these datasets do not contain a high number of samples for the algorithm to generalise well over multi-scene. For example, while the *UCF-CC-50* (Idrees, Saleemi, et al. 2013) dataset presents a wide range of head counting annotation in their samples, it only contains 50 images (hence, 50 in their dataset name too). In section 2.6.1 we discuss in detail about the available datasets.

- **Density map generation with flaw technique:** As crowd analysis is carried out using density map. The density map must present the correct information. However, the common crowd counting dataset provides annotation in the form of head centred pixel location instead of masking the entire head region. Further, apply a two-dimensional (2D) Gaussian filter or a dynamic 2D Gaussian based on K nearest neighbour to generate the density map. Due to this nature, the generated map is not content-aware and incorporates a significant amount of false information into the ground truth map (Idrees, Tayyab, et al. 2018; Yingying Zhang et al. 2016).

- **Accurate training data for crowd estimation and counting:** The real reason behind an inadequate number of crowd data sets available is the difficulties involved in generating such data. Annotating thousands of heads in a single image is an extremely labour-intensive task and on top of that, annotating correctly is another more significant challenge. Due to the perspective distortion in images, some heads can be as small as a single pixel value, annotating such heads is almost impossible for humans. Hence, it is common to find the inaccuracy in most publicly available datasets such as in Shanghaitech dataset (Yingying Zhang et al. 2016) and UCF-CC-50 (Idrees, Saleemi, et al. 2013) dataset.

[7] In this thesis, we focus on Deep Learning techniques for crowd analysis purposes

1.4 Our contributions

The contribution of this thesis can be summarised in the following:

1. To solve the problem with a small number of crowd datasets, we put forward a multi-purpose synthetic data generation tool that utilises a real-time graphics engine to generate copious quantities of data necessary for Deep Learning problems. The tool is adaptable to the user's needs and can generate data real-time for crowd analysis, pedestrian detection, 3D pose estimation, image segmentation and depth map. Chapter 3 goes in detail about the tools and shows that it is possible to improve the state-of-the-art results by merely retraining the network with a synthetic dataset and fine-tuning with a real dataset.

2. We proposed to use a reverted ray tracing mechanism to reduce the effect of inter-reflection and enhance the appearance of generated data in chapter 4. Inter-reflection is a major problem in generating 3D synthetic data from images. We proposed to employ a traditional method called Photometric stereo and reverse engineer the ray-tracing approach to extract the environment noise caused by inter-reflection and reduce it from the final synthetic data generation process. We showed that our approach improved the existing 3D generation process and might be useful for crowd domain problems which suffer from inter-reflection.

3. In chapter 5, we proposed a method that addresses the limitation in density map generation through a content-aware annotation technique that applies a combination of the nearest neighbour algorithm and unsupervised segmentation to generate the density map head masks. Furthermore, we demonstrate that with simply changing the way we generate the data, the existing state-of-the-art network can achieve higher accuracy.

4. Finally in chapter 6, we explore the problem of perspective distortion in the crowd counting and estimation field and propose a Deep Learning architecture

with an effective way to capture multi-scale feature using pyramid contextual module in combination with scale aggregation and self-attention mechanism. We also proposed a novel loss function *Switch loss function* to maximise the quality of the predicted density map and accuracy. The loss function utilises multiple methods such as PSNR, SSIM, and Root, which means square error to achieve a higher quality density map and accuracy. We also illustrate that by using a variation of the above method, we can achieve state-of-the-art results.

1.5 Constraints applied to the proposed method

The research in this thesis primarily focuses on scene analysis and in particular crowd analysis. Following are the few assumptions and limitations that were made in this thesis.

For the proposed synthetic data generation method in chapter 3, the primary limitation was the diversity of synthetic avatars. We mainly focused on the demonstrable application instead of the diversity in the avatar used for data generation. Hence, the generated data is limited in terms of variety and diversity such as equal representation of gender, inclusion of less able people such as people in wheelchairs, different ethnicity and ages. We assumed that despite the lack of balance representation of various groups, the synthetic dataset is still useful in combating the problem of lack of datasets in the crowd counting field. Our core focused on the expected outcomes of the application instead of the usability of the synthetic data in the Deep Learning field and the problem of tackling the lack of data in the crowd counting field. Also to demonstrate that synthetic datasets can still improve the trained models and generalise well to real scenes.

In terms of the proposed inter-reflection removable method described in chapter 4, while we considered the wider range of possibilities of using the method, we limited our experiments in a specific and controlled environment. Hence, while it might be theoretically possible to apply the method in other scenaiours, more

experiments are required to validate it. Our primary goal was to demonstrate the use of the proposed method as well as the benefit of using our method.

For the case of density map generation in chapter 5, we focused on the brute-force method which can be limiting in many scenarios. We primarily concentrated our efforts on solving the existing problem. However, the approach could have been researched more to have a general purpose solution.

Finally, although we achieved state-of-the-art results on crowd counting with the new proposed deep learning architecture discussed in chapter 6, the architecture has quite limited lifespans. Every month new approaches are proposed which produce state-of-the-art results. Although the proposed method is limited, the various techniques that were applied in the architecture itself can be useful in various other applications.

1.6 Thesis outline

The remainder of the thesis is in the following order: Chapter 2, presents the background on the topics related to pedestrian and other information associated with Deep Learning and crowd analysis. The chapter 3 explored the necessity of large amounts of data for the scene analysis field in particular on human activity and a multi-purposed synthetic data generation tool is proposed as a possible solution for the problem. Chapter 4 examined the inter-reflection problem in the 3D generation from images and proposed iterative 3D reconstruction using ray-tracing to improve the quality of synthetic data. In chapter 5, the current problem with density map generation is investigated and proposed a novel method for generating content-aware density maps. Chapter 6, a Deep Learning architecture called ASVnet is proposed to solve scale variation in a crowd due to perspective distortion. Finally, chapter 7 summarised the thesis and discussed future research.

"There is only one corner of the universe; you can be certain of improving, and that is yourself."

— Aldous Huxley

2

Before we begin: Core concept of Pedestrian and Crowd detection

Contents

In chapter 2, we dive into the essential aspect of Machine Learning techniques initially focusing on essential methods involved in pedestrian and crowd detection approaches, followed by the exploratory background knowledge which looks into the evolution of crowd detection to estimation approach. Deep Learning components are discussed after that. Finally, we present publicly available datasets that are commonly used for training purposes in crowd analysis.

2.1 Pedestrian detection and Crowd Analysis

Pedestrian detection and crowd analysis come under the human activity recognition field (Grant et al. 2017). Early human recognition research focused on single human activities such as hand gesture recognition, action recognition such as walking, kicking, and a combination of such recognition to perform more complex actions detection such as baseball throwing.

Generally, crowd analysis is carried out in two major approaches; direct (also known as *detection-based*) and indirect (also known as *map-based* or *measurement-based*) crowd estimation (Conte et al. 2010). The direct approach applies the techniques used in pedestrian detection methods where individuals are first detected and counted. Whereas for the indirect approach, the counting is performed by measuring some features which do not require separate detection of individuals in the scene. The key point to note here is that crowd analysis focusing on estimation and counting has evolved from using detection-based approach to a map-based approach, also known as density map. The benefit of using a density map is that it preserves much more information than just a headcount estimation. In contrast to counting, density map provides the crowd's spatial distribution and more significant insights of crowd behaviour (Yingying Zhang et al. 2016). In figure 2.1, we can see that instead of just head counting, the density map can visualise the dense and less dense area in the image. These kinds of information are also applicable in identifying abnormal behaviour in the crowd (Yingying Zhang et al. 2016).

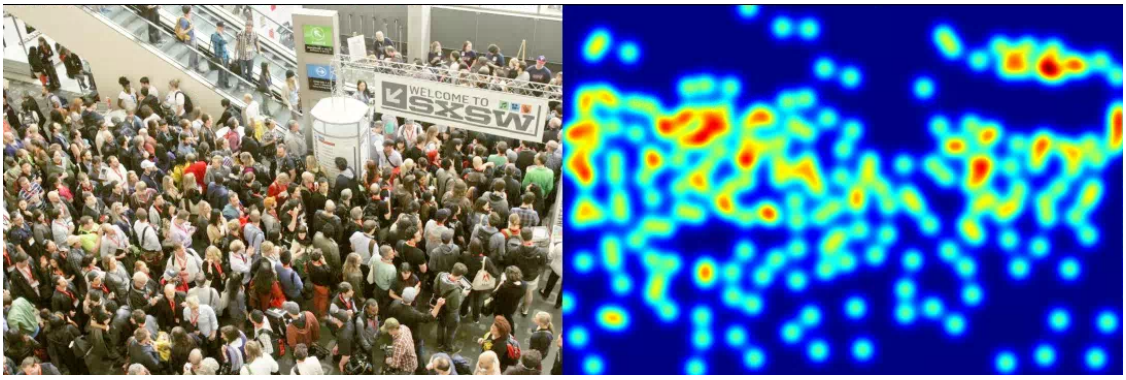


Figure 2.1: Sample image from Shanghaitech (Yingying Zhang et al. 2016) dataset (left) with density map of the crowd on the right image.

2.2 Features for pedestrian and crowd detection

Feature extraction is transforming arbitrary data such as image into derived values (i.e., features) (Errami et al. 2016) which can be useful for improving learning performance and better generalisation. In the following, we examine a couple of feature extraction techniques in pedestrian and crowd detection methods.

2.2.1 Histograms of Oriented Gradients (HOG)

Histograms of oriented gradients (HOG) are among the most widely used feature extraction methods in object detection. Freeman et al. (1995) introduced orientation histogram as a feature vector for hand gesture classification and interpolation. Followed by Dalal et al. (2005) who carried out a detailed analysis of HOG in human detection. The work done by Dalal et al. (2005) is widely cited in HOG pedestrian detection methods. The HOG is also used together with different methods such as Ada-Boost (Jin et al. 2012), Support Vector Regression (SVR)(Errami et al. 2016). Jin et al. (2012) applied HOG and tracking-by-detection (Ada-Boost classifier) integrated with crowd simulation to improve crowd tracking. In Errami et al. (2016), applied HOG jointly with SVR was used for pedestrian detection. Dee et al. (2010) used HOG in combination with KLT feature tracker to analyse crowd behaviours. Ge, Robert T. Collins, et al. (2012) presented a study on pedestrian crowds which automatically detect small to medium groups of individuals using a full-body HOG detector combined with correlation tracker for localising pedestrians'

in-crowd. M. Li, Z. Zhang, et al. (2008) also utilised HOG based head-shoulder detection alongside Mosaic Image Difference (MID) based foreground segmentation to estimate people in-crowd.

The typical procedure for generating HOG descriptors can be divided into three stages. Firstly, we calculate the horizontal and vertical gradients of a given image. It is accomplished by using the following kernels as masks at each pixel of the image.

$$K_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}, K_y = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \tag{2.1}$$

As shown in the figure 2.2, after applying K_x and K_y to the input image (a), we get G_x (b) and G_y (c) respectively. In second stage, we estimate the magnitude m and orientation θ with following equation:

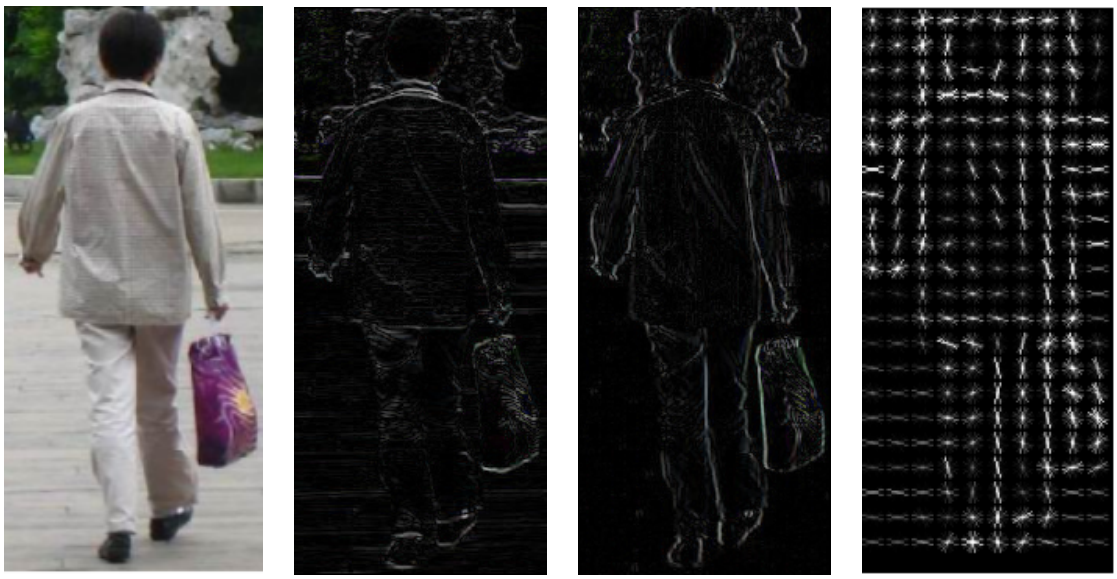
$$m = \sqrt{G_x^2 + G_y^2} \tag{2.2}$$

$$\theta = arctan\left(\frac{G_y}{G_x}\right), \text{ where } \theta \in [0, \pi] \tag{2.3}$$

Lastly, the input image is divided into small cells (usually 8×8 pixel). Then HOG is calculated for each of them based on orientation in the range $[0, \pi)$ and equally divided into nine bins corresponding to angle $(0, 20, 40 \ldots 160)$.

2.2.2 Haar-like Features

P. Viola et al. (2001) introduced Haar-like features and implemented the first real-time face detection. The features were driven by the study (Oren et al. 1997), which showed that while the absolute intensity values of different regions in images changed drastically under different lighting conditions, the overall relationship among regions remained unaffected. P. Viola et al. (2001) also showed that features are scalable and compute efficiently in regular periods. The key edge of Haar-like features over other methods is its computational speed. These features are rectangle filters in shape as shown in figure 2.3. Yongzhi Wang et al. (2010) applied Haar-like features with the Ada-Boost classifier for pedestrian detection. Similar techniques have been applied



(a) Input image **(b)** Horizontal gradient computation **(c)** Vertical gradient computation **(d)** Gradients Orientations

Figure 2.2: Generation of HOG feature vector. Sample Image used from Penn-Fudan dataset (L. Wang et al. 2007).

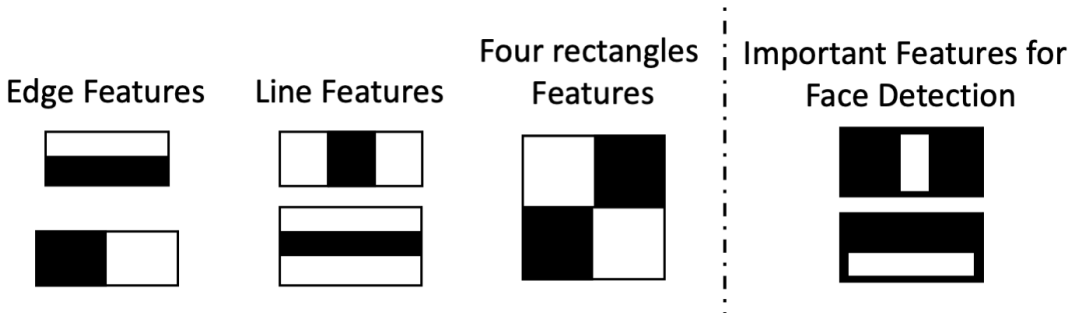


Figure 2.3: Left) Different types of Haar-like features. Right) two vital features used in face detection (Dey 2018).

in (H. Sun et al. 2011; Cerri et al. 2010), for pedestrian detection focusing on different types of cameras (i.e., RGB, infrared). Sim et al. (2008) exploited Haar-like features as well as boosted classifiers to detect individuals within the crowded environment. Nevertheless, like pedestrian detection features, extractor Haar-like features are not as good as HOG. The main reason behind it is the absence of gradients based on Haar-like features that could barely extract pedestrians' contour characteristics.

2.2.3 Deep Neural Network Features

As discussed earlier in chapter 1, Deep Learning extracts the features automatically. Deep Learning is a set of approaches that can take in raw data and automatically learn the representations (LeCun et al. 2015). DL falls in the representations ML methods which apply multiple levels of representations to extract the information. Several non-linear but straightforward modules transform the representation from the previous level into a higher abstract level (Hosseini et al. 2020). The transforms start with the raw input and with the composition of enough such transformations, complex features and inferences can be learned. The term *Deep* in Deep Learning does not represent the meaning of deeper understanding acquired by the method; instead, it describes the number of layers used in the architecture.

The figure 2.4 illustrates the architecture of a Deep Learning network used for handwritten number recognition and shows the multiple successive layers and features extracted in each layer. At present, it is common to see the Deep Learning structure to have tens to hundreds of layers. Simultaneously, there are other approaches where learning is focused on using the small number of layers and sometimes known as *shallow learning* (Chollet 2018).

2.3 Classifiers

The classification involves taking a task and mapping function $f(x)$ to discrete value y, where x is an input variable, and y is an output variable. For example, a dog and cat classifier, where $y \in \{dog = 1, cat = 2\}$, when presented with dog image x_{dog}, the function $f(x_{dog})$ should give us $y = 1$. A classifier learns the mapping by the training data to assign the input to certain output values. In the following, we will discuss some of the widely used classifiers in the crowd analysis field.

2.3.1 Support Vector Machine (SVM)

Support Vector Machines (SVM)[1] in its modern form was proposed by Cortes et al. (1995) and is one of the most widely used and effective statistical supervised

[1] Also known as Support Vector Network (SVN)

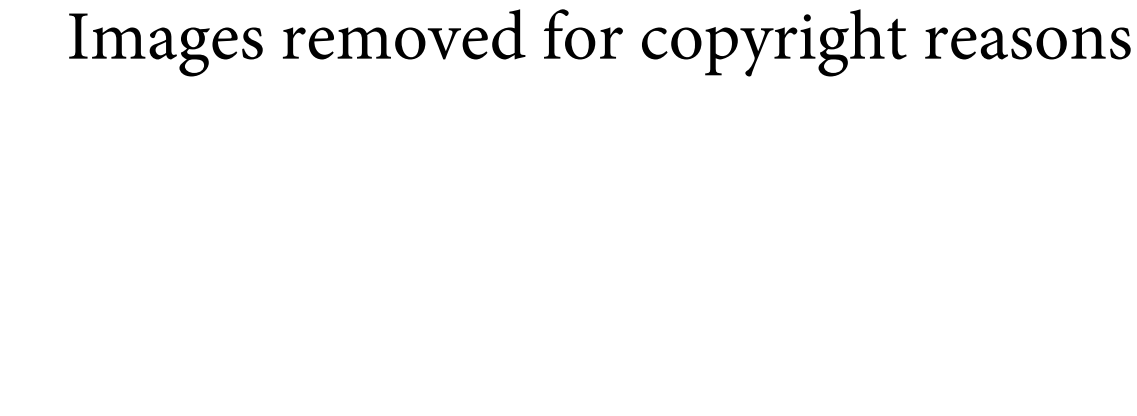


Figure 2.4: Visualisation of the layers of Deep Learning model which shows the transformation of input image (number five) into higher more abstract form to produce the result (A Classification model which is able to recognise the handwritten characters)

machine learning method. The SVM leverages the support vectors and hyperplane to transform the input vectors into higher dimensional features. Simply put, the core principle of SVM is to identify the pair of parallel hyperplanes that results in the maximum boundaries between two classes (M. Zhu et al. 2019).

There are a number of examples that applied SVM and other techniques for crowd analysis purposes (Manfredi et al. 2014; Solera et al. 2013; Xiaohua et al. 2006). Manfredi et al. (2014) took advantage of SVM to classify the static crowd detection and localisation. In (Solera et al. 2013), SVM based learning mechanism was applied in combination with annotation dataset to detect groups and calculate distance.

$$\begin{aligned} &\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{k} \zeta^i \\ &\text{subject to } y_i(w^T x^i + b) \geq 1 - \zeta^i \text{ and } \zeta^i \geq 0 \end{aligned} \tag{2.4}$$

As mentioned above, the primary goal of SVM is to identify the support vectors with maximum distance. When a vector x which belongs to two classes A and B and vector $y \in \{1, -1\}$, SVM for soft margin classifier is calculated with equation () (2.5). The main objective is to find weight vector w and bias term b by maximizing

the margin between the classes and penalising when a sample is within the margin boundary or misclassified. In ideal scenarios, $y_i(w^T x^i + b)$ would be 0 for all the samples (i.e. 100% prediction). However, problems which are not perfectly separable with hyperplane, $\zeta^i \geq 0$ distance is introduced to some samples to allow it to be at a certain distance from the margin boundary. Furthermore C is a penalty term which controls the strength of the penalty.

In addition, the biases b are generally calculated based on the support vectors that lie on the margins (i.e. $0 < x_i < C$). The main reason behind is due to $y_i(w^T x_i + b) = 1$. When $y_i^2 = 1$, bias b can be calculated as follow:

$$b = y_i - w^T x_i \tag{2.5}$$

2.3.2 Adaptive Boosting (AdaBoost)

Adaptive boosting (AdaBoost) is an ensemble boosting classifier which is jointly used with weak classifier[2] to improve the overall performance.

Many researchers have employed AdaBoost techniques for crowd estimation and counting. D. Kim et al. (2012) used multi-class AdaBoost with spectral texture features to estimate the crowd density. In (Qiming et al. 2017), they used Haar-like features and AdaBoost algorithms to detect faces and identify crowd attention. Other research (Jin et al. 2012; Dee et al. 2010), applied HOG and AdaBoost together for crowd analysis.

In AdaBoost, a set of weak binary classifiers is trained at the beginning. Further training is then carried out with the weighted version of the training points where weights are increased for misclassified points and decreased for the correctly classified points. Once training is completed, the sum of N individuals classifiers results into final classifier output.

$$F_M = \sum_{m=1}^{N} f_m(x) \tag{2.6}$$

[2]Weak classifier is defined as a classifier which shows poor performance when used alone (Sugiyama 2016)

Figure 2.5: Visualisation of n-dimension perceptron linear classifier, where x_i represents input vector and w_i as weights. The figure shows a single block of artificial neural network.

In the equation 2.6, F_M is the final classifier output, where $f_m(x)$ is a weak classifier in the range of $\{m = 1, ..., N\}$. x is the training samples.

2.3.3 Neural Network Perceptron Linear Classifier

Rosenblatt (1957) first introduced perceptron as a generalised computational framework for solving linear problems (Joshi 2020). Perceptron is a fundamental framework of Artificial Neural Network (ANN) and figure 2.5, shows a single building block of ANN. In perceptrons, weights and bias are learned from the equation 2.7 where weights expose the strength of the particular neuron and bias makes it possible for activation function curves to go up-down.

In n-dimensional space, a single layered perceptron with linear mapping represents a linear plane. For input vector $\{x_1, x_2, ..., x_n\}$ and the weights $\{w_1, w_2, ..., w_n\}$ in n-dimensions we can represent perceptron as following.

$$f(X, W) = \phi(\sum_{i=1}^{n}(w_i \cdot x_i)) \tag{2.7}$$

In the equation 2.7, X and W represents input and weights of the neuron. i is a number of weights and inputs. Here, multiplication of x and w is summed and fed into the activation function ϕ.

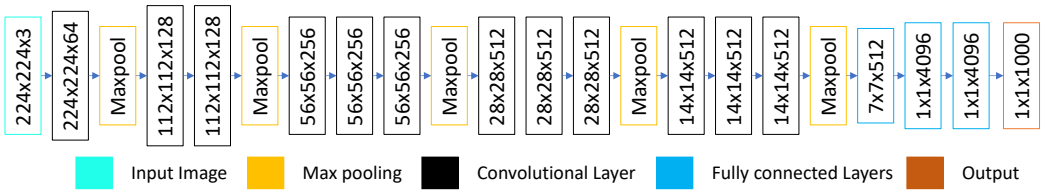


Figure 2.6: Overview of VGG-16 (Simonyan et al. 2015) architecture. All the layers of VGG-16 from input image to output layers.

2.4 Deep Learning components

Deep Learning is composed of numerous technologies, and some core technologies are discussed in the following.

2.4.1 Convolutional Neural Network (CNN)

The concept of Convolutional Neural Network (CNN) was first introduced by Fukushima (1980) and was later Lecun et al. (1998) improved it immensely and proposed the LeNet-5 neural network architecture to recognise handwritten characters. Initially proposed for computer vision problems, the Convolutional Neural Network has gained popularity virtually in every sub-field of Deep Learning.

In figure 2.6, we can see one of the simple but widely popular deep convolutional neural networks called VGG-16. VGG-16 was proposed by Simonyan et al. (2015). The neural network achieved top-5 test accuracy in ImageNet (Russakovsky et al. 2015) in ILSVRC-2014 (ImageNet Large Scale Visual Recognition Challenge 2014). The network is well-known due to its simplistic network architecture. It uses a simple 3×3 convolutional layer stack, on top of each other in increasing depth. Here, the number 16 in the name denotes the total layer present in the network. The network consists of 16 layers, where 13 are convolutional layers, and 3 are dense layers. VGG-16 is also widely used in the crowd analysis field as pre-trained front-end architecture. Due to its implementation as front-end architecture in CSRNet (Y. Li et al. 2018), It was able to achieve state-of-the-art results in crowd estimation and counting.

- **Convolutional Layers**

Images removed for copyright reasons

Figure 2.7: Simple example of Convolutional Operation within the convolutional layers.

The core target of the convolutional layer is to extract useful features from the image such as edges, lines, blobs of colour and other visual elements (Heaton 2015).

* **Convolutional operation**[3]**:** When two function I and K produces a new third integral function C which shows the amount of level of overlap of function K as it shifted over the I function then it is called convolutional operation (Haohan Wang et al. 2017).

$$C(x, y) = (I \cdot K)(x, y) = \sum_{M} \sum_{n} I(x + m, y + n)K(m, n) \quad (2.8)$$

In figure 2.7, the leftmost matrix is an input matrix. The middle is *kernel*[4] matrix. When convolutional operation is applied with the left and middle matrix, we get the rightmost matrix. The operation is an element-wise product followed by sum.

2.4.2 Activation function

All the convolutional layers are linked with activation functions. In figure 2.5, we show that weighted sum is fed to activation for a single block of ANN. The activation function is the key component in the convolutional layer which enables it to approximate the non-linear or complex functions. Without the activation function, neural network outputs would be a simple linear function (Sharma 2017). One of the key activation functions in Deep Learning are **Sigmoid** and **ReLU**.

[3] Also known as *kernels*
[4] Also called a *convolutional filters*

– **Sigmoid activation function:** Sigmoid is a most common activation function that is used in Deep Learning networks. Sigmoid is a nonlinear function and transforms the input value in the range of $\phi(x) \in \{0, \ldots, 1\}$.

$$\phi(x) = \frac{1}{1 + e^{-x}} \tag{2.9}$$

In general, Sigmoid functions are used at the last layer of convolutional networks. When used in other layers, it tends to suffer from the *"vanishing gradients"* problem where the gradient values are too low, and the network seems to stop learning.

– **Rectified Linear Unit (ReLU):** Rectified linear unit (ReLU) was proposed by Nair et al. (2010) and is another widely used activation function. The ReLU function has a range of $\phi(x) \in \{0, ..., \infty\}$

$$\phi(x) = max(0, x) \tag{2.10}$$

In the ReLU function, only positive input values are kept, and the rest are set to zero. It is noted that the ReLU function is much more computationally efficient than Sigmoid. Hence, faster training can be achieved with ReLU.

2.4.3 Pooling

It is a common technique to reduce data size with some local aggregate functions (Hope et al. 2017). As shown in figure 2.6, multiple max-pooling layers have been utilised in VGG-16 architecture. The pooling reduces the size of the data to be processed in the back-end. As a result, it can dramatically decrease the number of overall parameters in the model. In theory, pooling should also make the model robust to the small changes in the network (Hope et al. 2017).

In figure 2.6, VGG-16 applies 5 max-pool layers. The max-pool layer uses a stride of size 2 which means reduction in the dimension width and height

of features maps in half size. For example, the input image has resolution $224 \times 224 \times 3$ after max-pool will be $112 \times 112 \times 128$.

2.4.4 Optimisation

In order to evaluate the performance of CNN learned parameters, a loss function is used. It tells *"how good"* the model is at making predictions. Likewise, to minimise the loss function, optimisation is applied. One of the most common algorithms for optimisation is gradient descent. It is an iterative process which finds the minima of a function. In this case, minima of the loss function.

Let the equation 2.11 be a loss function;

$$f(m, b) = \frac{1}{N} \sum_{i=1}^{n} (y_i - (mx_i + b))^2 \tag{2.11}$$

Then we can calculate Gradient as:

$$f^{'}(m, b) = \begin{bmatrix} \frac{}{dm} \\ \frac{}{db} \end{bmatrix} = \begin{bmatrix} \frac{1}{N} \sum -2x_i(y_i - (mx_i + b)) \\ \frac{1}{N} \sum -2(y_i - (mx_i + b)) \end{bmatrix} \tag{2.12}$$

An iterative process solves the gradient. The iteration takes place on the data points using the new m and b values and partial derivatives are computed. The gradient dictates the next move to update the parameters. It provides a slope for the loss function at current position/direction, and updates are applied based on the learning rate.

2.5 Evolution of Crowd estimation and counting

This section explores diverse crowd estimation methods and counting techniques, starting from detection-based approaches, which were the primary

research topic in the early phase of crowd estimation—followed by a regression-based approach where the crowd analysis topic moved from counting to map-based estimation also known as density map regression. Finally, a newer convolutional neural network-based approach which mainly focuses on density map based methods are discussed.

2.5.1 Detection-based approaches

Methods in this category use pedestrian detection algorithms to locate people in images. Assuming a correct localisation of each person in the scene, crowd counting becomes trivial, and an accurate estimate of the crowd density. In this category of methods, either part of a pedestrian or the person's full body is used. Early approaches of crowd estimation often focused on detection-based approaches with hand-crafted features, and leverage pedestrian or body-part detectors to identify objects and count their number (M. Li, Z. Zhang, et al. 2008; Felzenszwalb et al. 2009; Dollar et al. 2011; Ge and Robert T Collins 2009). The significant aspect of the detection-based methods is a sliding window-based approach applied to images. These approaches require a well-trained classifier to extract low-level features from the human body, such as the Histogram Oriented Gradients (HOG)(Dalal et al. 2005) or the Haar wavelets (Paul Viola et al. 2004) to perform crowd counting. However, the performance degrades when congested scenes are analysed or when most of the targets are occluded. Moreover, these methods are limited by occlusions and cluttered backgrounds.

2.5.2 Regression-based approaches

Considering that detection-based methods' performance degrades when a scene is highly congested, alternative methods such as the regression-based approach have been employed. Techniques have been proposed to directly learn a mapping from features of image patches to the density of a local region

(K. Chen et al. 2012). Additional lower level features have been generated using foreground and texture patterns (Antoni B Chan and Vasconcelos 2009). Idrees, Saleemi, et al. (2013) proposed a method to fuse the features extracted from Fourier analysis, Scale-Invariant Feature Transform (SIFT) (Antoni B Chan and Vasconcelos 2011) and head detection. Lempitsky et al. (2010) applied a technique of linear mapping between the extracted features and density map. Furthermore, Pham et al. (2015) observed that linear mapping approaches have performance limitations. Therefore it was proposed using random forest regression techniques to learn the non-linear mapping between the local region and density maps.

2.5.3 CNN-based approaches

Convolutional Neural Network (CNN) provides superior performance in visual classification and recognition tasks, including crowd estimation problems. Recently, numerous works have been proposed focusing on the varieties of CNN approaches for crowd counting and density estimation[5]. (C. Wang et al. 2015; Fu et al. 2015) adopted a CNN based method for the crowd estimation problem. A deep CNN regression method is proposed in (C. Wang et al. 2015), where the AlexNet architecture (Krizhevsky et al. 2012) is adapted, replacing the last layer with a single neuron for crowd counting prediction. The adapted network is trained with negative samples such as buildings and trees without humans present in the captured scenes. Unlike (C. Wang et al. 2015), (Fu et al. 2015) approached crowd counting as a classification problem. They proposed to divide the crowd into five different classes, such as very high, high, medium, low, and very low density. They utilise (Sermanet et al. 2012) method of multi-stage ConvNet to tackle the shift, scale and distortion problem. Furthermore, they employ two classifiers for better results, where one classifier sampled the misclassified images, whereas the other reclassified

[5]Qiu et al. (2019), D. Kang and A. Chan (2018), Hanhui Li et al. (2018), Onoro-Rubio et al. (2016), Walach et al. (2016), Lingbo Liu, Qiu, et al. (2019), Q. Wang et al. (2019a), Yuan et al. (2015), Kok et al. (2017), and Zhou et al. (2020)

the rejected samples. In C. Zhang, Hongsheng Li, et al. (2015), the analysis was conducted on existing methods and concluded that their performance degrades when new scenes are tested which are different to the training dataset and showed existing networks were not generalised well. Therefore, they proposed a method to learn the mapping from images to crowd count and fine-tune the mapping to new target scenes. They trained the network with two objective functions: crowd counting and density estimation. By training the network alternatively, the goal is to obtain better local optima. The network is further fine-tuned with the existing training samples, similar to the target scene, to generalise new scenes. The important characteristic of this approach is that no new data is introduced to the network. They also proposed to use perspective information to generate ground truth density maps, which makes the network more robust to scale and perspective variations. Inspired by (C. Zhang, Hongsheng Li, et al. 2015), Walach et al. (2016) proposed a method to perform layered boosting and selective sampling. The process iteratively adds CNN layers to the architecture, where each layer is trained to approximate the residual error of the earlier estimation. The layered boosting is based on gradient boosting machine (Friedman 2001), a subset of ensembles techniques. In contrast to previous approaches, which employ the patch-based training method, Shang et al. (2016) proposed a method which utilises the entire image for crowd counting. Also, the method reduces the overall complexity of network architecture. The network simultaneously learns to estimate the local counts and can be viewed as learning a patch level counting model which enables faster training. The architecture incorporates three modules, a pre-trained GoogLeNet (Szegedy, Wei Liu, et al. 2015), a long-short time memory (LSTM) decoder and a fully connected layer. The GoogLeNet is used to compute the higher dimensional feature maps from the crowd image. The LSTM module is then used as a decoder for local blocks to extract features for local crowd counting, followed by the fully connected layer, which maps the LSTM local count to the global estimate. To capture the

semantic information in a crowded scene, Boominathan et al. (2016) proposed a network, which combines deep and shallow convolutional networks. Such architecture produces better results even in the presence of wide-scale and perspective variation. Motivated by the success of multi-columns architecture for image recognition (Ciregan et al. 2012), (Yingying Zhang et al. 2016) propose a similar network MCNN for the random crowded images. The network consists of three convolutional layer columns to ensure robustness in large scale variation. These columns are composed of different filter sizes to capture scale variations. Besides, (Yingying Zhang et al. 2016) also proposed a new technique to generate the ground-truth density maps. In contrast to existing practices, where sums of Gaussian kernels with fixed variance or perspective maps are applied, Zhang suggested considering the perspective distortion by estimating the spread parameters of the Gaussian kernel on the size of the head of each person within an image. However, in practice, they used vital information observed in a highly dense crowd, where people's head size is correlated with the distance between the centre of two neighbouring persons, to generate the density map. The key difference is that the method does not require the perspective maps to employ the perspective distortion information in the ground truth density map. A handful of other works have made further improvements on MCNN (D. Kang and A. Chan 2018; Vishwanath A Sindagi et al. 2017a; Vishwanath A Sindagi et al. 2017b; Walach et al. 2016) to cater to the scale problem. Similar to the above approach, Onoro-Rubio et al. (2016) proposed a scale aware counting model called Hydra CNN. Inspired by the Guanbin Li et al. (2015) work, they designed the network firstly by developing a deep fully convolutional network, which they called Counting CNN (CCNN) based on the observation of earlier work (C. Zhang, Hongsheng Li, et al. 2015; Loy et al. 2013) that incorporated the perspective information for geometric correction of the input feature maps. Secondly, they designed a Hydra CNN architecture that consists of 3 heads. Each head learns feature maps for a particular scale. Then the features are concatenated and fed to the body. The

body consists of two sets of fully connected layers, which are later concatenated to estimate the density map. In contradiction to (Yingying Zhang et al. 2016) architecture, where all the network columns are trained for all the input patches, (Sam, Surya, et al. 2017) discussed that the performance could be improved by training the network columns with a particular set of training patches. Therefore, proposed a network called switching CNN that adaptively selects optimal regressors suitable for the particular input image patch. The network architecture is similar to multi-column network (Yingying Zhang et al. 2016) combined with a multi independent regressor with variant receptive field and switch classifier. Here, the images are sampled in a grid form, and the switch classifier is trained to select appropriate columns for input, whereas the multi-columns CNNs are trained on the patches. The switch classifier and the independent regressors are alternatively trained. Similar to (Yingying Zhang et al. 2016; Onoro-Rubio et al. 2016), (Kumagai et al. 2017) proposed the Mixture of CNNs (MoCNN) architecture based on the previous observation, where single column architecture is not sufficient to estimate crowd density. The proposed network employs the combination of "expert CNN" and a "gating CNN " that adaptively selects the suitable CNN among the experts based on the input image's appearance. For estimation purposes, the expert CNN approximates the crowd count based on the input image, while the gating CNN calculates the appropriate probability for each of the expert CNNs. These probabilities are used as weighting factors to compute the weights' average of the count prediction by all the expert CNNs. Encouraged by the results achieved by simulation learning in (R. Ranjan et al. 2017; Yi et al. 2016), (Vishwanath A Sindagi et al. 2017a) and (Marsden et al. 2017) investigated the multi-tasking learning to improve the network performance. (Marsden et al. 2017) employed ResNet-18 (He et al. 2016) architecture for simultaneous crowd counting, violent behaviours detection and crowd density level classification. Sindagi (Vishwanath A Sindagi et al. 2017a) proposed a cascaded CNN network, which simultaneously learns to

Images removed for copyright reasons

Figure 2.8: Density map visualisation of head centred annotation. In the left figure around the top area, more dense crowds are present and the density map highlights it clearly with red colour.

classify the crowd into various density levels and estimate density map similar to (J.-C. Chen et al. 2016). Numerous types of networks exist focused on the scale-invariant problem, a) (Weizhe Liu et al. 2019; Tian et al. 2019; Ze Wang et al. 2018; Wu et al. 2019) carried out research infusion strategies for various scale information, b) (Cao et al. 2018; Zeng et al. 2017) studied the multi-blob scale aggregate network, c) (S. Huang et al. 2020; Y. Li et al. 2018; N. Liu et al. 2019; Ze Wang et al. 2018) worked on scale-invariant convolutional or pooling layers (Sam and Babu 2018; Babu Sam et al. 2018; L. Zhang, M. Shi, et al. 2018) studied the automated scale adaptive network. (Y. Li et al. 2018) propose CSRNet, which utilises VGG-16 as a backbone and exploited dilated convolutional layers to enlarge receptive fields to improve network performance. Janet was proposed by (Cao et al. 2018), which employs multi-scale aggregated features to produce better crowd estimation. Besides, other varieties of studies utilises perspective maps (M. Shi et al. 2018), geometric constraints (Cheng et al. 2019; Youmei Zhang et al. 2019), and region-of-interest (ROI) (N. Liu et al. 2019) to improve the counting accuracy.

2.5.4 Density map generation

Instead of regression of a headcount from the crowd image, a density map is used to localise the head and a dense and sparse area with a heat map. The

core reason for using a density map is it preserves more information than a count value.

We generate a density map using geometry-adaptive Gaussian kernels based on (Yingying Zhang et al. 2016). Geometry-adaptive kernels are used to address highly congested scenes. The density map can be generated by blurring each head annotation using a normalised Gaussian kernel as in (Yingying Zhang et al. 2016; Sam, Surya, et al. 2017; Vishwanath A Sindagi et al. 2017b). The geometry-adaptive kernel is defined as:

$$D(x) = \sum_{i=1}^{N} \delta(x - x_i) \times G_\sigma(x), \quad \text{with} \quad \sigma = B\bar{d}_i \tag{2.13}$$

where, x_i is the target object for the ground truth δ. $\bar{d}_i$ is the average distance of k nearest neighbours. The density map was generated using $\delta(x - x_i)$ and the Gaussian kernel with standard deviation σ_i (standard deviation), where x is the position of the pixel in the input image. The value for $\beta = 0.4$ was set according to (Yingying Zhang et al. 2016) with minor changes.

2.6 Datasets and Evaluations

2.6.1 Crowd Datasets and Benchmark

In this section, we describe some publicly available dataset for crowd counting and estimation. We chose to have varieties of datasets in the thesis. Hence, we have used ShanghaiTech and UCF-CC-50 for chapter 3 and 5, whereas other datasets such Venice, Mall and UCSD were used in chapter 6. The other reason to select different datasets was as each chapter dealt with different aspects of the crowd counting methods, we had to choose the right dataset in

(a) Sample image from Shanghaitech Part-A

(b) Sample image from Shanghaitech Part-B

(c) Sample images from UCF-CC-50 dataset

(d) Sample images from Mall dataset

(e) Sample image from Venice dataset

(f) Sample image from UCSD

Figure 2.9: Sample images from Shanghaitech, UCF-CC-50, Mall, Venice and UCSD datasets for crowd counting and estimation

Table 2.1: Overview of the crowd counting and estimation dataset

Dataset	Total Samples	Avg. Image resolution (W×H)	Attributes
Shanghaitech (Part-A)	482	864 × 589	Congested
Shanghaitech (Part-B)	716	1024 ×768	Free Scenes
UCF-CC-50	50	2888 × 2101	Congested
Mall	2000	640 × 480	1 fixed camera
Venice	167	1280 × 720	4 Fixed Scenes
UCSD	2000	238 × 158	1 Fixed Scenes

order to compare our results with other papers. Table 2.1 provides a brief overview of the datasets.

- **Shanghaitech dataset:** The Shanghaitech dataset was introduced by Yingying Zhang et al. (2016). The dataset comprises 330, 165 people with the centre of their head annotated and makes up 1198 annotated images. The dataset further splits into two sub-dataset Part-A and Part-B. There are a total of 482 images in Part-A, consisting of randomly crawled images from the internet. The Part-B contains 716 images captured in the Shanghai metropolitan area. The number of crowds varies considerably between two datasets. Likewise, Part-A contains an average image resolution of 864 × 589 whereas Part-B with fixed 1024 × 768 resolution. Both datasets are further divided into train and test sets. In Part-A, 300 images are part of the training set and remaining 182 images as a test set. Part-B has 400 images for training and 316 for testing purposes.
- **UCF-CC-50 dataset:** Idrees, Saleemi, et al. (2013) collected publicly available images from Flickr and annotated 50 images. The images consist of a highly diverse number of individuals ranging from 94 to 4543 with an average of 1280 people per image, a total of 63705 annotated heads. Likewise, scenes in the dataset also include varieties of events. On average, image resolution is around 2888 × 2101.

- **Mall dataset:** The mall dataset was introduced by K. Chen et al. (2012) and contains 2000 frames captured from a shopping mall. Each frame has a fixed resolution of 640×480; The first 900 frames are used as training frames, and the remaining 1200 frames are used for testing. We follow the predefined settings to use the first 800 frames as the training set and the rest 1200 frames as the test set. The validation set (180 images) is selected randomly from the training set.
- **Venice dataset:** The Venice dataset is a relatively small size dataset and was published by Weizhe Liu et al. (2019). The dataset contains 4 different sequences with a total of 167 labelled images with 1280×720 resolution. Weizhe Liu et al. (2019) proposed to use a total of 80 images for training taken from a single long sequence of images and remaining 3 sequence images for the evaluation purpose. Also, the dataset has the Region Of Interest (ROI) for testing purposes.
- **UCSD dataset:** Antoni B Chan, Liang, et al. (2008) published UCSD dataset and the dataset consists of 2000 grayscale image frames of a stationary camera collected from surveillance video. The dataset is from a video recorded at 10 FPS with dimensions 238×158. The Region Of Interest (ROI) is also provided to ignore irrelevant objects. Following the settings in (Antoni B Chan, Liang, et al. 2008), the frames from $601 - 1400$ were used as the training data and the remaining 1200 frames as test data.

2.6.1.1 Dataset Limitation

All the mentioned datasets have various types of limitations such as incorrect annotation, small sample datasets, or lack of diverse samples in the dataset. Some samples in UCF-CC-50 and Shanghai tech dataset have either incorrect head annotation or missed the head completely. The other limitation includes the sample size and the resolution of images. For example, in the UCSD

database on average, the images are only 238×158 resolution. Furthermore, datasets such as Mall and UCSD do not have diverse scenes and are captured only from a single point of view.

2.6.2 Evaluation Metrics

Mean Absolute Error (MAE) and Mean Square Error (MSE) are the two most common evaluation metrics used in crowd estimation and counting fields. Unlike image comparison where it is common practice to use PSNR and SSIM to evaluate the quality of the image, in-crowd estimation instead of the direct pixel-wise comparison of ground-truth density map and predicted density map, the performance of the model is evaluated based on the sum of density pixel value which is an estimated crowd count.

- **Mean Absolute Error (MAE):**

  It measures the average magnitude of the errors in a set of predictions, without considering their direction. It is the average over the test samples of the absolute difference between prediction and actual observation where all individual differences have equal weights.

$$\mathbf{MAE} = \frac{1}{N}\sum_{1}^{N}|y_i - \hat{y}_i| \tag{2.14}$$

  where y_i is the ground truth density map and $\hat{y}_i$ is the density map learned by the proposed network. And N is the number of samples.

- **Mean Square Error (MSE):**

  MSE measures the average squared difference of estimated value and actual value. Here, MSE evaluates trained models' performance, and the value is always positive, and values closer to zero are considered better results.

$$\mathbf{MSE} = \sqrt{\frac{1}{N}\sum_{1}^{N}(y_i - \hat{y}_i)^2} \tag{2.15}$$

Where y_i is the ground truth density map and $\hat{y}_i$ is the density map learned by the proposed network and N is the number of samples.

2.7 Summary

In this chapter, we examined a number of techniques that are involved in the crowd analysis field. We started the chapter with popular techniques that are involved in the feature extraction process; then we investigated the additional methods *classifier* that are used in conjunction with the feature extraction process to improve the crowd analysis performance. Then, the core aspect of Deep Learning components, which makes up Deep Learning techniques were discussed. Further topics on crowd estimation and counting were explored, mainly focusing on the evolution of crowd estimation methods. Finally, we examine the publicly available crowd estimation and counting dataset. The core goal of the chapter was to introduce the essential tools and techniques applied in crowd analysis. In the next chapter, we discuss our contribution toward the problem of the inadequate dataset in the crowd analysis field and propose a synthetic data generation tool that tries to mitigate the problems.

He who lives in harmony with himself lives in harmony with the universe

— Marcus Aurelius

3

Synthetic data generation for scene analysis

Contents

In chapter 1, we introduced the challenges that lie in the crowd analysis field. The primary difficulty in training a Deep Learning network is its requirement

of large amounts of data for generalisation. However, due to the labour-intensive task of annotating thousands of heads in a single image, there are only a limited number of publicly available crowd estimation and counting datasets. Such as UCF-CC-50 (Idrees, Saleemi, et al. 2013) has only 50 images for both testing and training purposes. Hence, we proposed a tool to handle such issues. The tool is able to generate a large quantity of datasets in a short time.

3.1 Multi-Purpose synthetic data generation

Although in recent years Deep Learning (DL) has seen huge increases in research, the requirement of a large amount of data to train is a major blockade. It is a time-consuming and expensive task. It typically involves collecting and manually annotating a large amount of data for supervised learning. This requirement becomes more difficult when the data acquisition process requires domain expertise or data that cannot be captured in large quantities and sufficient quality at a given time or cost. As a result, accurate crowd density estimation, 3D human pose estimation are lagging as creating datasets for such problems at large scale is expensive. Likewise, it is not possible to annotate in detail real-world data: a human cannot manually enter a pixel-accurate flow field. Similarly, even when there are existing datasets, there might be some imbalance where one class has more examples than another class. This leads to biased outcomes. Furthermore, a recent study by (Hestness et al. 2017) indicated that the current DL might not be limited by the algorithms themselves but by the type and amount of supervised data available. Therefore, improvement is needed not only on the algorithms but also on the data generation, both for learning and qualitative evaluation. Deep learning networks (DL) dominate the state-of-the-art results in computer vision (CV) and other fields. One of the primary reasons why DL outperforms existing algorithms is that these produce superior results when more labelled

data are used. Nonetheless, it is well known that DL requires a large quality of data to generalise well. Collecting and labelling these datasets are expensive, time-consuming and sometimes impossible. Therefore, researchers tried to use alternative techniques, such as graphics simulators to automatically generate labelled datasets. However, these techniques are still expensive and require domain knowledge to produce good datasets. In this chapter, therefore, a graphics simulator is presented which automatically generates multi-model datasets in real-time providing the corresponding ground truth and annotation. The tool concentrates on pedestrian and crowd analysis including 3D human pose estimation, pedestrian detection as well as crowd density and flow estimation.

A good approach for tackling such limitations is to utilise synthetic data simulators to generate labelled data automatically. A number of synthetic datasets have been published such as Flying Chairs (Dosovitskiy et al. 2015), MPISintel (Butler et al. 2012), SceneNetRGB-D (McCormac et al. 2016), among others. These datasets are expensive to generate, requiring artistic knowledge to meticulously design specific environments. These datasets have been proven to be successful in training and testing networks for geometric problems such as optical flow, pose estimation, classification and segmentation. Synthetic data is widely used in research. (Hattori et al. 2015) presented a scene-specific pedestrian detector using only synthetic data. A synthetic dataset of human bodies was published by (Varol et al. 2017), where datasets were used to estimate the human depth and part segmentation from RGB-images. Similarly, synthetic data for 3D human pose estimation were used by (W. Chen et al. 2016). Likewise, (d. Souza et al. 2017) used synthetic video for human action recognition with deep neural networks.

Most of the above techniques discussed either dealt with real or synthetic data only and only a number of papers considered the method of training models with synthetic and real data together. (Marín et al. 2010) used synthetic humans to detect pedestrians. (Cheung et al. 2018) used a mixed reality

dataset which is composed of real-world background images and synthetically generated static human-agents for pedestrian detection. (Pishchulin, Jain, et al. 2011) used synthetic human bodies rendered on random backgrounds for training a pedestrian detector.

Although all the above worked with synthetic data, which requires labour-intensive 3D environment models, we concentrate on augmented reality where we utilise the synergies of real and synthetic data. In contrast to a purely synthetic dataset, we obtain a large variety of realistic data efficiently. Furthermore, as shown by our experiments in section 3.5, combining real and synthetic data within the same image results in models with better generalisation performance. Unlike existing mixed-reality approaches for training data generation which are either simplistic where they consider single objects or augmented objects in front of random backgrounds. Our goal is to create high fidelity augmentations of complex multi-object scenes at high resolution in real-time.

Moreover, all the existing datasets have fixed scenarios and samples. However, we are presenting a dataset generator tool which synthetically generates realistic images of humans superimposed on real scenes and can be used for any number of scenarios and samples. In addition, the results are generated almost in real-time as the tool utilises a real-time 3D rendering engine. The tool is mainly focused on computer vision problems such as 3D human pose estimation, pedestrian detection, and crowd estimation.

3.2 Methodology

In this section, the core aspects of the proposed algorithm are described. The section first details the perspective ground plane extraction, followed by pedestrian and crowd simulation and finally the synthetic data generation. Figure 3.1 shows flow of proposed methodology.

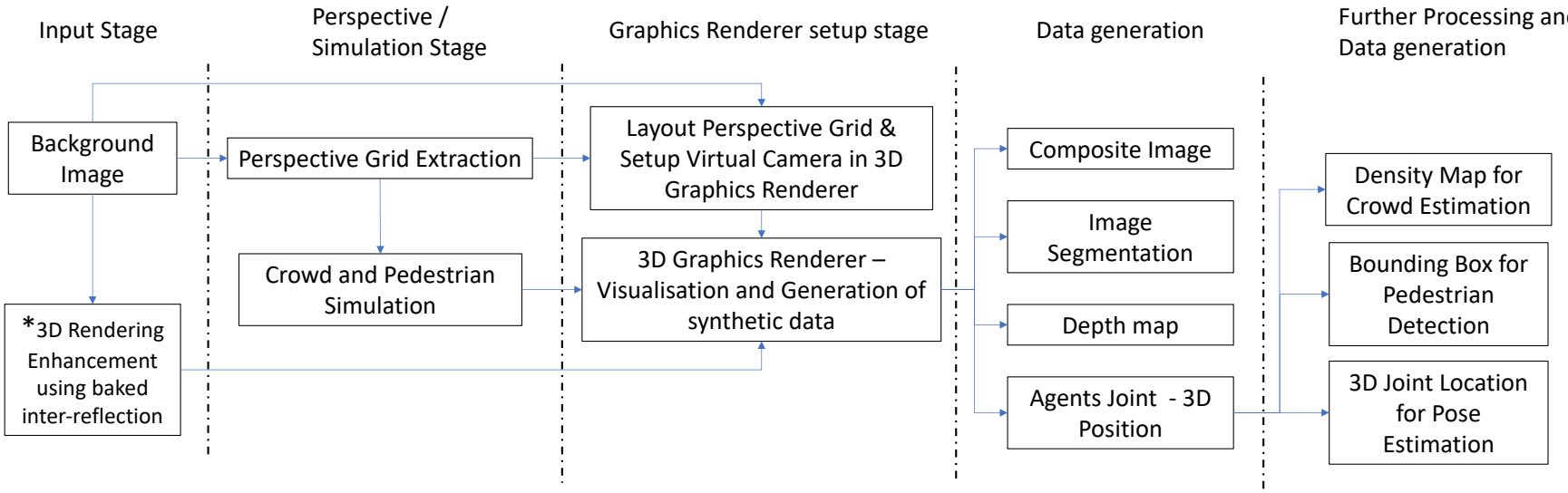


Figure 3.1: An overview of the proposed method. Our approach consists of 3 major stages to generate the data: Input, Perspective/Simulation and Graphics renderer setup stage. *The 3D rendering enhancement using baked inter-reflection is discussed in chapter 4

3.2.1 Perspective Plane Extraction

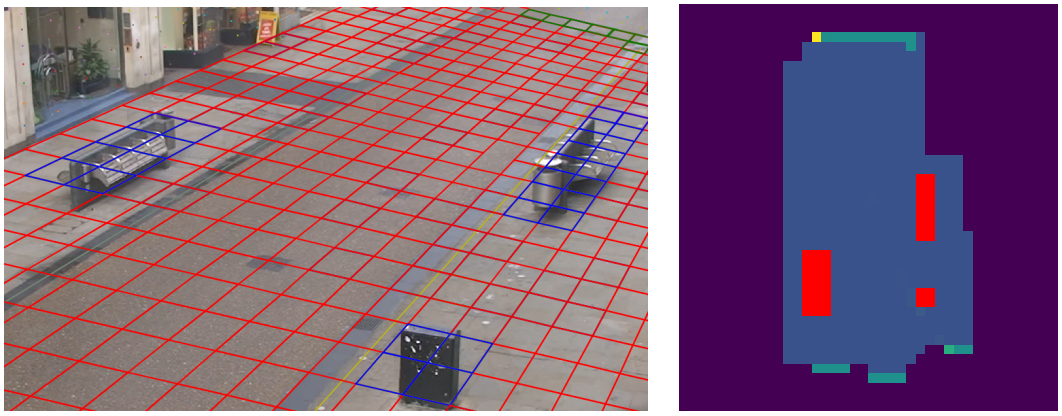
The real-world background image is required to generate augmented datasets. Numerous background extraction methods such as (Bouwmans et al. 2017) can be applied to acquire a clutter-free background image. The most obvious way to extract background is when a series of images are provided, these images can be used to acquire an unobstructed background view (Hua et al. 2018).

Once the initial part of background extraction is completed, the process for the perspective ground grid estimation method is applied. The perspective estimation can be obtained using the concept of perspective scale. When two parallel lines are defined which point towards the vanishing point, the distance in arbitrary units within this perspective space can be measured. More detail can be found in (Dupre et al. 2019). Figure 3.2 (a) and (b) shows the extracted perspective and top-down view of the Oxford town center dataset (Benfold et al. 2011).

3.2.2 Pedestrian and Crowd Simulation

A social force-based model is implemented (Karamouzas et al. 2009). The model simulates simple crowd behaviours such as separation, object avoidance and agent collision detection based on their field of view.

For the algorithm to simulate a crowd, key attributes such as the number of agents, frame rate, minimum distance (total number of cells in the grid) covered by agents and their speed are provided by the user. Using the extracted perspective grid, the algorithm automatically calculates the entrance and exit for each agent based on the shortest path, assigns the agent's radius (i.e. the space between each agent in the simulation), acceleration and rotational velocity. Finally, the social force simulation model (Karamouzas et al. 2009) utilises this information to simulate the agents. To ensure that the agent speeds are consistent between different simulation models and environment,



(a) Perspective Plane **(b)** Top Down View

Figure 3.2: Perspective Plane Extraction

the simulation is set to a standard frame rate (30 frames per second) and the agent position and rotation are recorded at each frame. The generated information is later used in a graphics simulator to position the agents in 3D space for synthetic data generation purposes.

3.3 Type of generated data - Primary data

The graphics simulator is used to generate datasets based on the total number of simulated frames and generates four different types of data. 1) composite image, 2) 3D agent's joints location, 3) Image segmentation and 4) depth map.

3.3.1 Composite Image

The composite image is generated by superimposing the synthetic agent on top of the real-world background. An arbitrary image size can be set for the final results. Figure 3.3 (a) shows a sample of the final composite image.

3.3.2 3D Joints

The 3D joints location, their unique IDs for each agent are captured during the data generation process. Total of 18 different joint locations were captured

as shown in figure 3.3(d). The joint locations were selected based on the popular benchmarking dataset called MPII Human Pose Dataset (Pishchulin, Andriluka, et al. 2013).

3.3.3 Image Segmentation

Image segmentation of the crowd is also generated during the capture process where the background is set to black colour and each individual agent is assigned a unique colour. The figure 3.3 (b) shows the results of image segmentation.

3.3.4 Depth Map

A depth map describes how far (per pixel) an agent is from the camera. The map is generated in the range of $0 - 0.5$ value where 0.5 is the most distant and 0 is the most closed. And value 1 indicates the background. Figure 3.3 (c), white background refers to a very far distance.

3.4 Other varieties of data generation - Secondary data

The agent joints information is further processed to generate other labelled datasets. Such as a dataset for crowd counting from the head location of each agent. The bounding box for pedestrian detection, and finally, 3D joint location is used to generate data for pose estimation as shown in figure 3.4.

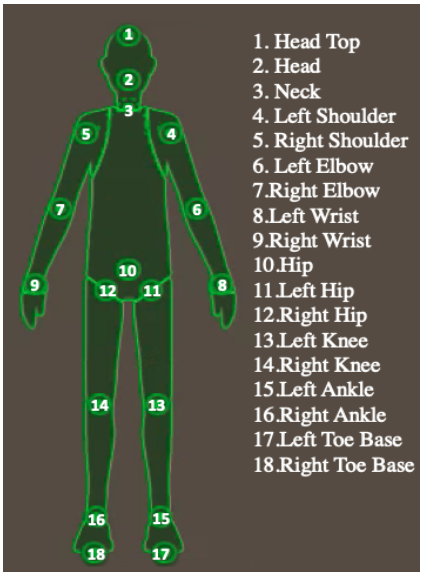
3.4.1 Density Map for Crowd estimation

In section 2.5.4, we introduce the density map generation process and its importance. We also incorporate techniques to generate density maps as secondary data from synthetic agent head locations.

(a) Composite Image

(b) Segmented

(c) Depth map



(d) Agent Joints Location

Figure 3.3: Sample data generated by propose tool and visualisation of all the joints that are capture during the data generation process

The density map utilises equation 2.5.4 to generate a map from head location. Figure 3.4(b) shows the sample generated density map from the datasets. As you can see, the density map is not only used to show the density variance in the scene but also used for actual estimation of the total number of people in the crowd. For the purpose of data generation, we used fixed $\sigma_i = 4$

3.4.2 The bounding box for pedestrian detection

To generate bounding boxes for the pedestrian detection, we utilise the head top, left toe base, right toe base joints for the height of the agent and left

shoulder, right shoulder joints for the width of the box. The bounding box is commonly represented in two different ways.

- minimum and maximum pixel location of the box,
- minimum pixel location, width and height value.

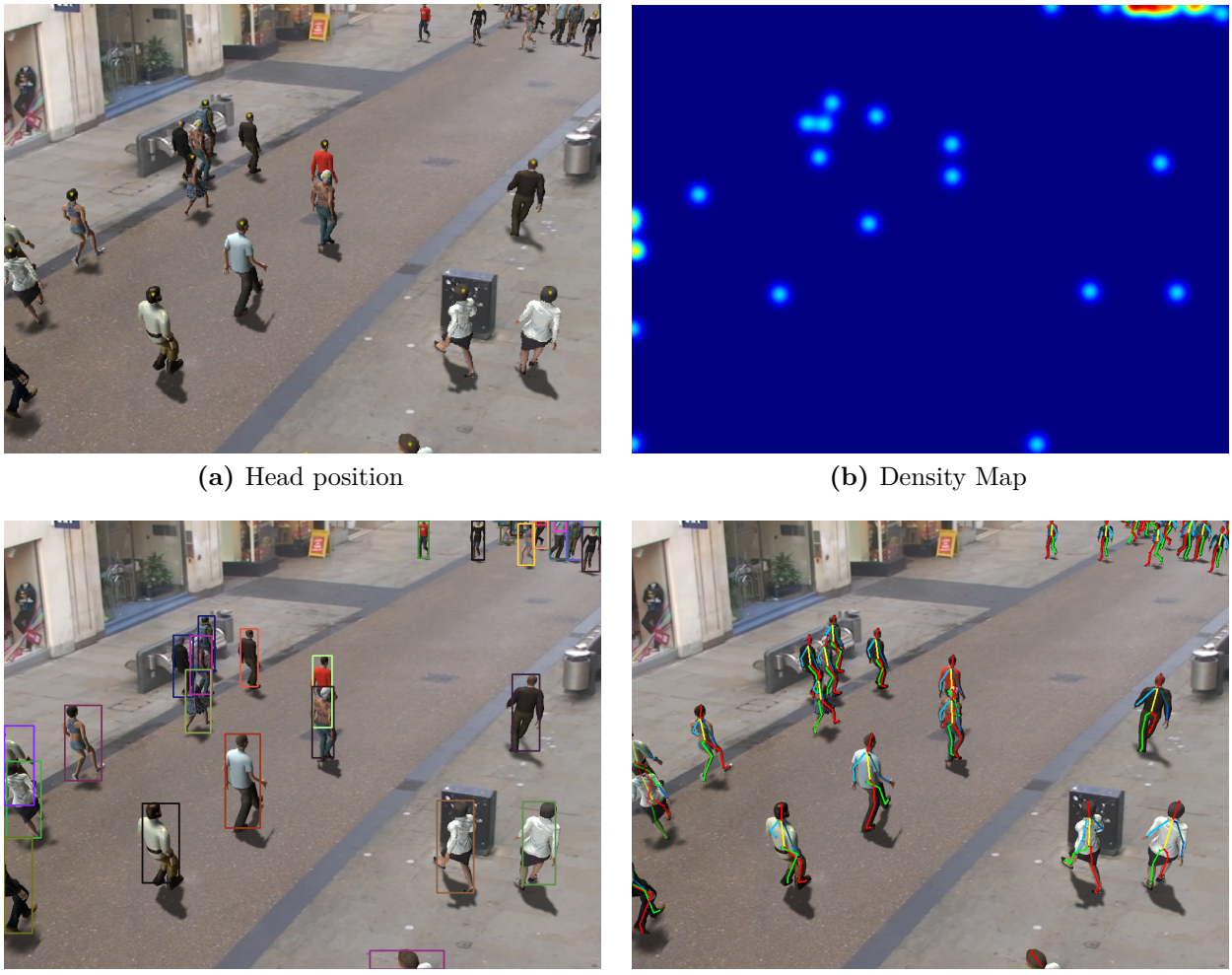
3.4.3 3D Joint location for Pose estimation

The key joint location points in the synthetic agents were selected based on the established benchmarking dataset called MPII Human Pose Dataset (Pishchulin, Andriluka, et al. 2013). The dataset also consists of 18 different key points similar to the figure 3.4(d). However, there are other datasets such as MS coco dataset (T.-Y. Lin et al. 2014) which provides additional key points for faces. We primarily focused on the body and head key points instead of faces. The 3D pixel location of the joints are generated automatically and no further procedures are required to generate label data for pose estimation.

3.5 Experiments and Analysis

To demonstrate effectiveness of our approach, we choose to validate it in two different Deep Learning network architectures. The purpose of different architecture is to analyse if our methods improve the network performance even in the presence of completely different network architecture. For the evaluation metrics, we choose the commonly used metrics Mean Absolute Errors (MAE). MAE is widely used in various crowd counting benchmarking datasets (V. Sindagi et al. 2017b; Idrees, Saleemi, et al. 2013).

We conducted experiments on CMTL by (V. Sindagi et al. 2017b) and CRSNet by (Y. Li et al. 2018). Likewise, two publicly available datasets Shanghai Tech (V. Sindagi et al. 2017b) and UCF-CC-50 (Idrees, Saleemi, et al. 2013) were used for the experiment.



(a) Head position

(b) Density Map

(c) Pedestrian BBox

(d) Pose Estimation

Figure 3.4: Other varieties of labelled data generated from joint information

The Shanghai Tech (SHT) database was introduced by Yingying Zhang et al. (2016) and it contains 1198 annotated images with a total of 330,165 people. The dataset is divided in two parts. Both parts are further divided into training and test sets. Part-A has 482 images where 300 images are used for training. Whereas Part-B has 716 images where 400 images are used for training.

Similarly, the UCF-CC-50 dataset was introduced by Idrees, Saleemi, et al. (2013). The dataset contains 50 annotated images of extremely dense crowds. However, density in the dataset varies between 94 and 4543 persons with an average of 1280 persons per image.

3.5.1 Evaluation Metrics

For the evaluation purpose the standard metrics Mean Absolute Error (MAE) was used, defined as:

$$\mathbf{MAE} = \frac{1}{N}\sum_{1}^{N}|y_i - \hat{y}_i| \tag{3.1}$$

Where y_i is the ground truth density map and $\hat{y}_i$ is the density map learned by the proposed network. And N is the number of samples in the dataset.

Finally, these generated data are fed into both CMTL (V. Sindagi et al. 2017b) and CSRNet (Y. Li et al. 2018) for evaluation.

3.5.2 Training and implementation

The training and evaluation was performed on the NVIDIA GTX TITAN-X GPU using the Pytorch framework. While we tried to leave all other settings in the code as published by the author, we used the well known *Adam*(Kingma et al. 2015) optimiser instead of the standard Stochastic Gradient Descent (SGD) (Sutskever et al. 2013). CMTL (V. Sindagi et al. 2017b) and CRSNet (Y. Li et al. 2018).

The Adam optimiser also known as Clipped SGD is an adaptive variant of SGD. Since Adam optimiser is SGD, the primary reason to choose Adam over SGD was mainly due its less memory requirements for training as well as faster convergence. It has been shown (Wilson et al. 2018) that either of the optimisers will lead to similar coverage with enough training epochs. In another paper (Luo et al. 2019), it suggested that SGD generalizes better than other adaptive optimization methods such as Adam. Hence, it can be said that choosing one optimizer over another is simply the authors' preference and does not affect the overall outcome of the deep learning network.

For the training, we generated 8000 images. These images included 8 different scenes with varying time (different day and night time). Some examples of

Table 3.1: MAE for CMTL and CSRNet, with and without synthetic data (*lower value are better)

Method	SHT Part-A	SHT Part-B	UCF-50
CMTL (without)	101.3	20.0	322.8
CMTL (with)	88.06 ↓	17.0↓	300.2↓
CSRNet (without)	68.2	10.6	266.1
CSRNet (with)	42.81 ↓	6.49↓	245 ↓

generated images are included in appendix A. In addition, the images were further augmented by slicing it into 9 small patches. The groundtruth density maps were generated for each patch based on equation (2.5.4). The single patch is $\frac{1}{4}^{th}$ size of the original image. These images are then formatted based on the CNN requirement. For CMTL (V. Sindagi et al. 2017b) the images were converted into grayscale whereas for CSRNet (Y. Li et al. 2018) images were in RGB format. No additional data augmentation has been applied to patches.

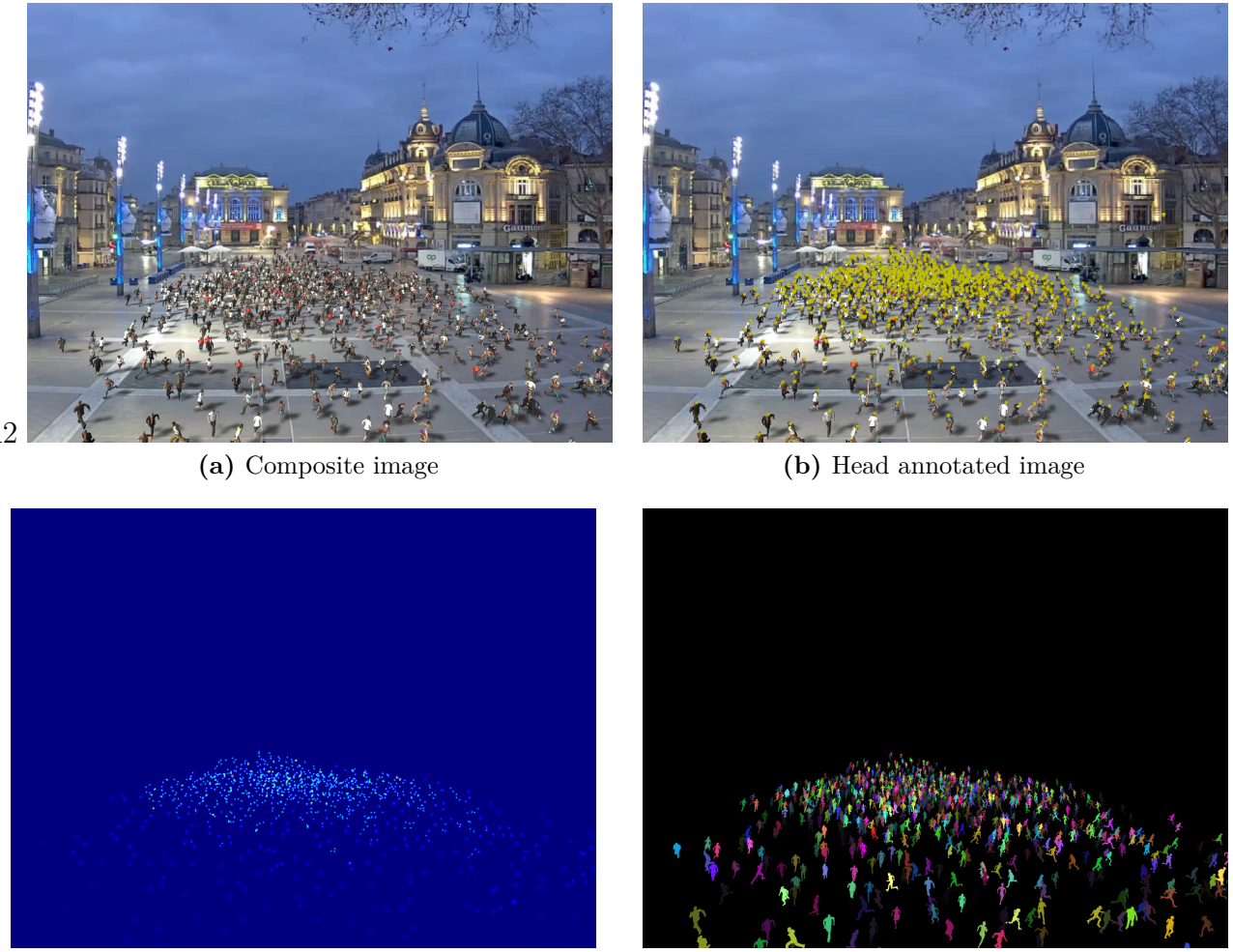
3.5.3 Results and Discussions

The evaluation was carried out with the two networks CMTL(V. Sindagi et al. 2017b) and CSRNet (Y. Li et al. 2018) with both real and synthetic datasets. Table 3.1 shows the network's results with and without the use of synthetic data. The results for the real dataset (*without) are directly from the authors paper (V. Sindagi et al. 2017b; Y. Li et al. 2018). The results demonstrated the overall improvement in both networks after pre-training with a synthetic dataset and fine-tuning with the real world dataset, in spite of differences in network architecture. CMTL (V. Sindagi et al. 2017b) takes grayscale images as input, whereas CSRNet (Y. Li et al. 2018) takes in an RGB image as input. It can be observed that CSRNet (Y. Li et al. 2018) improves more than CMTL (V. Sindagi et al. 2017b). It can also be observed that it's beneficial to have an RGB image as input rather than grayscale image. However, further studies are required to validate the benefit of the RGB over grayscale image

as input. Figure 3.5, shows images generated by the proposed tool where up 1200 agents were simulated.

Although we demonstrated the improvement in the overall performance of the deep Learning network, few key points should be noted about the generated data. Various cases such as people wearing numerous types of clothing such as hats were not considered. While these factors might have improved the performance of deep learning networks even more, no significant degradation in the performance of the network were seen without it. In addition, people of different sizes, colours and disable people such as people in wheelchairs were also not included in the experiments, mainly due to the nature of the crowd counting annotation method. The annotation is carried out only in the heads region and other parts are generally disregarded.

l2



(a) Composite image

(b) Head annotated image

(c) Crowd Density map

(d) Image segmentation

Figure 3.5: Sample image generated by the proposed synthetic data generation tool. 1200 agents were simulated in the image. More images can be found in appendix A.

3.6 Summary

In this chapter, we proposed a method which can be used to mitigate the problem that exists in the crowd analysis field of not having enough dataset to train the Deep Learning networks. Not only that, we also proposed a general approach which can be used to generate data required for other fields of computer vision such as pedestrian detection, 3D pose estimation, image segmentation and depth estimation. By taking advantage of augmented data, we demonstrated that the state-of-the-art results can be considerably improved despite the difference in the network architectures. Furthermore, our approach tackles the problem such as training overfitting, as well as dataset accuracy by generating high-quality synthetic data.

In the next chapter we further look into the aspect of improving the quality of synthetic data generation by removing a phenomenon called inter-reflection to generate better synthetic dataset.

"Education is not the filling of the pail, but the lighting of the fire."

— William Butler Yeats

4

Scene and crowd analysis using synthetic data generation with 3D quality improvement

Contents

In the previous chapter 3, we looked into the method of the synthetic data generation process and proposed a tool to achieve the goal. In this chapter, we focus on a light property called inter-reflection also known as global illumination. Inter-reflection is a major problem in composite images as unwanted colour bleeding occurs to the object from the environment. In addition, it is quite difficult to remove the inter-reflection from the object, especially from the reflective objects such as metals. For example, it is widely

common to use a green screen background in films and green-screen keying is an essential part of post production workflow. The process is labour and cost intensive as it requires expertise and time. It is even more difficult due to inter-reflection (i.e., green background colour) occurring in various objects in the scene (Aksoy et al. 2016). In the figure 4.1, we see the effect of inter-reflection where green colour bleeds into the reflective metal armour.

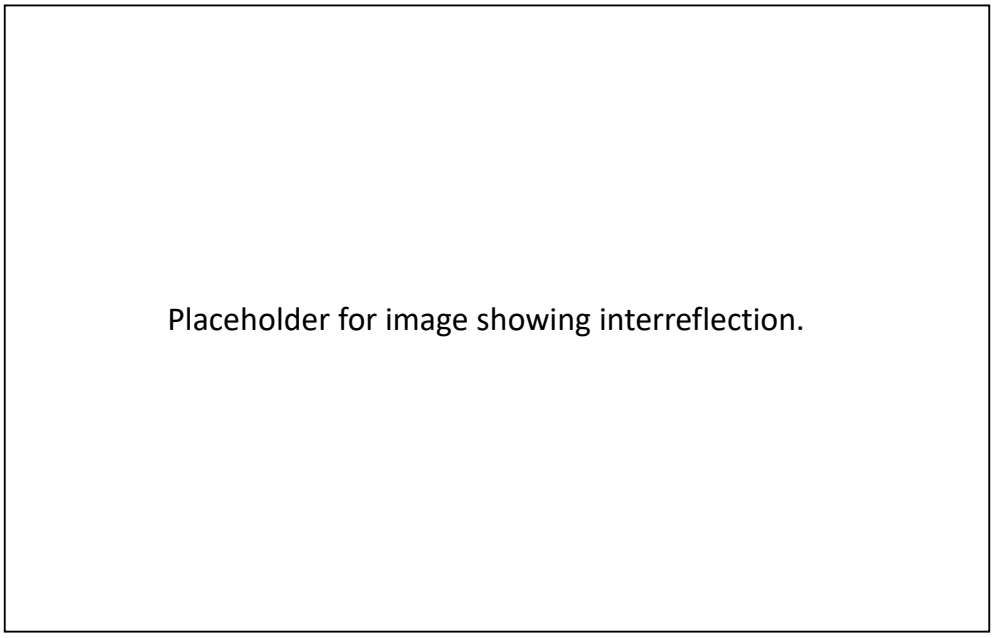


Figure 4.1: Example of inter-reflection that occurs during the film production (Insider 2020)

Photometric stereo (PS) methods for 3D reconstruction recover the shape and reflectance properties of an object using multiple images taken with variable lighting conditions from a fixed viewpoint. PS assumes that a scene is illuminated only directly by the illumination source. As a result, indirect illumination effects due to inter-reflections introduce strong biases in the recovered shape. Our suggested approach is to recover scene properties in the presence of indirect illumination. To this end, we proposed an iterative PS method combined with a reverted Monte-Carlo ray tracing algorithm to overcome the inter-reflection effects aiming to separate the direct and indirect lighting. This approach iteratively reconstructs a surface considering both

the environment around the object and its concavities. We demonstrate and evaluate our approach using three datasets and the overall results illustrate improvement over the classic PS approaches.

4.1 Motivation

In this chapter, we present a method which examines the inter-reflection phenomena occurring due to the concavities, proposing a novel approach to extract and remove inter-reflection. Furthermore, in order to demonstrate the effectiveness of our approach, reconstruction methods such as Photometric Stereo were selected for inter-reflection colour, intensity map generation and the evaluation. Our method accounts for inter-reflections in a calibrated photometric stereo environment and utilises a reverted Monte Carlo ray tracing method to extract the inter-reflection colour and intensity map. This approach not only accommodates the concave surfaces but also any object in a scene with inter-reflections. The proposed method Iterative Ray Tracing Photometric Stereo - IRT PS iteratively applies Photometric Stereo (PS) and a reverted ray tracing algorithm based on a Monte-Carlo implementation to reconstruct with higher accuracy the observed surfaces. This approach iteratively reconstructs the surface and separates the indirect from direct lighting considering also the environment around the object. In addition, the core principle of inter-reflection extraction approach is that it can be utilised in other scenarios such as generating inter-reflection maps (i.e., baked inter-reflection map similar to light map) which can be precomputed to be used in a real-time environment or be integrated as a shader.

Scene and 3D object reconstruction is the process of capturing their shape and appearance using various methods and approaches such as stereo, structure from motion, shape from shading, and many more (Remondino et al. 2006). The reconstruction is highly applicable in a number of fields as it provides the ability to understand 3D scenes and objects on the basis of 2D images. The

applications range from robotics and automated industrial quality inspection over human-machine interaction (example gesture and face recognition) to films and architectural applications (Herbort et al. 2011). Additionally, the method is commonly used to analyse the surfaces of a celestial object, such as the Moon (Hicks et al. 2011).

Photometric stereo (PS) is a well-established technique that is used for 3D surface reconstruction (Esteban et al. 2008; ou et al. 2009). The approach generally inherits the principle of appearance analysis of a 3D object on its 2D images. Based on the intensity information, these approaches attempt to infer the shape of the depicted object (Herbort et al. 2011). It estimates shape and recovers surface normals of a scene by utilising several intensity images obtained under varying lighting conditions with an identical viewpoint (Argyriou, Petrou, and Barsky 2010; Tankus et al. 2005; Hayakawa 1994). By default, PS assumes a Lambertian surface reflectance; a standard reflectance model which defines a linear dependency between the normal vectors and image intensities. The definition of the model then can be used to determine the 3D space in the image (Belhumeur et al. 1998). However, just a single Lambertian image is not adequate to correctly determine the surface shape. Therefore, the PS uses several images whose pixels correspond to a single point on the object and is able to recover surface normals and albedos (Tan et al. 2008).

Light displays complicated attributes while interacting with objects resulting in direct and indirect illumination as shown in figure 4.2.However, classical PS naively assumes that a scene is illuminated only directly by the emitting source. In presence of indirect illumination, it produces erroneous results with reduced reconstruction accuracy (Ikeuchi 1981). For example, an indirect illumination such as inter-reflections makes concave objects appear shallower (Nayar et al. 1990).

4.1.1 Inter-reflections

An image captured by the camera is the result of a complex sequence of reflections and inter-reflections. When light is emitted from the source, it bounces off the scene's surface one or more times before reaching a camera.

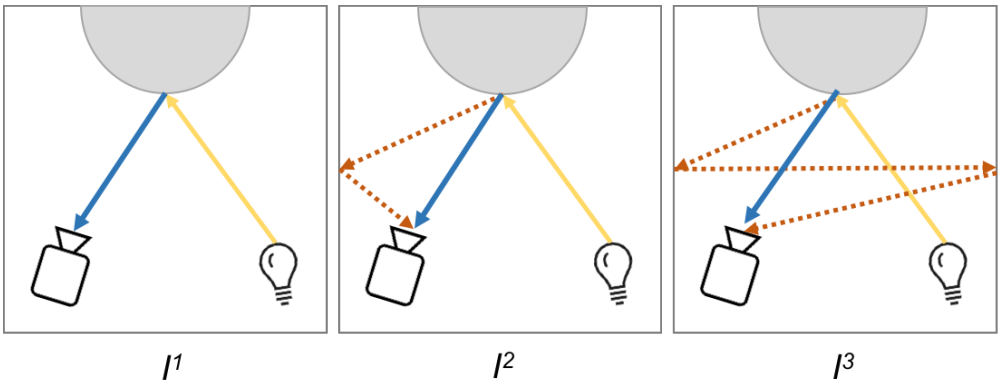


Figure 4.2: (Left)Direct and (Middle)(Right)indirect light bounce around the environment

In theory, every image can be captured as an infinite sum, $I = I^1 + I^2 + I^3 + .. + I^n$, where I^n denotes the total contribution of light that bounces n times before reaching the camera as shown in figure 4.2. For example, I^1 is the captured image if it was possible to remove all the indirect illumination from reaching the camera sensor, while the infinite sum $I^2 + I^3 + \ldots + I^n$ describes the total contribution of indirect illumination. Although we can capture the final image I using a camera, the individual *"n-bounce"* images are not directly measurable in the real-world scenario.

Nevertheless, the techniques for simulating inter-reflections and other light transport effects are not new in computer vision and graphics. The algorithm that simulated the forward light transport was solved by (Kajiya 1986). The algorithm is also known as *rendering equation*. The rendering equation is an integral in which the radiance leaving a point is given as the sum of emitted plus reflected radiance under a geometric optics approximation.

$$I(x,x') = g(x,x')\left[e(x,x') + \int_s p(x,x',x'')I(x',x'')dx''\right] \tag{4.1}$$

Where $I(x,x')$ is related to the intensity of light passing from x' to point x. $g(x,x')$ is a "geometry" term, $e(x,x')$ is related to the intensity of emitted light from x' to x and $p(x,x'x'')$ is related to the intensity of light scattered from x'' to x by a patch of surface at x'.

An algorithm such as ray tracing (Foley et al. 1996; Jarosz et al. 2008) solved the equation 4.1 by using Monte-Carlo methods, whereas radiosity (Foley et al. 1996; Immel et al. 1986) used a finite element method to produce near realistic looking images in the field.

For a Lambertian object illuminated by a light source of parallel rays, the observed image intensity $\mathbf{a}$ at each pixel is given by the product of the albedo ρ and the cosine of the incidence angle θ_i (the angle between the direction of the incident light and the surface normal) (Horn 1977). The above incidence angle can be expressed as the dot product of two unit vectors, the light direction $\mathbf{l}$ and the surface normal $\mathbf{n}$, $\mathbf{a} = \rho\cos(\theta_i) = \rho(\mathbf{l}\cdot\mathbf{n})$.

Let us now consider a Lambertian surface patch with albedo ρ and normal $\mathbf{n}$, illuminated in turn by several fixed and known illumination sources with directions $\mathbf{l}^1$, $\mathbf{l}^2$, ..., $\mathbf{l}^{\tilde{Q}}$. In this case we can express the intensities of the obtained pixels as:

$$\mathbf{a}^k = \rho(\mathbf{l}^k\cdot\mathbf{n}), \quad \text{where} \;\; \mathrm{k} = 1,2,...,\tilde{\mathrm{Q}}. \tag{4.2}$$

We stack the pixel intensities to obtain the pixel intensity vector $\mathbf{A}_a = (\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_{\tilde{Q}})^T$. Also, the illumination vectors are stacked row-wise to form the illumination matrix $\mathbf{L} = (\mathbf{l}^1, \mathbf{l}^2, ..., \mathbf{l}^{\tilde{Q}})^T$. Equation (4.2) could then be rewritten in matrix form:

$$\mathbf{A}_a = \rho\mathbf{L}\mathbf{n} \tag{4.3}$$

If there are at least three illumination vectors which are not coplanar, we can calculate ρ and $\mathbf{n}$ using the Least Squares Error technique, which consists of using transpose of $\mathbf{L}$, given that $\mathbf{L}$ is not a square matrix:

$$\mathbf{L}^T\mathbf{A}_a = \rho\mathbf{L}^T\mathbf{L}\mathbf{n} \Rightarrow (\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T\mathbf{A}_a = \rho\mathbf{n} \tag{4.4}$$

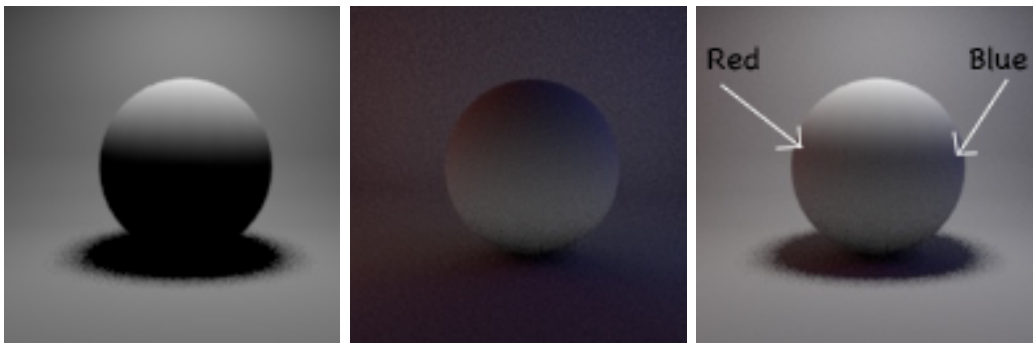
Since $\mathbf{n}$ has unit length, we can estimate both the surface normal (as the direction of the obtained vector) and the albedo (as its length). Extra images allow one to recover the surface parameters more robustly.

4.2 Proposed method to improve the quality of synthetic data

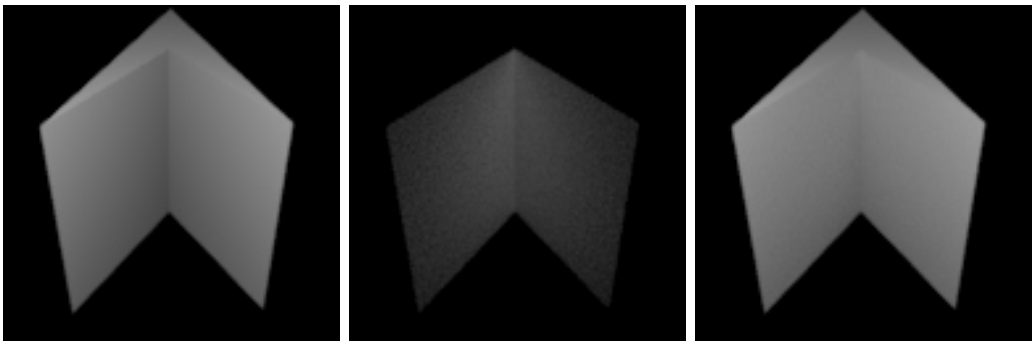
In nature, when we illuminate a surface, light not only reflects towards the viewer but also among all surfaces in the environment. This is always true, with exception to scenes that consist only of a single convex surface. In general, scenes include concave surfaces where points reflect light between themselves. Furthermore, inter-reflections can occur due to the environment and appreciably can alter a scene's appearance. In figure 4.3, to simulate the inter-reflections the sphere is placed within the Cornell box (Niedenthal 2002) and highlights the inter-reflections i.e. sphere receive the colours from its environment.

4.2.1 Inverted ray tracing

Existing computer vision algorithms do not account for effects of inter-reflections and hence often produce erroneous results. The algorithms that are directly affected by inter-reflections are the shape-from-intensity algorithms including Photometric Stereo. Due to the common assumption of single surface reflections (direct illumination) and disregarding higher order (inter-reflections, a subset of global illumination), photometric methods produce erroneous results when applied to open scenes.



(a) (Left)Image with no inter-reflection, (Middle) Image with inter-reflection from Environment only, (Right) Combined Image



(b) (Left)Image with no inter-reflection, (Middle) Image with inter-reflection from Concavity only, (Right) Combined Image

Figure 4.3: Example images of Inter-reflection from environment and concavity

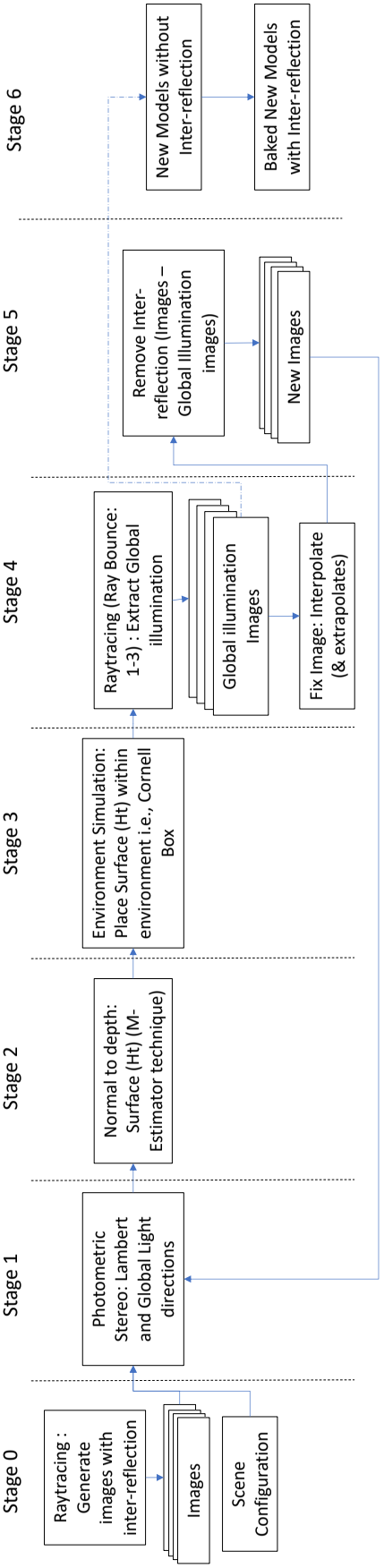


Figure 4.4: An overview of the proposed IRT-PS algorithm from Stage 0 - 5 and additional Stage 6 for inter-reflection baking purposes.

The figure 4.4 shows the proposed pipeline. The first stage (stage 0) of the proposed method is performed only once throughout the process and involves the acquisition of the initial input images. In practice, these images are acquired from the wild and assume that inter-reflections are present and that the captured surface is within the known environment. In our case within a Cornell Box.

Moving to the following stage, PS is applied to the images acquired at stage 0 using equation 4.4 to obtain the initial albedo ρ_t and normals $\mathbf{n}_t$. Integrating over the obtained normals a 3D surface H_t is obtained using the M-estimator technique. This initial surface that is affected by the presence of the inter-reflections becomes the input to the following stage, that involves the proposed reverted ray tracing algorithm.

As environment information is known prior to reconstruction, we can implement our environment. The Cornell Box was set up as the environment at the following stage 3. More realistic textures can be used for the walls without affecting the proposed methodology.

In stage 4, we simulate the environment assuming the Cornell box is given or estimated. In our case, this approach can be extended to other realistic environmental projections such as Hemispherical Dome Projection (Bourke 2005) without affecting the proposed methodology. Then we place the generated H_t surface within this environment.

In the following stage, based on the equation 4.7, the reverted ray tracing algorithm is applied. Since we are only interested in inter-reflections, only the indirect illumination is calculated. To implement the ray tracer for the Lambertian surface, we solve the rendering equation by integrating Monte Carlo estimators.

$$L_0(p, w_o) = \int_{\Omega} f(p, w_0, w_i) L_i(p, w_i) cos\theta_i dw_i \tag{4.5}$$

Where $L_0(p, w_0)$ is the total outgoing radiance reflected at p along the w_0 direction. $L_i(p, w_i)$ is the radiance incident at p along the w_i direction. $f(p, w_0, w_i)$ determines how much radiance is reflected at p in direction w_0, due to an irradiance incident at p along the w_i direction. $cos\theta_i$ is from Lambert's cosine law: diffuse reflection is directly proportional to $cos(\theta)$ of the normals and the incident illumination (i). Finally, $\int_{\Omega} dw_i$ is an integral over a given hemisphere.

Monte-Carlo approximation is a method to approximate the expectation of a random variable, using samples.

$$E(X) \approx \frac{1}{N} \sum_{n=1}^{n} X_n \tag{4.6}$$

Where, $E(X)$ is an approximation of the average value of a random variable X.N is the sample size. And when we integrate it to equation 4.5 we solve the rendering equation.

$$\langle L_0(p, w_o) \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(p, w_0, w_i) L_i(p, w_i) cos\theta_i dw_i}{p(w_i)} \tag{4.7}$$

However, Monte-Carlo estimator is affected by noise, the ray tracer algorithm also inherited such a problem. For example, to half the noise in an image rendered by ray tracing, we need to quadruple the number of samples.

To estimate the environmental colour, we first hit the H_t surface with rays from each pixel, considering techniques such as hemisphere sampling, we randomly reflect the rays toward the environment. As a result, the images of the environment are captured for the various levels/depths of ray reflection. In this study, we only use up to 3 reflection rays (1 to 3) with just a single sampling, as shown in figure 4.5. Because we are not calculating all the ray reflections within the environment, we will have pixel locations without intensity values. An example can be seen in figure 4.6. Therefore, we are using a non-uniform interpolation algorithm (Thévenaz et al. 2000) to approximate

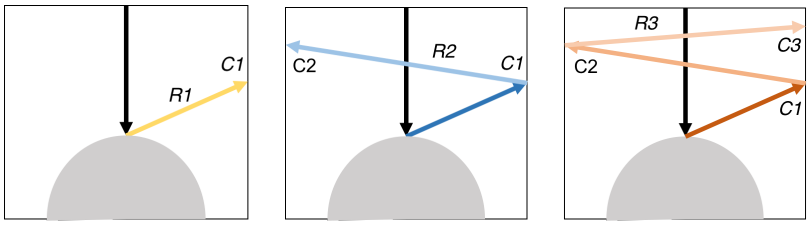


Figure 4.5: Extraction of Environment Intensities in 3 different ways (a) Only extract colour (c1), (b) reflect ray one time and combine the intensities (c1 * c2), and (c), reflect one more time and combine all the colours (c3*c2*c1).

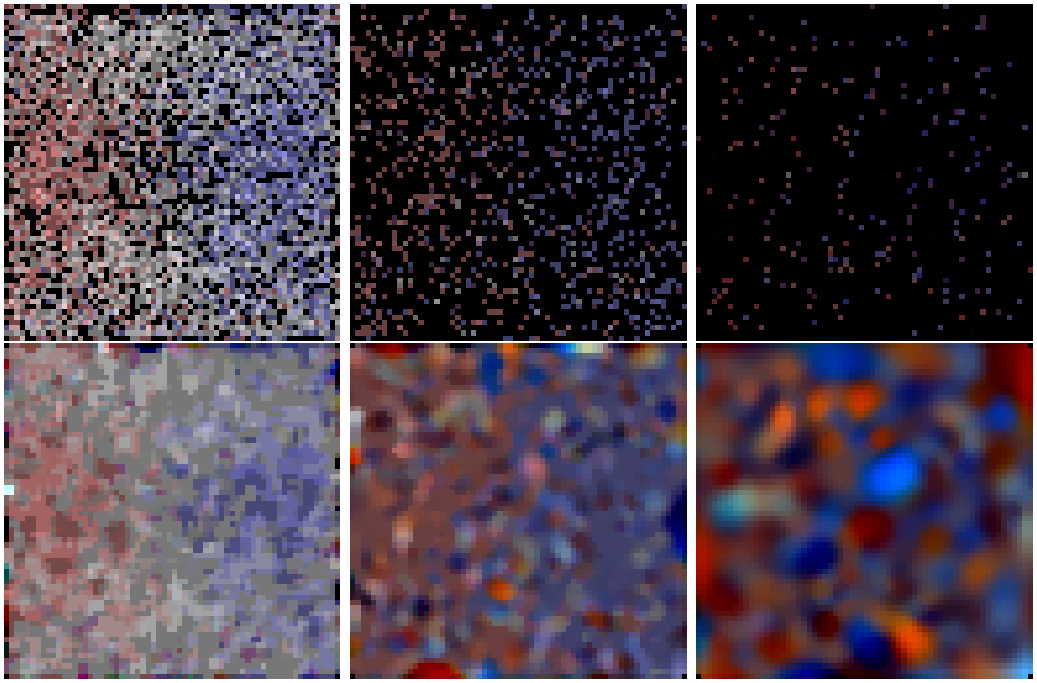


Figure 4.6: Sample image of Environment colour captured by R1 - R3 rays and their interpolated images

the missing values in the obtained environmental intensity images E_t^r, where r corresponds to the number of ray reflections.

In figure 4.6, we see that the more ray reflects, the less bright the pixels become. The main reason behind this phenomenon is because of the ray tracing algorithm and considering that the first ray $r1$ has more influence on the final pixel intensity than the ray $r3$. Therefore, when we have more ray reflections, the intensity of the pixels needs to be reduced, accordingly.

In stage 5, we generate the new input images $A_{t+1} = A_t - E_t^r$ by subtracting

the environmental intensity reducing the inter-reflections from the original input images. There are three different sets of images for each ray reflection $r1$, $r2$ and $r3$.

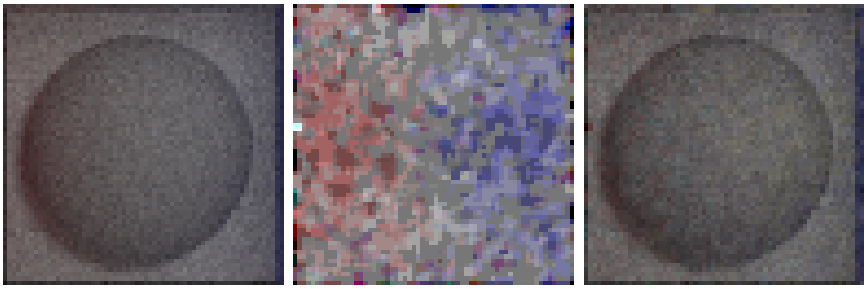


Figure 4.7: (Left) Image with inter-reflections, (Middle) estimated environmental intensity image and (Right) obtained image without inter-reflections.

Finally, the obtained images which have fewer inter-reflections (example difference image is shown in figure 4.7) are used for as input to photometric stereo, generating a new H_{t+1} surface. The whole process can be applied iteratively for a certain number of iterations or until the difference $D_H = H_{t+1} - H_t$ between a new 3D surface and the previous one is less than a given threshold.

Additionally, stage 6 shows how the extracted global illumination information can be baked to the other synthetic model. This stage can be particularly important for improving the quality of real-time generated models such as the data generated by the approach proposed in chapter 3 or other compositing based applications. The inter-reflection map similar to the commonly used shadow map can be generated using the proposed techniques. For example, the inter-reflection map can be generated when the environment information is known such as environment texture map, camera specification (e.g. FOV, position and angle). The spherical dome project method can be used to project the environment texture and the inter-reflection can be captured accordingly. Ideally, the information can be applied in a similar fashion to a shadow map. Nevertheless, the inter-reflection baking is not implemented in the proposed method and should be investigated further.

4.3 Results

In our comparative evaluation study, three different datasets with ground truth were used. Scan data from the Harvard PS dataset (Frankot et al. 1988), a dataset with faces (Argyriou and Petrou 2008) and synthetic data generated by simulated objects.

4.3.1 Experiments and Analysis

We used the photometric stereo approach to reconstruct the sets of the acquired H_t surface, with and without inter-reflections considering different numbers (1 to 3) of ray reflections in the proposed reverted Monte-Carlo ray tracing algorithm. We then estimate the height-, albedo- and normal-error compared to classic PS method (J. Sun et al. 2007) using the available ground truth.

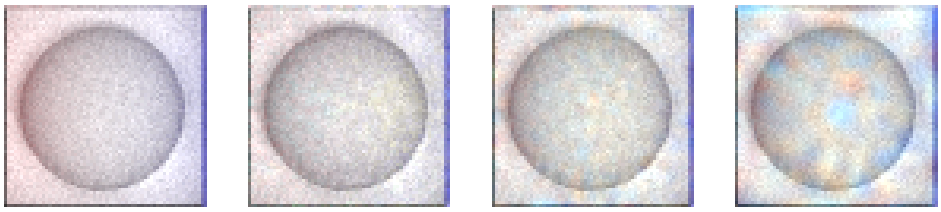
To calculate the height-error we used the equation,

$$\overline{H}_{err} = \frac{1}{n}\left(\sum_{i=1}^{n} |H_{GT} - H_t|_i\right) \tag{4.8}$$

$\overline{H}_{err}$ is the mean for height error. H_{GT} is the height value of the ground truth surface, whereas H_t is the height value of the reconstructed surface. Regarding the albedo-error we use the equation below,

$$\begin{aligned} P^r_{err} &= |P^r_{GT} - P^r_H| \\ P^g_{err} &= |P^g_{GT} - P^g_H| \\ P^b_{err} &= |P^b_{GT} - P^b_H| \\ P^{rgb}_{err} &= \frac{\overline{P^r_{err}} + \overline{P^g_{err}} + \overline{P^b_{err}}}{3} \end{aligned} \tag{4.9}$$

where P^{rgb}_{err} is the albedo-error from mean of individual colour channel; Red P^r_{err}, Green P^g_{err}, and Blue P^b_{err} channel.



(a) Albedo using Classic PS **(b)** Albedo using IRT-PS method 1 **(c)** Albedo using IRT-PS method 2 **(d)** Albedo using IRT-PS method 3

Figure 4.8: Example of the estimated albedo using classic PS (J. Sun et al. 2007), and the proposed IRT-PS method using 1-, 2- and 3-ray reflections.

Likewise, to calculate normal-error we utilised the following equation:

$$\begin{aligned} N^{x}_{err} &= |N^{x}_{GT} - N^{x}_{H}| \\ N^{y}_{err} &= |N^{y}_{GT} - N^{y}_{H}| \\ N^{z}_{err} &= |N^{z}_{GT} - N^{z}_{H}| \\ N^{xyz}_{err} &= \frac{\overline{N^{x}_{err}} + \overline{N^{y}_{err}} + \overline{N^{z}_{err}}}{3} \end{aligned} \tag{4.10}$$

N^{xyz}_{err} denote the mean normal-error for all the axis x, y, and z.Where $\overline{N^{x}_{err}}$ is a mean error for X axis, $\overline{N^{y}_{err}}$ is mean error for Y, and $\overline{N^{z}_{err}}$ is mean error for Z, N^{xyz}_{H} is normal from reconstructed surface.

Table 4.1: Obtained results for the synthetic data, the Harvard and the face PS database comparing the (J. Sun et al. 2007) method, with the 3 variations of the proposed IRT-PS approach.

Synthetic	**PS**	**IRT-PS r1**	**IRT-PS r2**	**IRT-PS r3**
Height	18.653	18.460	18.565	**18.436**
Albedo	0.082	0.082	**0.081**	0.087
Normal	0.825	0.824	0.824	**0.823**
Harvard	**PS**	**IRT-PS r1**	**IRT-PS r2**	**IRT-PS r3**
Height	8.150	8.140	8.097	**7.296**
Albedo	0.522	**0.518**	0.520	0.521
Normal	0.840	0.839	**0.838**	0.840
Face	**PS**	**IRT-PS r1**	**IRT-PS r2**	**IRT-PS r3**
Height	9.341	9.181	9.272	**8.835**
Albedo	0.235	0.231	**0.230**	0.241
Normal	0.823	0.823	0.8221	**0.822**
Overall	**PS**	**IRT-PS r1**	**IRT-PS r2**	**IRT-PS r3**
Height	**12.049**	11.927	11.978	**11.523**
Albedo	**0.280**	**0.2772**	0.2773	0.283
Normal	**0.829**	0.829	**0.8283**	0.8288

Figure 4.9: Overall results for three dataset: Synthetic, face and Harvard dataset. The r1 (Ray1) and r3 (Ray3) produced the best results for albedo and height estimation, respectively.

From the table 4.1, and charts in figure 4.9, we can see that the overall trend of mean Height, Albedo, and Normal errors are reduced with our approach than the classic photometric stereo one. In table 4.1, text highlighted in red are the average overall results of the (J. Sun et al. 2007) photometric stereo method. Whereas best results from our IRT-PS approach are highlighted in the green text. From the figure 4.9, we can see the general trend of the height error: Results improve with each additional ray and the best result is achieved by Ray 3. Likewise, the best results for Albedo and Normal are given by Ray1 and Ray 2 repectively. The indirect illumination captured by our method were able to reduce the inter-reflection effect from the original images. This shows that if we improve the captured indirect illumination then it should result in more accurate and detailed reconstructed surfaces.

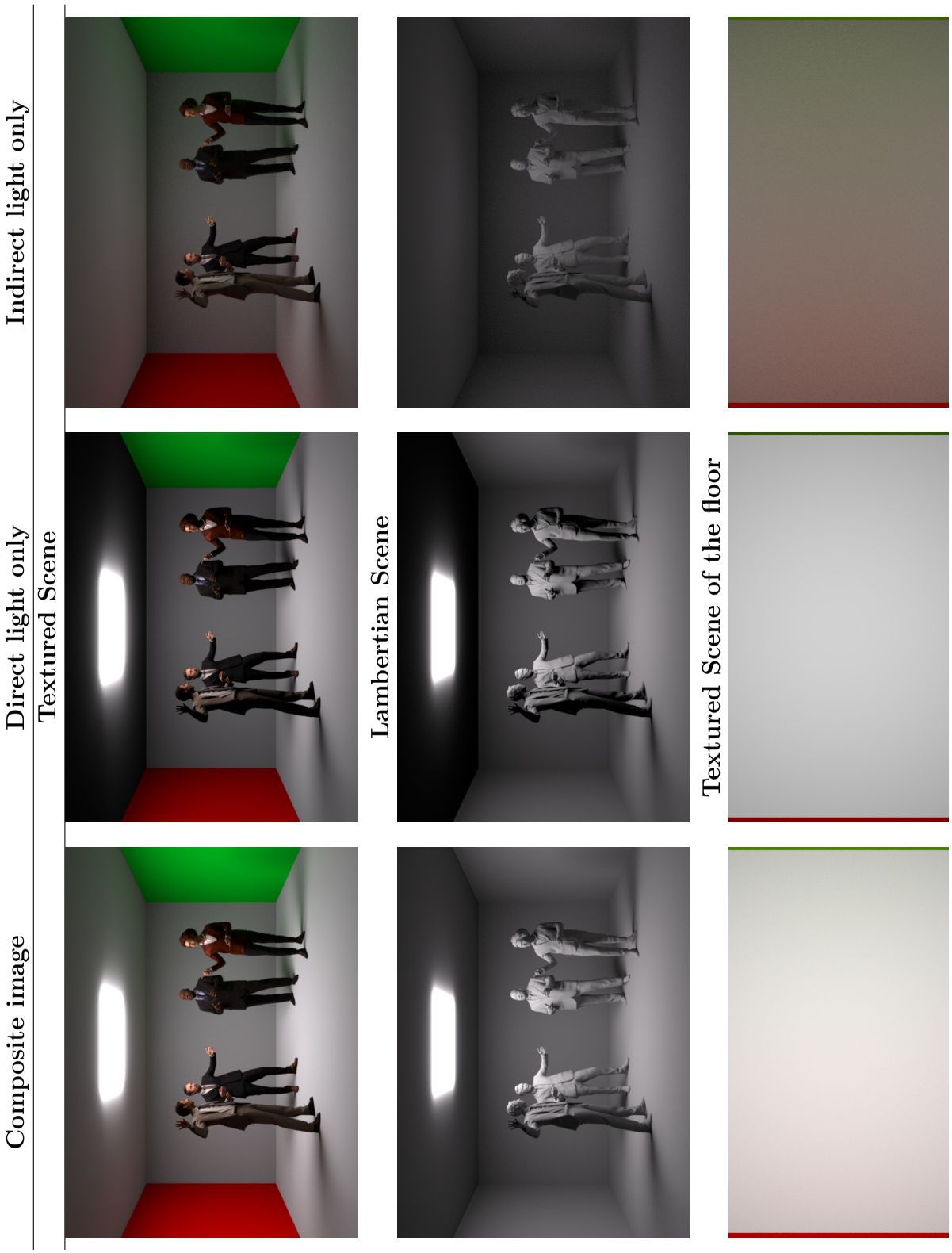


Figure 4.10: Sample images to demonstrate more complex scenes with humans.

4.4 Discussions

The figure 4.10 demonstrates the more complex scenarios. We can see the effect of inter-reflection in the composite image. The effect of inter-reflection is more prominent in the scenes with the texture than the scenes without. As shown in the third row, due to the inter-reflection, the colour of the box bleeds into the floor and red and green colour can be seen. In addition, the scene without texture is also affected by the phenomena. In our experiments, in order to demonstrate the proposed method, we generated images with inter-reflection. The composite image can be split into two parts: direct and indirect scenes. All the images in the first column are composite of direct (second column) and indirect images (third column). Likewise, the scene with direct light is also commonly found in the gaming environment where due to the requirement of real-time rendering, the scene usually does not contain inter-reflection or bake in the texture map itself. The third column in the scene with only inner-reflection and is not usually found. Hence, the aim of the proposed method is to extract this information and reduce it from the final composite images so that these images can be used for 3D reconstruction from image methods (i.e. such as photometric stereo). As mentioned previously, the 3D reconstruction from direct images can generate better 3D mesh then the images with inter-reflection.

In regards to the performance of our approach, depending on the software language choice, the rendering time of the direct and indirect light with complex scenes can be much different. In most cases, rendering direct light is faster than with indirect lights. Likewise, simple experiments with 2 different languages MATLAB and C++ had vastly different results in rendering 64x64 pixels images with a simple Cornell box and 3D sphere. We implemented our approach in MATLAB with multi-core cpu support and it took more than 15 minutes to render the image whereas only 4 seconds in C++. Hence, further work is required for detailed analysis on the performance of our approach.

4.5 Summary

In this work, a novel iterative method considering inter-reflection both due to concavities and the environment was proposed. The IRT-PS approach iteratively applies Photometric Stereo and a reverted Monte-Carlo ray tracing algorithm, reconstructing the observed surface and separating the indirect from direct lighting. A comparative study was performed evaluating the reconstruction accuracy of the proposed solution on three different datasets and the overall results demonstrate improvement over the classic approaches that do not consider environmental inter-reflections. In the next chapter, we examine the difficulties that lie in the crowd analysis field such as perspective distortion, and we propose a novel deep learning architecture which applies a number of approaches to overcome such problems.

In the next chapter, we examine the problems in the density map generation method commonly used in crowd counting. At present, the generated map is not content aware i.e., it does not know the relative size of humans' heads. Hence, we proposed a technique which takes in consideration of the content of images and generates a density map accordingly.

"There's no one particular road that will lead you to success. I think everybody will find it differently."

— Marina and the Diamonds

5

A content-aware density map for better Crowd estimation and counting

Contents

In the earlier chapter 4, we proposed a method which improved the 3D data generation process and enhanced the quality of the generated data. Similarly, in this chapter we examine the problem that exists in the density map generation process. As introduced in chapter 2, the core problem of density map is that it is not content-aware. Hence, in this chapter, we proposed a method to generate a content-aware and higher quality density map.

Precise knowledge about the crowd size, proximity and density can provide valuable information for various tasks such as crowd safety and security, event planning and analysing consumer behaviour. Creating a powerful machine learning model, capable to perform such complex task demands for a large and

highly accurate and reliable ground truth data. Unfortunately the existing crowd counting and density estimation benchmark datasets are not only limited in terms of the quantity, but also lack in terms of annotation strategy. This study attempts to address this issue through a content aware technique, using combinations of Chan-vese segmentation algorithm, two-dimensional Gaussian filter and brute-force the nearest neighbour search. The results show by simply replacing the commonly practised density map generators with the proposed method, higher levels of accuracy can be achieved using the existing state-of-the-art models.

5.1 Motivation

The study of human behaviours is a subject of great scientific interest and probably an inexhaustible source of research. One of the most cited and popular research topics in human behaviour analysis is study of crowd features and characteristics. In recent years, crowd analysis has gained a lot of interest mainly due to its wide range of applications such as safety monitoring, disaster management, public spaces design, and intelligence gathering, especially in the congested scenes like arenas, shopping malls, and airports (Pelechano et al. 2005; Silverman et al. 2005).

Crowd counting, localisation and density estimation are crucial objectives of automated crowd analysis systems. Accurate knowledge of the crowd size, location and density in a public space can provide valuable insight for tasks such as city planning, analysing consumer shopping patterns as well as maintaining general crowd safety. Several studies attempt to produce an accurate estimation of the true number of people present in a crowded scene through density estimation.

Classic computer vision and machine learning techniques were struggling with overwhelming complexity of crowd counting and behaviour analysis models. However, the emergence of deep learning in the last decade was a breath of

fresh air for crowd behaviour, counting and simulation studies and has largely advanced the state of the art in this domain (Marsden et al. 2016).

Generally, crowd counting and density estimation approaches can be divided into two categories: detection-based methods (atomistic) and regression-based methods (holistic). Detection-based methods generally assume each person in the crowd can be detected and located individually based on its individual features and characteristics. These approaches are preferable in sparse crowd analysis where crowd occlusion is negligible. Holistic crowd counting and behaviour analysis approaches utilise global crowd features and characteristics to estimate the crowd size, location and density. These approaches are preferable in dense crowd analysis where crowd occlusion is significant. Due to the high amount of occlusions these approaches only utilise heads as deterministic feature (Ryan et al. 2015).

Despite this, crowd counting and density estimation is not a trivial task. Several key challenges such as severe occlusions, poor illumination, camera perspective and highly dynamic environments further complicate this task. On top of these, lack of quality annotated training data further challenges the crowd counting and behaviour analysis studies performance. The existing crowd counting and density estimation benchmark datasets are not only limited in terms of the quantity, but also lack in terms of annotation strategy.

In regression-based crowd counting and density estimation approaches, heads are the only confidently visible body part in the image. Thus, these approaches use heads as the only discriminant feature. Meanwhile, the existing benchmark datasets such as UCF-CC-50 and ShanghaiTech are only providing the heads centroid pixel instead of masking the entire head region. Hence, the recreation of the ground truth head masks is accomplished through a static two-dimensional Gaussian filter or a dynamic two-dimensional Gaussian based on the K nearest neighbours. Despite, a dynamic Gaussian approach based on proximity of the nearest neighbours mitigates the issue to some extent,

but this technique is not content aware and incorporates significant amounts of false information into ground truth data (Idrees, Tayyab, et al. 2018; Yingying Zhang et al. 2016).

In this regard, this study attempts to address the limitation of the existing crowd counting and density estimation benchmark datasets through a content aware annotation technique, employing combinations of nearest neighbour algorithm and unsupervised segmentation to generate the ground truth head masks. The proposed technique first uses the brute-force nearest neighbour search to localise the nearest neighbour head point, then it identifies the head boundaries using Chan-vese segmentation algorithm and generates a two-dimensional Gaussian filter on that basis.

We believe that by simply replacing the kNN/Gaussian based ground truth density maps in an existing state-of-the-art network with the proposed content aware approach in this study, higher level of accuracy can be achieved.

5.2 Background

Over the last decade there have been several studies attempting to address crowd counting and density estimation through deep learning techniques. L. Liu et al. (2020) proposed a universal network for counting crowds with varying densities and scales. The proposed deep network in this study is composed of two components, i.e. a detection network (DNet) and an encoder-decoder estimation network (ENet). The input first runs through DNet to detect and count individuals who can be segmented clearly. Then, ENet is used to estimate the density maps of the remaining areas, where the numbers of individuals cannot be detected. Modified version of Xception used as an encoder for feature extraction and a combination of dilated convolution and transposed convolution used as decoder. Authors attempted to address the variations in crowd density with two literally isolated deep networks which significantly slows down the process.

In Valloli et al. (2019) proposed independent decoding reinforcement branch as a binary classifier which helps the network converge much earlier and also enables the network to estimate density maps with high Structural Similarity Index (SSIM). A joint loss strategy, i.e., Binary cross entropy (BCE) Loss and Mean squared error (MSE) Loss used to train the network in an end to end fashion. They have used variations of U-net models to generate the density maps. The proposed model shows notable improvements in recreation of the crowd density maps over the existing models.

A study by Oh et al. (2020) examined the uncertainty estimation in the domain of crowd counting. This study proposed a scalable neural network framework with quantification of decomposed uncertainty using a bootstrap ensemble. The proposed method incorporates both epistemic uncertainty and aleatoric uncertainty in a neural network for crowd counting. The proposed uncertainty quantification method provides additional auxiliary insight to the crowd counting model. The proposed technique attempts to address the uncertainty issue in crowd counting. However, the use of an unsupervised calibration method to re-calibrate the predictions of the pre-trained network is questionable.

Olmschenk et al. (2019) investigated the inefficiency of the existing crowd density map labelling scheme for training deep neural networks. This study proposes a labelling scheme based on inverse k-nearest neighbour (ikNN) maps which does not explicitly represent the crowd density. Authors claim a single ikNN map provides information similar to the commonly practiced accumulation of many density maps with different Gaussian spreads.

A study by Idrees, Tayyab, et al. (2018) stems from the observation that crowd counting, density map estimation and localization are very interrelated and can be decomposed with respect to each other through composition loss which can then be used to train a neural network.

Several other research including (Jiang et al. 2020; Varior et al. 2019; X. Liu et al. 2018; Change Loy et al. 2013; Hossain et al. 2019; Q. Wang et al. 2019b; Ze Wang et al. 2018; D. Kang and A. Chan 2018; Lingbo Liu, Hongjun Wang, et al. 2018; C. E. Kim et al. 2018; V. Ranjan et al. 2018; Z. Shi et al. 2018; Babu Sam et al. 2018) tried to address crowd counting, localization and density estimation issues yet the majority of these approaches employed the flawed ground truth density map generation approach.

5.3 Implementation

In dense crowd scenarios, aside from the heads which are usually fairly visible, the majority of the other body parts are subject to heavy occlusion. This makes heads the only reliable discriminant feature in dense crowd counting and localization. Existing crowd counting and density estimation benchmark datasets such as UCF-CC-50 (Idrees, Saleemi, et al. 2013) and ShanghaiTech (Yingying Zhang et al. 2016) are providing the heads centroid pixel location as labels. Conducting the crowd counting and density estimation as a regression task, seeks for regional isolation of the heads in the form of a binary mask. As the head size is subject to various factors such as camera specifications, point of view, perspective, distance and angle, generation of such a mask could be a challenging task, given the heads centroid pixel is the only provided form of annotation in existing benchmark datasets.

The formation of the ground truth binary head masks in large part of the existing studies is either accomplished through a static two-dimensional Gaussian filter or a dynamic two-dimensional Gaussian filter paired with k nearest neighbours approach. The static two-dimensional Gaussian filter assigns a fixed size Gaussian filter to each head regardless of the head size and proximity of the nearest neighbour. This approach does not attempt to compensate for crowd density, distance and camera perspective and incorporates significant amounts of noise into ground truth data. The dynamic

two-dimensional Gaussian filter approach employs the nearest neighbours search through the k-d tree space partitioning approach, prioritises the speed over integrity and does not deliver optimal results. In this approach the Gaussian filters are centred to the annotation points and spread based on the average euclidean distance among the three nearest neighbours. In both approaches, the spatial accumulation of all Gaussians creates the global density map for the given image. The following formula shows the commonly used dynamic two-dimensional Gaussian approach:

$$D(x, f) = \sum_{h=1}^{T} \frac{1}{\sqrt{2\pi f(\sigma_h)}} exp(-\frac{(x - x_h)^2 + (y - y_h)^2}{2f(\sigma_h)^2}) \tag{5.1}$$

Where T is the total number of the heads presents in the given image, σ_h is the sized for each head point positioned at (x_h, y_h) determined by k-d tree space partitioning approach based on the average euclidean distance among the three nearest neighbours and f is a scaling constant.

Despite, dynamic Gaussian approach based on proximity of the k nearest neighbours attempts to mitigate the crowd density, distance and camera perspective issues to some extent, but this technique is not content aware and it injects significant amounts of false information into the ground truth data which negatively affects the model's accuracy. Figure 5.1 shows some sample images from ShanghaiTech dataset (Yingying Zhang et al. 2016) along with their respective density maps. It can be observed that both approaches are fairly unreliable and inconsistent in determining the true head sizes.

5.3.1 Methodology

In order to address the shortcomings of the existing ground truth density maps generation approaches, this study offers a content aware technique using combinations of Chan-vese segmentation algorithm, two-dimensional Gaussian filter and brute-force nearest neighbour search.

Figure 5.1: From top to bottom: sample images from ShanghaiTech dataset (Yingying Zhang et al. 2016), density map based on static two-dimensional Gaussian filter and density map based on dynamic two-dimensional Gaussian filter using k-d tree space partitioning technique.

This technique is based on the Mumford-Shah functional for segmentation, and is widely used in the medical imaging field. The Chan-vese segmentation algorithm is able to segment objects without prominently defined boundaries. This algorithm is based on level sets that are evolved iteratively to minimise an energy, which is defined by weighted values corresponding to the sum of differences intensity from the average value outside the segmented region, the sum of differences from the average value inside the segmented region, and a term which is dependent on the length of the boundary of the segmented region. As the head boundaries in highly dense crowds are not clearly defined, this technique can be used to segment the head regions from the background. Chan-vese algorithms attempt to minimise the following energy function in

an iterative process (T. F. Chan et al. 2001).

$$\begin{aligned} F(c_1, c_2, G) = \mu.Len(G) + \nu.Area(in(G)) \\ +\lambda_1 \int_{in(G)} |u_0(x,y) - c_1|^2 dxdy \\ +\lambda_2 \int_{out(G)} |u_0(x,y) - c_2|^2 dxdy \end{aligned} \tag{5.2}$$

where G denote the initial head which manually set to a box of [5x5] pixels centered to the annotation head point, c_1 will denote the average pixels' intensity inside the initial head region G, and c_2 denotes the average intensity of a square box, centered to the annotation head point and its boundary extended to the nearest neighbour head point. λ_1 , λ_2 and μ are positive scalars, manually set to 1, 1 and 0 respectively. A two-dimensional Gaussian filter which extends to the G mean and centered to the head point is used to create the ground truth head mask.

Unlike the k-d tree space partitioning technique which does not always deliver the absolute nearest neighbours, brute-force nearest neighbour search technique always guarantees to find the absolute nearest neighbours regardless of the distribution of the points. The brute-force nearest neighbour search technique does take considerably longer time ($O(n^2)$ vs $O(n \log n)$) to find the nearest neighbours. However, since generating the ground truth density maps is a single-pass preliminary operation in crowd counting and density estimation, speed is a less of a priority. Since, the Chan-vese segmentation algorithm only uses the very nearest neighbour head point to determine the boundary of the outside region, the brute-force nearest neighbour search only looks for the very nearest head point. To create the global density map, we employed an exclusive cumulative of the Gaussians which address the head mask overlap issue. To maintain the count integrity, the density map has been normalized at each iteration.

5.4 Experiments and Analysis

In order to measure the effectiveness of our content-aware crowd density map generator, we have re-trained some notable state of the art deep models including Vishwanath A Sindagi et al. (2017a)[1], Z. Shi et al. (2018)[2], Y. Li et al. (2018)[3] and C. Zhang, Hongsheng Li, et al. (2015)[4] using the density maps generated by the proposed crowd density map generator. We used the original implementation of these algorithms provided by authors in GitHub.

All algorithms were trained and tested across both UCF-CC-50, ShanghaiTech (SHT) and synthetic datasets using the proposed content-aware crowd density map generator as well as the commonly used existing ground truth density map generator. In some cases we were unable to reproduce the reported performance in the original manuscripts. There might be a multiple reasons for not being able to get the same results that the original authors achieved. Few key points that should be noted are the total epoch i.e duration of training, additional augmentation methods and much more. However, as we were consistent with the experiments across both density map generators, validity and integrity of the comparison is not compromised.

Table 5.1 shows the Mean Square Error (MSE) comparison between the proposed and existing density map generator across ShanghaiTech dataset Part-A and B. It can be observed that using the proposed content-aware density map generator, MSE has been consistently decreased across relatively all investigated models. The improvement is more pronounced in the ShanghaiTech (SHT) Part-A dataset. ShanghaiTech Part-A dataset exhibits more challenging and dynamic crowd scenarios. The results convey the proposed method could deliver better depiction of the ground truth density maps. Table 5.2 compares the Mean Square Error (MSE) and Mean Absolute Error (MAE)

[1]https://github.com/svishwa/crowdcount-cascaded-mtl
[2]https://github.com/shizenglin/Deep-NCL
[3]https://github.com/leeyeehoo/CSRNet-pytorch
[4]https://github.com/wk910930/crowd_density_segmentation

Figure 5.2: From top to bottom: sample images from ShanghaiTech dataset, density map generated using the existing method and density map generated using the proposed method.

between the proposed and existing density map generator using an extremely challenging UCF-CC-50 dataset. Similar to the results in ShanghaiTech dataset, there is a notable improvement in both MSE and MAE metrics.

Figure 5.2 compares the density maps generated using the existing approach based on k-d tree space partitioning technique and the proposed content-aware crowd density map generator. It can be observed that in highly dense crowds, the proposed method generates more granular density maps with lesser overlaps between neighbour Guassians. The proposed method uses a combination of pixel intensity and nearest neighbours to adjust the size of the Guassians per head. Figure 5.2 shows this technique significantly improves the integrity of the density map relative to the input image.

Table 5.1: MSE comparison between the existing and proposed density map generator across ShanghaiTech (Part-A and B)dataset (* lower value is better).

Method	Existing Method (MSE)		Proposed Method (MSE)	
	Part-A	Part-B	Part-A	Part-B
Cascaded CNN (Vishwanath A Sindagi et al. 2017a)	152	31	149↓	28↓
D-ConvNet (Z. Shi et al. 2018)	112	26	110↓	26↓
CRSNet (Y. Li et al. 2018)	115	16	113↓	16
Crowd CNN (C. Zhang, Hongsheng Li, et al. 2015)	197	66	191↓	57↓

Table 5.2: MSE and MAE comparison between the existing and proposed density map generator across UCF-CC-50 dataset (* lower value is better).

Method	Existing Method		Proposed Method	
	MSE	MAE	MSE	MAE
Cascaded CNN (Vishwanath A Sindagi et al. 2017a)	397	322	397	320↓
D-ConvNet (Z. Shi et al. 2018)	415	293	414↓	286↓
CRSNet (Y. Li et al. 2018)	397	266	396↓	264↓
Crowd CNN (C. Zhang, Hongsheng Li, et al. 2015)	498	467	483↓	459↓

5.5 Summary

Creating an accurate model for crowd counting and density estimation demands for a large and highly reliable ground truth data in the first place. However, the existing crowd counting and density estimation benchmark datasets are not only limited in terms of size, but also lack in terms of annotation methodology. This study attempted to address this issue through a content-aware technique which employed combinations of Chan-Vese segmentation algorithm, two-dimensional Gaussian filter and brute-force nearest neighbour search to generate the ground truth density maps. Experiment results show that by replacing the commonly practised ground truth density map generators with the proposed content-aware method, the existing state-of-the-art crowd counting models can achieve higher level of count and localisation accuracy.

In the next chapter, we examine the difficulties in the crowd analysis field such as perspective distortion, and we propose a novel Deep Learning architecture which applies a number of techniques to overcome such problems.

"Create. Not for the money. Not for the fame. Not for the recognition. But for the pure joy of creating something and sharing it."

— Ernest Barbaric

6

Tackling one problem at a time with composite techniques using Deep Learning for crowd analysis

Contents

In chapter 5, we proposed a method which improved the quality of the density map. In addition, it demonstrated that generated density maps elevated the performance of existing Deep Learning networks.

In this chapter, we propose a novel deep network, called *Adaptive Scale Variance Network (ASVNet)*, for accurate and effective crowd counting. Designing a general purpose crowd counting network applicable to a wide range of crowd images is challenging, mainly due to the large variability in camera perspective. To address this, ASVNet extracts multi-scale features with a pyramid contextual module to provide long-range contextual information and enlarge the receptive field. A scale aggregation module, which extracts the multi-scale features and a multi-branch self-attention module, are proposed to cater for perspective variations. Most existing methods utilise the Euclidean loss function and assume that pixels are independent and overlook the local correlation in the derived density maps. We propose a novel method called *switch loss function*, which combines two sets of loss functions. Alongside the local pattern consistency loss (SSIM), we also propose Peak Signal-to-Noise Ratio (PSNR) as a novel loss function, to improve the performance of our model. We conducted comprehensive experiments on three major crowd counting datasets to test our proposed method and ASVNet produced a better performance than state-of-the-art methods.

6.1 Introduction

Crowd analysis has been subject of intense research because of its wide range of applications such as public safety management (e.g. rallies, sporting events), congestion avoidance (e.g. traffic control), and flow analysis (V. Sindagi et al. 2017a; D. Kang, Ma, et al. 2018). In this chapter, we examine the complexity

Images removed for copyright reasons

Figure 6.1: Sample images and related density maps from crowd counting datasets. The images present various challenges in crowd estimation such as severe occlusions, perspective distortion, and highly variable crowd density.

of crowd estimation in arbitrary images. When no prior information of a captured scene is available, such as camera specifications and the scene layout, deriving an accurate estimate of the crowd density and people count is a hard task. Given the complexity of people counting in a potentially crowded scene and the lack of ground truth, research has shifted to crowd density estimation, visualising the results with a density map (see figure 6.1). In an accurate density map, each pixel value corresponds to the crowd density at the corresponding location in the scene (C. Zhang, K. Kang, et al. 2016). However, generating accurate density maps is challenging due to the complexity of crowd scenes caused by various factors including occlusions, clutter, potentially large variations in perspective and highly variable crowd distribution.

Recent approaches adopt Deep Neural Networks (DNN) to estimate crowd density and the related generation of a high quality crowd density map. While DNN based methods (Yingying Zhang et al. 2016; Vishwanath A Sindagi et al. 2017b; Lingbo Liu, Hongjun Wang, et al. 2018; Cao et al. 2018; Y. Li et al.

2018) have significantly improved over time, there still exists the problem of accuracy degradation when applied to scenes where congestion and perspective affect greatly visibility of people and their size in the image. Our goal is to generate high-resolution density maps that help to capture accurately the crowd density of a captured scene.

Recent methods (Yingying Zhang et al. 2016; Sam, Surya, et al. 2017) have shown higher accuracy by using multi-scale architectures, to cater for the perspective problem. To tackle large variation in people's head size, several filter sizes are used to extract features. Most published research employs multi-column DNN (Yingying Zhang et al. 2016) or stacked multi-branch blocks (Cao et al. 2018) to extract features with different receptive fields, combining them either using a concatenation or using an average weighting method to generate the corresponding density map. Nevertheless, two main drawbacks still exist in approaches based on DNN. On one side, crowd estimation benefits from the design of a multi-scale representation of the multi-column architecture. However, the diversity in scale is entirely constrained by the number N of columns present in the architecture. For example, in (Yingying Zhang et al. 2016)only three multi-columns are used.

On the other hand, a Euclidean loss function is widely used for crowd estimation, which assumes that pixels have independent information. Furthermore, this loss function is known to produce blurred images (Isola et al. 2017). An adversarial loss (Goodfellow et al. 2014) has been applied in (Vishwanath A Sindagi et al. 2017b) to further improve the density map quality, achieving better performance. Nonetheless, the additional discriminator still remains computationally costly.

In order to address these issues, we focus on two points. First, features at different scales contain different information, while the shallower layers features present low-level appearance details. The work in (D. Xu et al. 2017; L. Zhang, Ju Dai, et al. 2018) has demonstrated that refining these

complementary features can help the network to be robust to scale variation. Nevertheless, simple strategies such as weight average and concatenation are generally adopted in most existing methods to combine multiple features, which cannot capture the scale variance information. Therefore, it is necessary to propose a more effective mechanism for the task of crowd counting, to fully exploit the information between features at different scales and improve their robustness. Second, the rich information of scale variance present in a crowd density map can be captured effectively by the loss function.

The architecture of the proposed network ASVNet, is shown in figure 6.2. Motivated by feature pyramid network architectures (J.-P. Lin et al. 2018) in image recognition domain, we employ the pyramid encoder by adaptively enlarging its receptive field and scale aggregation module, to improve representation ability and scale diversity of features. Using adaptive scale and self-attention modules with multi-branch architecture, high scale sensitivity and much better accuracy can be attained in detecting sparse and dense crowds. Finally, it generates high-resolution and high-quality density maps of which the resolution is exactly the size of input images. We propose a novel loss which uses a combination of peak signal-to-noise loss and local pattern consistency loss, to exploit the local correlation as well as the quality of the density map. The local pattern consistency loss is computed using the SSIM (Zhou Wang et al. 2004) to measure the structural similarity between the estimated density map and the corresponding ground truth. While extra computation is required for the loss, the computation cost is negligible, and the results show its implementation improves the performance.

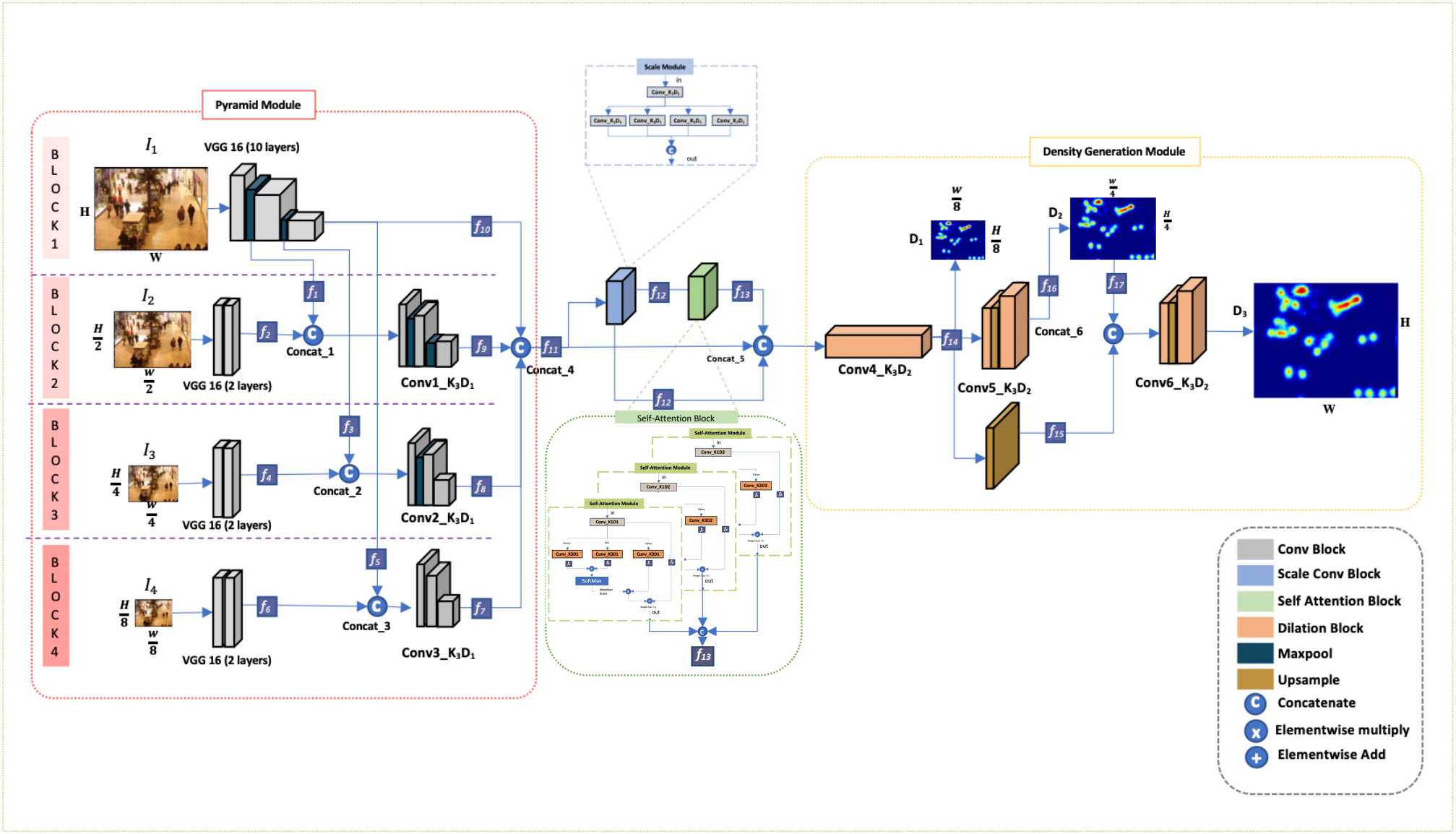


Figure 6.2: Overview of the proposed ASVNet architecture consisting of four major modules: pyramid, scale aggregation, self attention and density generation.

6.2 ASVnet for better crowd estimation

In this section, we present the details of the proposed Adaptive Scale Variance Network (ASVNet). We first describe how the ground truth density map is generated and then introduce the ASVNet architecture, followed by the proposed loss function.

6.2.1 Ground truth generation

We introduce the density map generation process under section 2.5.4. Here, we will briefly go through the creation of ground truth i.e. density maps. We generate our density map based on a geometry-adaptive Gaussian kernel proposed by (Wei et al. 2016).

Geometry-adaptive kernels are used to address highly congested scenes. The density map can be generated by blurring each head annotation using a normalised Gaussian kernel as in (Yingying Zhang et al. 2016; Sam, Surya, et al. 2017; Vishwanath A Sindagi et al. 2017b). The geometry-adaptive kernel is defined as:

$$D(x) = \sum_{i=1}^{N} \delta(x - x_i) \times G_\sigma(x), \ with\ \sigma = B\overline{d}_i \tag{6.1}$$

where, x_i is the target object for the ground truth δ. $\overline{d}_i$ is the average distance of k nearest neighbours. To generate the density map, we convolve $\delta(x - x_i)$ with a Gaussian kernel with parameter σ_i (standard deviation), where x is the position of the pixel in the input image. In our experiments, we configured the parameters according to (Wei et al. 2016), where $\beta = 0.3$, and $k = 3$. For a sparse crowd, we empirically set the Gaussian kernel based on the average head size of people in the input images.

6.2.2 Architecture of ASVNet

ASVNet consists of three major components: the pyramid contextual module, the adaptive scale aggregation module and the self-attention module. As shown in figure 6.2, we designed our ASVNet to cater for the scale variance problem by implementing a multi-scale feature extractor and generating a high-resolution density map to capture finer details. Existing approaches utilise multi-column architectures with various filter sizes to address the multi-scale issues (Yingying Zhang et al. 2016). The multi-scale features are extracted from shallower layers early on and then fed to various scale feature branches. Finally, the features are merged to predict the density map. We implemented a similar architecture with additional modules namely: the pyramid, the scale aggregation and the self-attention module.

Given an image of $H \times W$ size, we first build a four level image pyramid I_1, I_2, I_3, I_4, where $I_1 \in R^{H \times W}$ is the original image, $I_2 \in R^{\frac{H}{2} \times \frac{W}{2}}$, $I_3 \in R^{\frac{H}{4} \times \frac{W}{4}}$, and $I_4 \in R^{\frac{H}{8} \times \frac{W}{8}}$ are the downsampled ones. Each of these images is fed into one of the sub-network blocks. The extracted features of the Conv$N$ layers are denoted as $\{f_n\}$, where N is a scalar value. The features from various levels of block-1 are grouped with different sub-networks with the same resolution. Three sets of multi-scale features are extracted from the process $\{f_1, f_2\}, \{f_3, f_4\}, \{f_5, f_6\}$. Within each block, these features are fed into a convolutional layer and, depending on the resolution of the feature maps, features are further max-pooled. Conv$N$_$K_pD_q \rightarrow \{F, M, U\}$ is a convolutional layer where, N is the convolutional layer's number, K indicates the convolutional kernel, p is the kernel size, D is the dilation and q is the dilation rate. F is the total number of features, M represents max-pool and U is the upsampling. By default, all the convolutional layers use kernel size 3 denoted by K_3. In Block 2, features $\{f_1, f_2\}$ are fed into Conv1_$K_3D_1 \rightarrow \{64, 64, M, 128, 128, M, 256, 256\}$ that returns features $\{f_9\}$. In Block 3, $\{f_3, f_4\}$ is passed to Conv2_$K_3D_1 \rightarrow \{64, 64, M, 128, 128, 256, 256\}$, which

returns $\{f_8\}$. Likewise, $\{f_5, f_6\}$ to Conv3_K_3D_1, which generates features $\{f_7\} \rightarrow \{64, 64, 128, 128, 256, 256\}$. The different features present in each set complements each other, since the features are deduced from various receptive fields and are obtained from separate convolutional layers of multi-scaled images. For example, features $\{f_4\}$ primarily consist of appearance information, whereas $\{f_{10}\}$ represent some high-level semantic information.

To enhance the robustness of scale invariance, we improve the features in the succeeding three sets $\{f_7, f_8, f_9\}$ and features from block-1 $\{f_{10}\}$ are concatenated and fed to the scale aggregation module as $\{f_{11}\}$. The features extracted from scale aggregation module $\{f_{12}\}$ are then passed to self-attention module described in section 6.2.2.2. With richer scale information, the features $\{f_{13}\}$ become more robust to scale variation. The features $\{f_{13}\}$ are concatenated with $\{f_{11}\}$ and fed into Conv4_$K_3D_2 \rightarrow$ $\{256, 256, 256, 256, 256, 128, 128, 128, 128, 64, 64, 64, 32, 32,$
$32\}$ for deeper feature representation learning. The Conv4_K_3D_2 is a deep convolution layer with kernel size 3 and 2×2 dilation rate. After Conv4, the network produces a high-quality density map. Here, in order to generate high resolution map, a pyramid style architecture is utilised again. An 1×1 filter convolution layer after last features $\{f_{14}\}$ is applied, reducing its channel from $\{32 \rightarrow 1\}$ to generate a density map D_1. However, it produces a low-resolution density map D_1 of size $\frac{H}{8} \times \frac{W}{8}$, because of various max pooling layers throughout the network. The density map D_1 lacks spatial detail. Therefore, we generate two additional density maps D_2, D_3 at shallower layers, where D_n has resolution $\frac{H}{4} \times \frac{W}{4}$ and $H \times W$ respectively. Specifically, D_3 is computed by feeding the concatenation of upsampled features $\{f_{15}\}$ and $\{f_{17}\}$ into Conv6_$K_3D_3 \rightarrow \{32, U, 64, 128\}$, where $\{f_{17}\}$ is final density map D_2, followed by an 1×1 filter convolution layer to reduce features from $\{128 \rightarrow 1\}$. D_2 is obtained in similar manner, where features $\{f_{14}\}$ are fed to Conv5_$K_3D_2 \rightarrow \{32, U, 64, 128\}$, followed by an 1×1 filter convolution layer. U denotes upsampling of the features with an upsampling rate of 2×2. The

final crowd density map $D_3 \in R^{H \times W}$ has fine details of the crowd spatial distribution.

Finally, we train our ASVnet with the combined loss function. We call it *switch loss function* due to its nature of alternating between two sets of loss functions. We implemented PSNR and SSIM alongside conditional Root Mean Square Error (RMSE) loss function and the count loss function.

6.2.2.1 Pyramid module

The pyramid module aims to extract long-range contextual information and enlarge the receptive field. Following a similar concept to (Boominathan et al. 2016; Sam, Surya, et al. 2017; Vishwanath A Sindagi et al. 2017b), our approach incorporates the VGG-16 network (Simonyan et al. 2015) as the front-end of ASVNet, because of its transferability and flexible architecture. As shown in figure 6.2, the pyramid module consists of four blocks. Each branch is fed with multi-scaled versions of the original input image to extract the features. The block-1 consists of truncated VGG-16 layers. Here, the first ten layers of VGG-16 are used to have a larger valid receptive field, as well as to reduce the loss of spatial information. Blocks 2 to 4 utilise the first two layers of the VGG-16.

6.2.2.2 Scale-Aggregation module

The design purpose of both modules is to handle the scale variance in the scene. The scale aggregation module tackles by multi-kernel size, whereas the self-attention module utilises a multi dilation rate to extract the features. Both modules are adaptable and can be extended to arbitrary branches. In figure 6.2 the features f_{11} are fed to the adaptive scale aggregation module. Figure 6.3 shows the architecture of the scale aggregation module.

The scale aggregation module starts with a convolutional layer ConvK_1D_1 before the other branches to reduce the feature dimensions by half. Then we

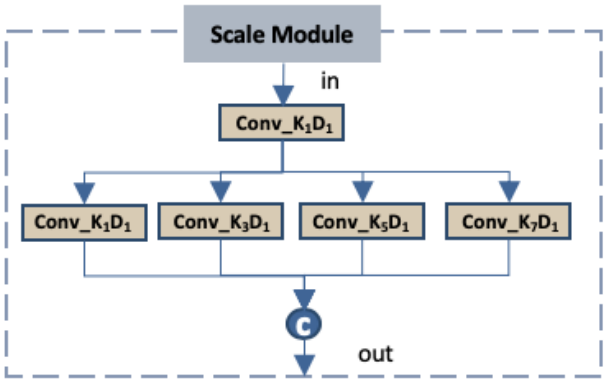


Figure 6.3: The scale Aggregation Modules consist of five branches with different kernel sizes.

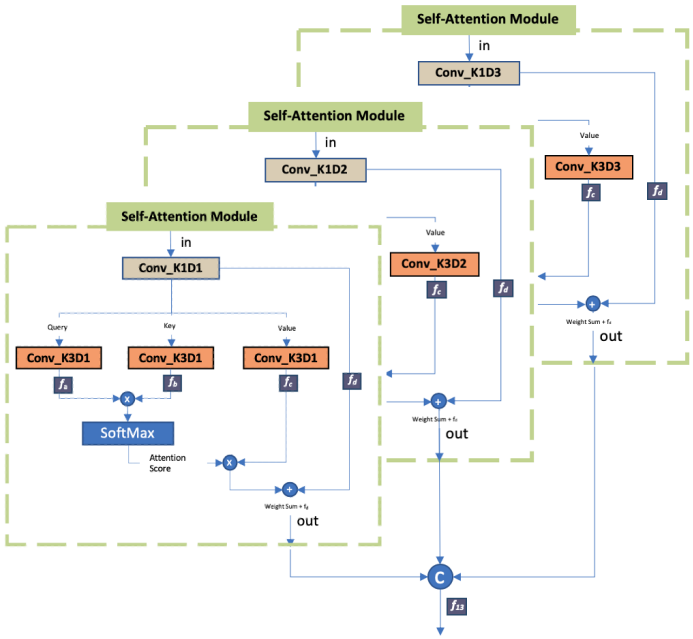


Figure 6.4: Self Attention Module consist of three dilation convolution branch

construct four branches with filter sizes K_1, K_3, K_5 and K_7. The ConvK_1D_1 branch is used to preserve the previous layer scale features and includes small targets, while other branches are used to increase the corresponding field size. Lastly, all the features are concatenated and passed to the next layers.

6.2.2.3 Self-Attention module

The self-attention block consists of three self-attention modules with similar architecture. A self-attention consists of three branches the same filter size $K_3 \rightarrow \{170\}$ with and in dilation ratio $(1 \times 1, 2 \times 2, 3 \times 3)$ (Yu et al. 2016). An additional convolutional layer with filter size K_1 is also added at the beginning of each branch to reduce the number of parameters. This can help to reduce memory requirements without sacrificing performance (Szegedy, Ioffe, et al. 2017). Given the set of feature maps from the previous layer, the attention module forwards them through a set of convolutional layers and a softmax function to generate an attention map with three channels. Each channel represents the importance of the relative features. The attention maps are calculated as follows: first we extract the three set of features using same dilation rate f_a, f_b, f_c, as shown in the figure 6.4. Here, we assume that f_a is query, f_b as key and then calculate attention score by element-wise multiplication, $f_a \times {f_b}^T$ then apply softmax to the score. The attention score is further multiplied with f_c to get the weighted value and finally sum the weighted values and add it with the feature f_d. The same method is applied to other two module expect difference in dilation rate. The feature extracted from each module is then concatenated and feed to following convolutions layers.

6.2.3 Switch Loss function

The standard way of calculating the crowd count is by summing up all the contributions from the density map, as follows

$$D_c = \sum_{i=0}^{n} D_i \tag{6.2}$$

where i is a pixel location and N is a total number of pixels in D density map, which represents the total number of people in the crowd. We propose a novel loss S_L function, which combines two sets of error estimation. During

constructing the function, we focus on two aspects of the image quality loss. The first set of loss focuses on pixel wise loss, which is calculated by $set_a\{RMSE, Count\ loss\}$, where $RMSE$ is the root mean squared error. The second set of loss function focuses on the quality of image $set_b\{PSNR, SSIM\}$, where $PSNR$ and $SSIM$ is the structural similarity loss function. Here, as far as we are aware, we are the first to propose PSNR as a loss function for crowd estimation.

The Set_a loss function switch to Set_b is based on the following condition:

$$S_L \begin{cases} Set_b, & \text{if } D_c - t < \hat{D}_c < D_c + t \\ Set_a, & \text{otherwise.} \end{cases} \tag{6.3}$$

where, $\hat{D}_c$ is the estimated count using the proposed network, D_c is the ground truth count and t is the threshold.

Here, we aim to reduce the pixel-level error when the crowd estimation count is higher or lower than a $D_c \pm t$. When the threshold is met, we focus on improving the quality of the density map. The threshold is set empirically by observing the type of crowd dataset. In this chapter, we calculate the threshold t based on:

$$t = \frac{\max\{D^n_{c,i}\}}{x},\ where\ D_c \in C_d \tag{6.4}$$

where, D_c is the total number of people in the i^{th} ground truth density map. N is the total number of images in the training dataset C_d and x is set to 4 empirically.

In the following section, we describe the loss function in detail. First we detail the proposed PSNR loss function, then the SSIM followed by the RMSE and Count loss.

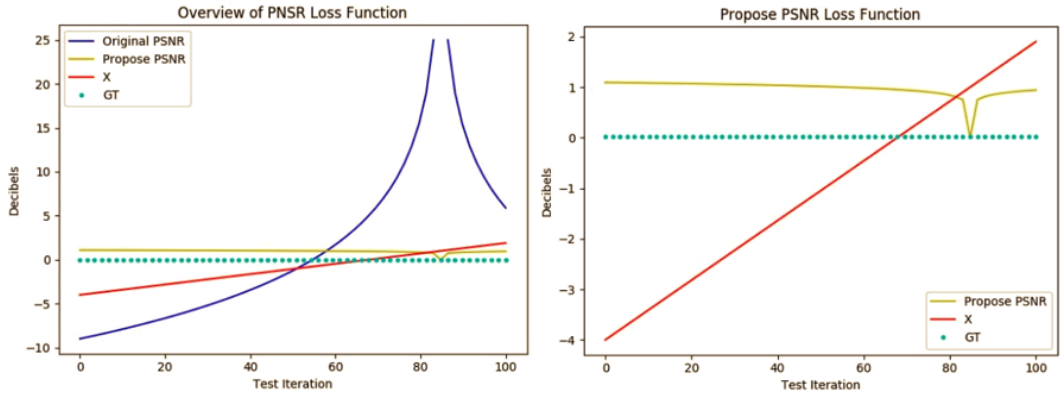


Figure 6.5: (Left) Overview of existing and proposed PSNR loss function. (Right) Proposed PSNR function equation 6.7. The existing PSNR (i.e Original PSNR) value goes to infinity when the MSE approaches zero whereas the proposed PSNR approaches zero.

.

6.2.3.1 PSNR Loss function

The peak signal-to-noise ratio is widely used to evaluate the quality degradation of reconstructed images especially due to lossy compression codecs. In PSNR, the original data is considered as a signal and the error produced by the compression or distortion is regarded as noise. PSNR calculates the ratio between the maximum possible signal value and value of distorting noise, which affects the quality of its representation. The ratio is computed in decibels. Typically, the PSNR is calculated as the logarithm term on the decibel scale as signals have a very wide dynamic range. For image and video, usually for 8-bit colour range, the PSNR values are in the 30-50dB and for 16-bit data, the range is between 60 and 80dB (Deshpande et al. 2018).

$$
\begin{aligned}
MSE &= \frac{1}{N}\sum_{j=0}^{N}(\hat{I}_j - I_j) \\
PSNR &= 10 \times Log_{10} \left(\frac{\gamma^2}{MSE} \right) \\
&= 20 \times Log_{10} \left(\frac{\gamma}{\sqrt{MSE}} \right) \\
&= 20 \times Log_{10}(\gamma) - 10 \times log_{10}(MSE)
\end{aligned}
\tag{6.5}
$$

where N is the total number of pixels, I is ground truth image and $\hat{I}$ is degraded image. j is the pixel location and N is the total number of pixels. Again γ is the maximum variation in the input image data. If the image is 32 bit then the γ is 1. Also, the PSNR value approaches infinity as the MSE approaches zero; this shows that a higher PSNR value provides a higher image quality. However, in Deep Learning lower loss is considered as higher quality,so we need to modify the PSNR accordingly to achieve the desired goal.

$$\hat{MSE} = MSE + \epsilon \tag{6.6}$$

To avoid the infinity problem with PSNR, we introduce ϵ and it is set to 1e$-$10. Using equation 6.6 to calculate the $\hat{MSE}$ for the proposed PSNR equation.

Following equation describes propose PSNR loss function:

$$L_{PSNR} = \left| \frac{\lambda - PSNR}{\lambda} \right| \tag{6.7}$$

where, λ is set empirically through a toy data experiment. In this chapter λ is set to 100.

We experimented with toy data to validate our equation. In our experiment, we have a 64×64, 32-bit floating-point density map D_t. We experimented with 100 different sequentially degraded clones of D_t and assumed it as a predicted density map. The degradation is performed by multiplying D_t by X. The X value ranges from -4 to 1. Figure 6.5(left) shows the overview of all the PSNR loss functions. When the value of X is increased from -4 to 1, which reduces the error $\hat{D}_t$, the value of the original PSNR improves from -10 to inf. The key aspect of the optimiser in Deep Learning is to reduce the loss value, the original PSNR is not suitable for the purpose of learning as the optimiser assumes -10 is the best case scenario which is false. Therefore,

we modify the value of the original PSNR loss function by introducing λ at equation 6.7. Figure 6.5(right) shows the good curve of PSNR value from high to a low when error is reduced. The lower PSNR value represents less noise in the $\hat{D}_t$ density map. As shown in the figure 6.5(right), the value of PSNR goes toward zero when the density map is similar. And, when the density map is identical the PSNR yields zero. We can observe that when the value of x is closer to 1 PSNR decreases in the proposed L_{PNSR} loss function.

6.2.3.2 SSIM Loss function

We also used the SSIM loss function to improve the local correlation in the density, which in turn yields quality results. Usually, the SSIM is used to analyse the quality of an image. Therefore, we exploit the SSIM index to evaluate the local pattern consistency of the predicted density map and the ground truth map. SSIM computes images such as mean, variance and covariance to assess the similarity between two images. The SSIM range is from -1 to 1 and when two images are identical it produces 1.

$$\begin{aligned} SSIM(p) &= \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \cdot \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \\ &= l(p)cs(p) \end{aligned} \tag{6.8}$$

where we omitted the dependence of means and standard deviations on pixel p. Means and standard deviations are computed with a Gaussian filter with standard deviation σ. The loss function for SSIM can be then written by setting $\epsilon(p) = 1 - SSIM(p)$:

$$L_{ssim} = \frac{1}{N} \sum 1 - SSIM(\hat{P}) \tag{6.9}$$

Equation 6.8, shows that the $SSIM(p)$ requires a neighbouring pixel P as large as the support of σ. In other words, in some boundary regions of P, $L_{SSIM}(P)$ and its derivatives cannot be computed. Nevertheless, the convolutional nature of the network allows us to write the loss as

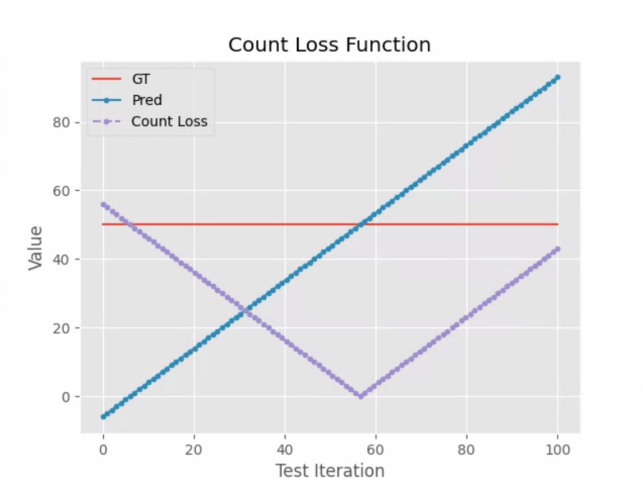


Figure 6.6: Results of count loss function with 100 iterations. GT is ground truth, Pred is predicated count value.

$$L_{SSIM}(P) = 1 - SSIM(\hat{p}) \tag{6.10}$$

where $\hat{p}$ is the centre pixel of path P. Again, this is because, even though the network learns the weights maximising $SSIM$ for the central pixel, the learned kernels are then applied to all the pixels in the image.

- **Root Mean Square Error (RMSE)**: The RMSE loss is used to measure estimation error at the pixel level, which is defined as follows

$$L_{RMSE} = \sqrt{\sum_{i=0}^{n}(\hat{D}_i - D_i)^2} \tag{6.11}$$

Where, $\hat{D}_i$ is the estimated density map, D_i is the ground truth density map, and N is the number of pixels in the density map. The RMSE loss is computed at each pixel and summed. Considering the input image size may be arbitrary in the dataset, the loss value of each sample is normalised by the total pixel number to keep training stable.

6.2.3.3 Count Loss function

We also propose a simple count loss function, which forces the optimiser not only to localise the human heads in a density map, but also the sum of

Figure 6.7: Results of switch loss function with 100 iterations. GT is ground truth, Pred is predicated count. The values have been normalise for the visualisation purpose.

the density map. The loss function will simply take the absolute difference between the sum of ground truth and the estimate in the density map. While the count loss function does not help the network to localise, it does help the network to produce the right sum for a given density map. We use the count loss in combination with RMSE loss L_{RMSE}.

$$L_{count} = \frac{|D_c - \hat{D}_c| + \epsilon}{|D_c + \hat{D}_c| + \epsilon} \tag{6.12}$$

where, D_c is the ground truth for the total number of people and $\hat{D}_c$ is the predicted count. We then normalise the value. We also add ϵ to avoid the infinity problem. Evaluation of the loss function with similar experiments as in the PSNR case and figure 6.6 shows that the error moves towards zero when the prediction is close to ground truth.

- **Final loss functions**: By weighting the above loss functions, we define the final loss function as

$$S_L \begin{cases} Set_a & = L_{RMSE} + L_{COUNT}\eta \\ Set_b & = L_{PSNR}\alpha + L_{SSIM}\beta. \end{cases} \tag{6.13}$$

where we empirically set α, β and η values. α is $1e-4$, β is $1e-3$ and η is $1e-4$. Again, we evaluated the loss function with similar experiments as in

the PSNR case and figure 6.7 shows that the error moves towards zero when the prediction is close to ground truth.

6.2.4 Implementation

As part of data augmentation, we crop the image into 9 small patches from each image at random locations. The patch size is $\frac{1}{4}$ of the original image. Then, we apply random horizontal flip, light augmentation, and noise while training. Our network can be trained end-to-end. For the purpose of pyramid architecture we downsample the patches into $\frac{H}{2} \times \frac{W}{2}, \frac{H}{4} \times \frac{W}{4}, \frac{H}{8} \times \frac{W}{8}$ sizes.

The first 10 convolutions layers in $block_1$ and first 2 layers in other blocks are fine-tuned by pre-trained VGG-16 (Simonyan et al. 2015) weights. For other network weights we use *He normal* (He et al. 2015) to initialise them. We use Adam (Kingma et al. 2015) optimiser with a polynomial decay learning rate, starting at $1e-5$, which gradually decreases at each increasing epoch. The implementation of our method is based on the Keras-TensorFlow framework. Likewise, the proposed network ASVNet can process 8 frames per second for the high quality density prediction in NVIDIA 2080 RTX. Although 8 - 10 fps is almost real-time and quite applicable for real-time application, further research can be done to architecture design to improve the speed as well as to run it in embedded devices such as Nvidia jetson nano. (Cass 2020).

6.3 Experiments and Analysis

In this section, we first present our implementation details and then compare the proposed ASVNet with state-of-the-art networks on three publicly available crowd datasets, Venice (Weizhe Liu et al. 2019), Mall (K. Chen et al. 2012) and UCSD datasets (Antoni B Chan, Liang, et al. 2008). An extensive ablation study is then conducted to reveal the contribution of each component in ASVNet.

We use the mean absolute error (MAE) and the mean squared error (MSE) to evaluate the network performance.

$$MAE = \frac{1}{N}\sum_{i}^{N}|D_i - \hat{D}_i| \tag{6.14}$$

$$MSE = \sqrt{\frac{1}{N}\sum_{i}^{N}(D_i - \hat{D}_i)^2} \tag{6.15}$$

where N is the total number of test images, D_i is the ground-truth count of people in the image, $\hat{D}_i$ is the predicted number of people in the i^{th} image.

Images removed for copyright reasons

Figure 6.8: Results of our network ASVNet. The top 2 rows are from the Venice dataset, the middle 2 rows are Mall dataset and bottom 2 rows are UCSD dataset. The first column is the input image, second is ground truth density map and third is predicted density map.

Method	MAE	MSE
DecideNet (J. Liu et al. 2018)	1.52	1.90
R-FCN† (Jifeng Dai et al. 2016)	6.02	5.46
Faster R-CNN† (Ren et al. 2015)	5.91	660
Count-Forest (Pham et al. 2015)	4.40	2.40
Exemplary-Density (Yi Wang et al. 2016)	1.82	2.74
Boosting-CNN (Walach et al. 2016)	2.01	-
Mo-CNN (Kumagai et al. 2017)	2.75	13.40
Weighted-VLAD (Sheng et al. 2016)	2.41	9.12
ASVNet(Ours)	**1.48**	**1.88**

Table 6.1: The Comparison of performance with other networks over the Mall dataset.† These results are obtained from (J. Liu et al. 2018)

.

6.3.1 Results and Discussions

6.3.1.1 Mall dataset

We compare our ASVNet with several regression-based approaches and the evaluation results are shown in Table 6.1.

From table 6.1, we can observe that our proposed ASVNet obtains the minimum error on both MAE and MSE metrics. Compared to the best approaches 'DecideNet', ASVNet achieves 2.64 % improvement on MAE. This is achieved without using the ensemble scheme employed by the 'MoCNN' (Kumagai et al. 2017) and "Boosting CNN " (Walach et al. 2016) methods. Moreover, the MSE of the ASVNet is only 1.88, which is significantly lower than other state-of-art methods.

6.3.1.2 Venice dataset

We also evaluated our network with the venice dataset. The Venice dataset has a relatively fixed camera angle. However, the number of people in the scene is greater than the Mall dataset. Figure 6.8 shows samples from the dataset, ground truth and predicted density maps. The table 6.2 shows the results of our experiments.

6.3.1.3 UCSD dataset

We also evaluated our network with the UCSD dataset. Similar to the Mall dataset, the video is taken from a fixed angle camera and has an average of 25 people.

Model	MAE	MSE
MCNN (Yingying Zhang et al. 2016)	145.5	147.3
Switch-CNN (Sam, Surya, et al. 2017)	52.8	59.5
CSRNet† (Y. Li et al. 2018)	35.8	50.0
CAN† (Weizhe Liu et al. 2019)	23.5	38.9
ECAN† (Weizhe Liu et al. 2019)	20.5	29.9
ASVNet(Ours)	**18.7**	**26.4**

Table 6.2: The Comparison of the performance of our network with other networks in the Venice dataset. †These results are obtained from (Weizhe Liu et al. 2019).

Model	MAE	MSE
MCNN (Yingying Zhang et al. 2016)	1.07.5	1.35
Bidirectional ConvLSTM (Xiong et al. 2017)	1.13	1.43
CSRNet (Y. Li et al. 2018)	1.16	1.47
SANet (Cao et al. 2018)	1.02	1.29
E3D (Zou et al. 2019)	0.93	1.17
ASVNet(Ours)	**0.91**	**1.14**

Table 6.3: The Comparison of the performance of our network with other networks in the UCSD dataset.

6.4 Ablation

An ablation study was carried out on the UCSD dataset. This provides further analysis of the relative contribution of a given component in our architecture. In order to validate the effectiveness of the pyramid context module, the scale aggregation module and the self-attention module, we train variants of our model and conduct the experiments.

6.4.0.1 Effectiveness of the Baseline

Our baseline network consists of Block-1 with 10 layers of VGG-16 without pyramid modules, scale aggregation and scale attention module. The module takes a single image that is fed to block-1. The extracted features $\{f_{10}\}$ directly pass to the backend Cov4K_3D_2 dilation block. Table 6.4 shows that with just the single VGG-16 as the encoder, the results are not promising compared with other modules.

Model	MAE	MSE
Baseline	1.69	3.37
Baseline + PCM	1.51	3.32
Baseline + PCM+SAGM	1.49	1.82
Baseline + PCM+SAM	1.23	1.54
Baseline + PCM+ SAGM + SAM	0.91	1.14

Table 6.4: The table shows the Comparison between Baseline,Pyramid context module (PCM),Scale Aggregation module (SAGM) and Self-attention module (SAM).

6.4.0.2 Effectiveness of Pyramid context module (PCM)

In this experiment, we divided the image into 3 different scales and the original image is fed to block-1 and reset to other blocks {2..4} respectively. As a result the network is context-aware and the performance of the network has improved significantly. Table 6.4 shows the results of the baseline against the pyramid context module.

6.4.0.3 Effectiveness of Scale aggregation module (SAGM)

We then compared the results of the model with the pyramid context module and attention module, with and without the scale aggregation module. The scale aggregation module has improved the results of the pyramid context modules, however it alone cannot achieve better results. Table 6.4 shows that the scale aggregation module is more accurate than both Baseline and Baseline + PCM networks.

6.4.0.4 Effectiveness of Self-attention module (SAM)

We analyse the contribution of the self-attention module in our model. In table 6.4, we show that the addition of the self-attention modules on baseline + PCM has performed better than baseline + PCM as well as the baseline + PMC + SAGM modules. The results demonstrate the self-attention module has a greater contribution to the final results, and we achieve the best performance.

6.5 Summary

In this chapter, we presented an adaptive scale variance network (ASVNet) for crowd counting, which deals with the very large variation in people size in a scene.

To exploit the scale variance we propose a pyramid module, which can fully encode the contextual information. We also embed the adaptive scale aggregation and self-attention modules to automatically choose the most appropriate branches and naturally enlarge the receptive field. Likewise, we propose a novel loss function we called: switch loss function which utilises a combination of novel loss peak signal-to-noise loss, local consistency loss, a root mean squared error loss and count loss to alternate from one set of loss to another improving the quality of density maps. Extensive experiments show that our method achieves the best performance to state-of-the-art methods on three crowd counting benchmarks.

"If you fell down yesterday, stand up today."

— H. G Wells

7 Conclusions and Future Work

Contents

In this thesis, we examined various aspects of the scene analysis, in particular we analysed the challenges that exist in the human detection and counting field. We introduced the scopes and difficulties in human detection and analysis in chapter 1 and in chapter 2, we explored the essential concepts that are present in the field of pedestrian and crowd analysis. From chapter 3 to 6, we proposed our solution toward the challenges that exist in human analysis where we concentrate the majority of research in the synthetic data generation and crowd analysis field.

7.1 Summary of Thesis Achievements

In this thesis, we have provided our solutions towards the challenges that are present in crowd estimation and counting. We tried to tackle four core aspects of the problem :

Firstly, we looked at the inadequacy of crowd dataset in the field and proposed a multi-purpose synthetic data generation tool which leverages the real-time graphics

engine and can be used to produce large amounts of mix-reality dataset. The important aspect of the technique is that it not only can be used to generate crowd related data but also for other computer vision problems such as pedestrian detection, 3D pose estimation, image segmentation and depth estimation. In addition, due to the use of real data for the background we also increase the exposure of non-artificial data to the network which in turn performs better in real-world scenarios. Finally, we demonstrated that use of synthetic data can improve performance of state-of-the-art results and achieve higher accuracy.

Secondly, we explored the problem of inter-reflection and proposed a novel approach of utilising the reverted ray tracing mechanism to reduce the effect of such inter-reflection. The primary contribution is that we demonstrated the method to capture the environment noises present in the data generation process and the ways to reduce it from the final results. While the chapter is fully focused on removing inter-reflection, the global goal of improving crowd estimation can be achieved through better quality of synthetic data. Hence, the chapter achieves its aim in improving such data.

Thirdly, we proposed a solution for the limitation that exists in the density map generation process. At current, almost all the researchers use the density generation process which is not content-aware i.e. the head size in the density map does not correlate with the actual head size in the image. Hence, we proposed a content aware density map with the combination of the nearest neighbour algorithm and unsupervised segmentation to generate the density map head masks. Furthermore, we illustrated that simply modifying the method of density map generation can improve the performance of state-of-the-art networks.

Finally, we tried to handle the difficult challenge that was present in the crowd scene, perspective distortion. For this, we propose entirely new deep learning architecture which employs a number of clever techniques to minimise the effect of perspective distortion in crowd estimation and counting. We propose a deep learning architecture with an effective way to capture multi-scale features by the use of pyramid contextual modules in combination with scale aggregation and self-attention

mechanism. In addition, we also proposed a novel loss function *Switch loss function* to maximise the quality of the predicted density map and accuracy. The loss function utilises a combination of multiple functions such as PSNR, SSIM, Root mean square errors to achieve higher quality density map and accuracy. We also illustrated that by using a variation of the above method we can achieve state-of-the-art results.

7.2 Future Work

There are diverse areas where future work can focus on, however handful of the notable fields whereas immediate focus can go are discussed below:

- In chapter 3, we saw a difference between the network performance with grays and RGB image as input. We observed that networks which utilised RGB images were able to learn more from the generated data than the network which used grayscale images. Therefore, additional research can be performed to validate the observation by training more varieties of crowd counting networks.

- Further research is needed on the proposed method described in chapter 4. Although we have presented the core aspect of the IRT-PS method which describes a method to capture the inter-reflection, a more detailed analysis should be carried out to improve the scope of the proposed method. Additional work can be done to evaluate the performance of the IRT-PS method and identify other applications for the method.

- Likewise, in chapter 6, we proposed an ASVNet architecture for crowd counting. Due to the presence of millions of parameters in the proposed network, the AVSNet does not perform inference in real-time. Hence, further work can be done on the research using the Generative Adversarial network (GAN) to improve the quality of the density map as well as reduce the size of the network and preserve or improve the model accuracy.

- In the thesis, while we demonstrated that the synthetic data can elevate the results of existing deep learning networks, it was also clear that the synthetic data alone was not enough for deep learning models to generalise well. Crowd events and large gatherings in current time are rather common and it does not take much man power to capture such video. Nevertheless, it is difficult to manually annotate such large data. Therefore, further research can be carried out in methods which can leverage the existing deep learning model to roughly estimate the crowd in the unlabelled data and allow humans to fine tune the head localisation. This can not only save time but might also help to annotate more accurate dataset.

- At present, evaluation of crowd estimation and counting are only carried out in either single images or video. Techniques developed for one are generally not well suited for the other purpose. Currently Deep Learning architectures that are designed are primarily focused on particular dataset and usually are not generalised enough to be used in crowd scenes. Therefore, developing the approach which combines both image and video and general enough is not a trivial task and more research can be done in this area.

- Another area where crowd analysis suffers is its performance in terms of real-world applicability. As in most scenarios such as security and critical application requires real-time analysis, further intuitive techniques should be the priority for researchers to deliver real-time crowd analysis.

Appendices

"One finds limits by pushing them."

—Herbert Simon

A
Appendix 1

Contents

A.1 Synthetic data generation method

A.1.1 Synthetic crowd with real-world background

We can generate a variety of images with a proposed synthetic data generation method discussed in chapter 3. For the following sample, we utilised publicly available images of *Place de la comédie montpellier* as background with a varying day and night time. In addition, the light in the scene can also be adjusted according to the background which helps to generate realistic shadows.

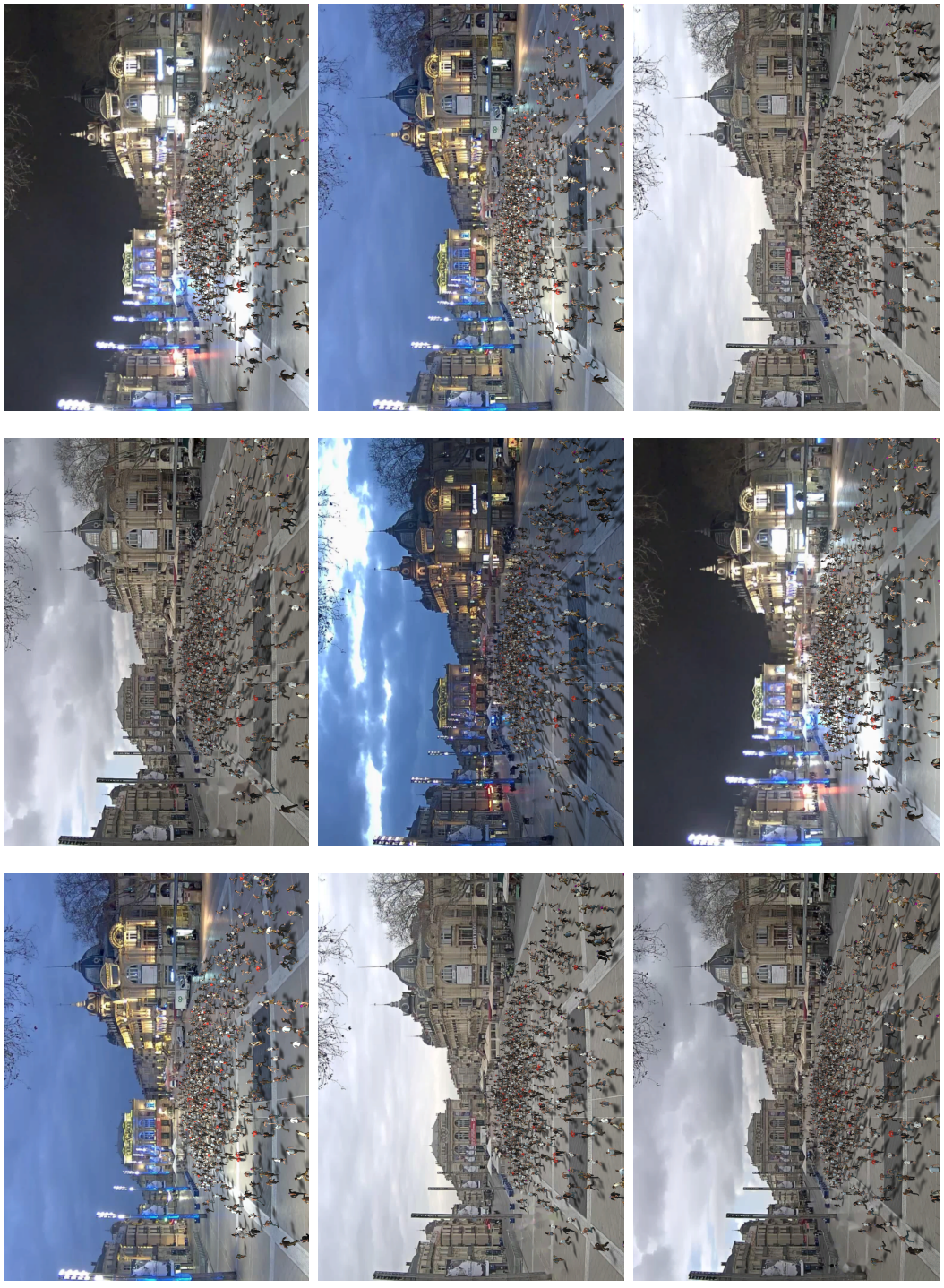


Figure A.1: Sample image generated by synthetic data generation method. 1200 agents were simulated in the image.

Figure A.2: The images show the automatic head centre annotation of the agent using the proposed method.

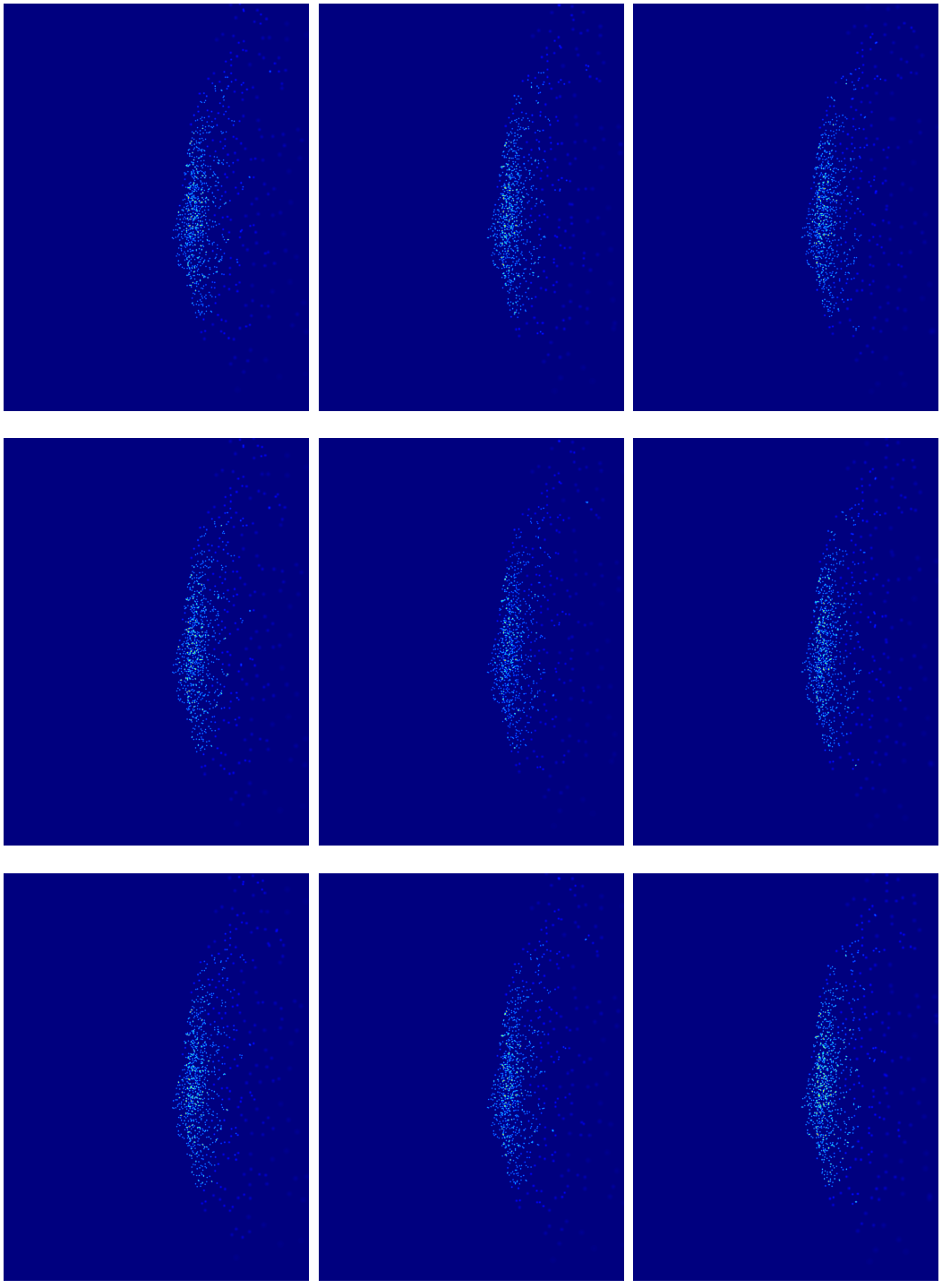


Figure A.3: Based on head annotation, further data such as density maps can be generated for crowd counting purposes.

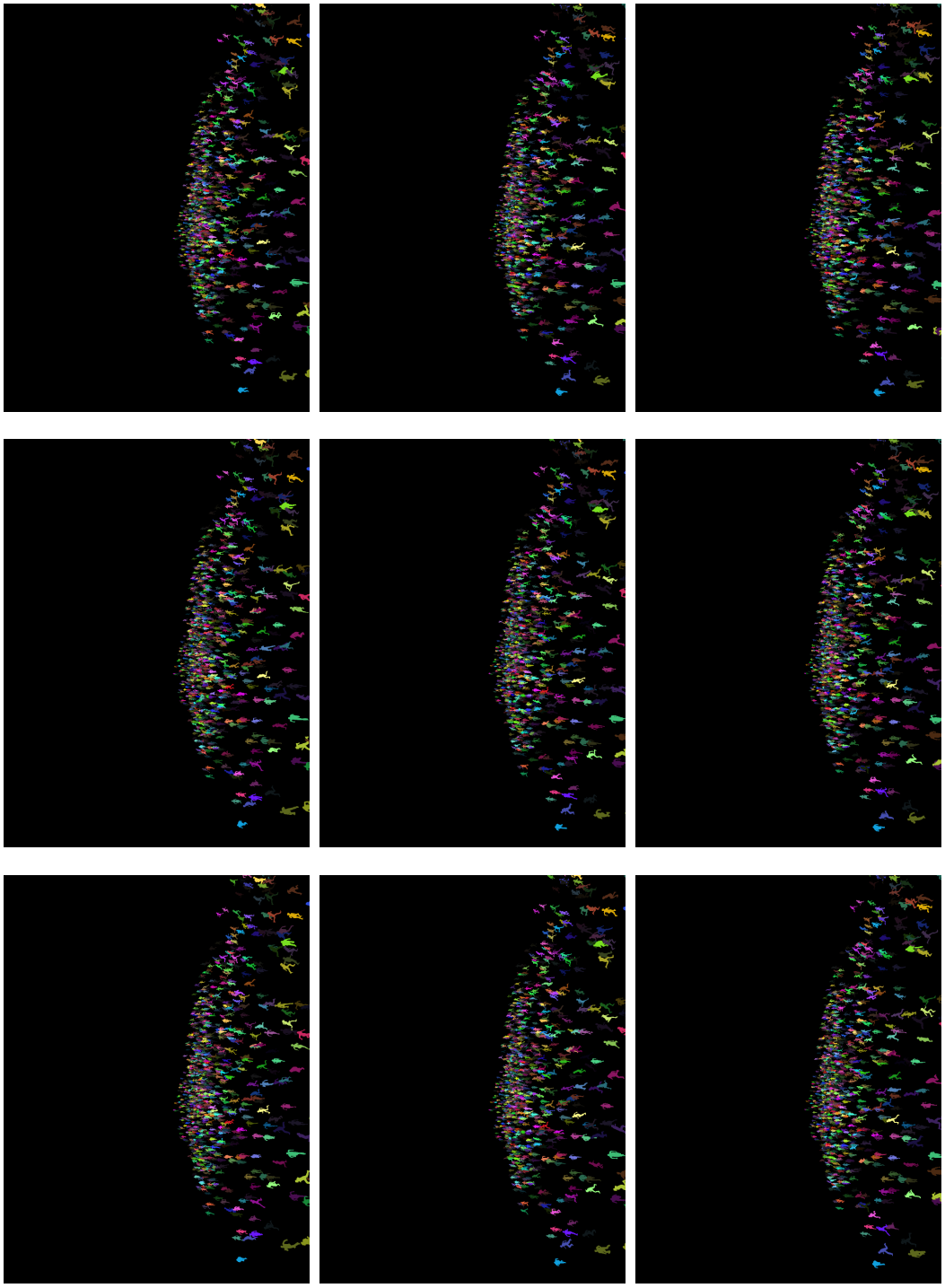


Figure A.4: Sample image shows the automatic image segmentation where all agents have unique colour.

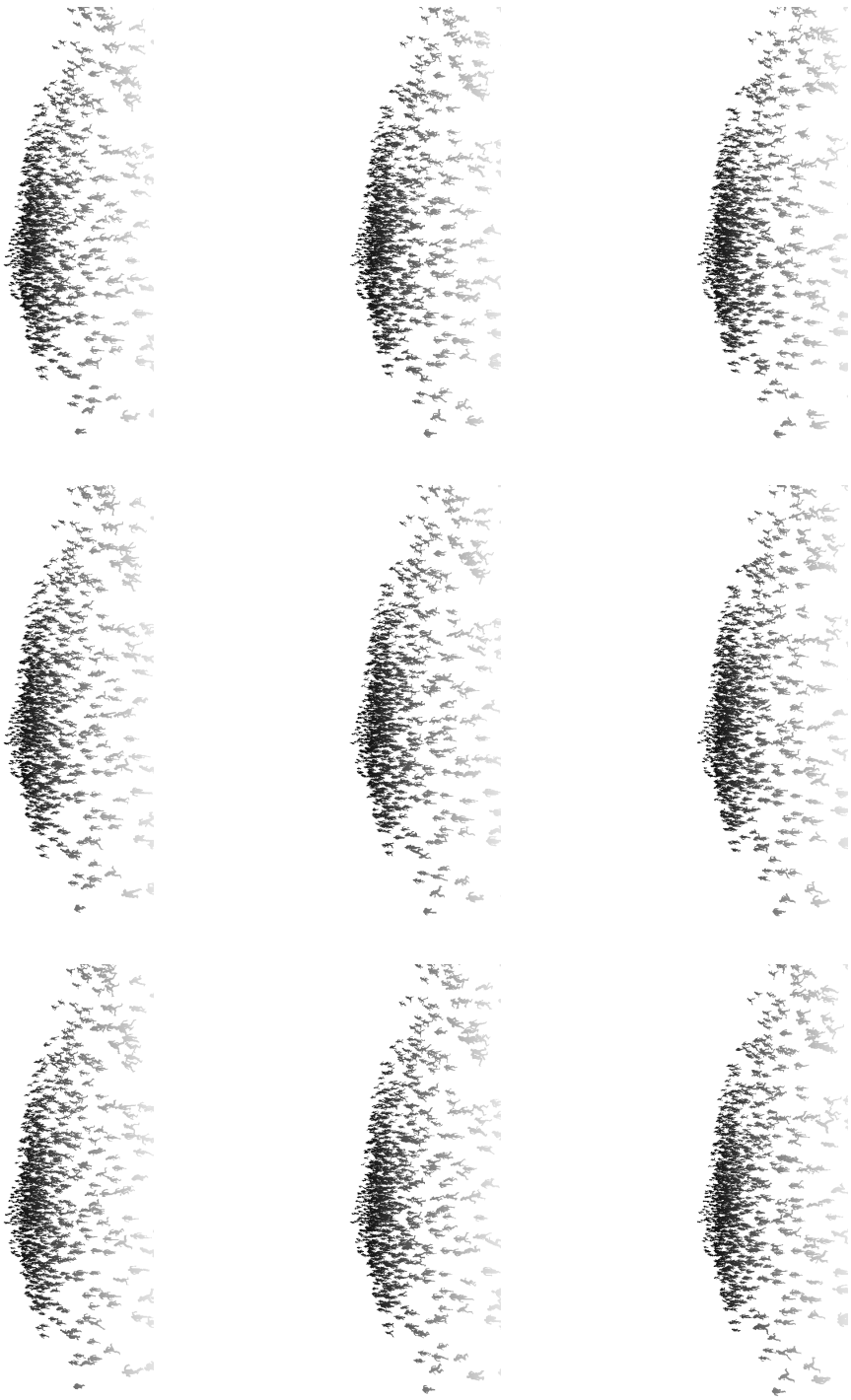


Figure A.5: Depth map can also be generated using the synthetic data generation method.

Figure A.6: Various types of agents were used to generate the crowd. Few of the 3D models are presented in the figure.

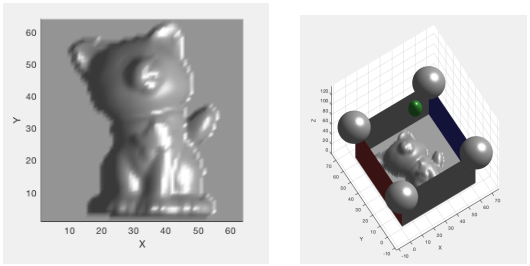


Figure A.7: As described in chapter 4, a 3D environment is set up for capturing photometric stereo images. Four spheres at the core of the box represent lights and a green sphere at the centre represents a camera

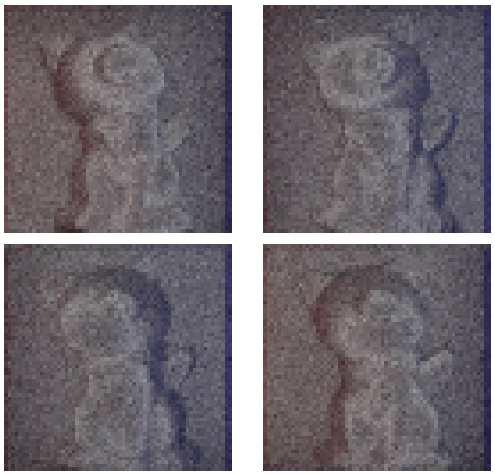


Figure A.8: We captured four different images based on varying light and later used these images for 3D reconstruction.

Algorithm 1: IRT-PS algorithms

Input: Images with inter-reflections i
Output: 3D mesh from images

Function `ClassisPS(`i`):`
    `/* Generate 3D surface` S `using images` i `*/`
    S = NormalToDepth(i)
    return S

`/* Simulate 3D environment using techniques such as Spherical domo projection. Requires Environment information` Env_{Info} `such as HDR images */`
Function `SetEnvironment(`S, Env_{Info}`):`
    E = Simulate3DEnv(Env_{Info})
    `/* Add generated surface` S `inside the 3D environment` E `*/`
    if S *!= none* **then**
        E += S
    return E

`/* capture the environment colour within 3D environment` E `*/`
Function `CaptureInterReflection(`E`):`
    $\hat{i}$ = revertedRaytraying(E)
    return $\hat{i}$

`/* Reduce the inter-reflection from original input image` i `using new generated image` $\hat{i}$ `*/`
Function `RemoveInterReflection(`i, $\hat{i}$`):`
    $i = i - \hat{i}$
    return i

Function `Main:`
    i = Images with inter-reflection
    Env_{Info} = Environment Texture map, Camera specification
    Env = SetEnvironment(Env_{Info})
    `/* Set the total loop` t `*/`
    $t = 3$
    while $t > 0$ **do**
        S = ClassisPS(i)
        Env = SetEnvironment(S, Env_{Info})
        $\hat{i}$ = CaptureInterReflection(Env)
        i = RemoveInterReflection(i, $\hat{i}$)
        $t = t - 1$
    return 0

Algorithm 2: Chan-Vese algorithms

Input: Head annotated images i
Output: Image segmentation

Function `CalChanVese`(i, μ λ_1, λ_2, t)**:**
  $\hat{i}$ = initValue(i,c_1,c_2)
  `/* It calculate the total energy of the current level */`
  $energy$ = calEnergy(i, $\hat{i}$,μ, λ_1, $_2$)
  `/*` S_i `is an image segmentation */`
  $S_i = \hat{i} > 0$
  $x = 0$
  `/* The loop iterates till` x `<` t `*/`
  while $x < t$ **do**
    $\hat{i}$ = calculateVariation(i, $\hat{i}$, μ, λ_1,λ_2)
    $\hat{i}$ = resetLevelSet($\hat{i}$)
    $S_i = \hat{i} > 0$
    $energy$ = calEnergy(i, $\hat{i}$, μ, λ_1, $_2$)
    $x = x + 1$
  return S_i

Function `Main`**:**
  i = Greyscale image
  G = 5x5 pixel centered to head
  c_1 = Initial average pixels intensity inside G region
  c_2 = Initial average pixels intensity of square box G which extends to the nearest head point
  `/* Parameters are set according to (T. F. Chan et al. 2001) */`

  $\lambda_1 = 1$
  $\lambda_2 = 1$
  $\mu = 0$
  `/* User defined parameter; Total iteration to calculate the segmentation */`

  $t = 500$
  segmentation = CalChanVese(i, μ λ_1, λ_2, t)
  return 0

"The science of today is the technology of tomorrow"

— Edward Teller

Bibliography

Abdelghany, Ahmed et al. (2014). "Modeling framework for optimal evacuation of large-scale crowded pedestrian facilities". In: *European Journal of Operational Research* 237.3, pp. 1105–1118.

Aksoy, Yağiz et al. (2016). "Interactive High-Quality Green-Screen Keying via Color Unmixing". In: *ACM Transactions on Graphics* 36.4, 61b:1.

Alsalam, Bilal Hazim Younus et al. (2017). "Autonomous UAV with vision based on-board decision making for remote sensing and precision agriculture". In: *Proceeding of IEEE Aerospace Conference*. 2017 IEEE Aerospace Conference, pp. 1–12.

Ancheta, Roxanne A et al. (2018). "FEDSecurity: Implementation of Computer Vision Thru Face and Eye Detection". In: *International Journal of Machine Learning and Computing* 8.6, p. 6.

Arakeri, Megha. P. and Lakshmana (2016). "Computer Vision Based Fruit Grading System for Quality Evaluation of Tomato in Agriculture industry". In: *Procedia Computer Science*. Proceedings of International Conference on Communication, Computing and Virtualization (ICCCV) 2016 79, pp. 426–433.

Argyriou, Vasileios and Maria Petrou (2008). "Recursive photometric stereo when multiple shadows and highlights are present". In: *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*. IEEE, pp. 1–6.

Argyriou, Vasileios, Maria Petrou, and Svetlana Barsky (2010). "Photometric stereo with an arbitrary number of illuminants". In: *Computer Vision and Image Understanding* 114.8, pp. 887–900.

Babu Sam, Deepak et al. (2018). "Divide and grow: Capturing huge diversity in crowd images with incrementally growing cnn". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3618–3626.

Belhumeur, Peter N. and David J. Kriegman (1998). "What is the set of images of an object under all possible illumination conditions?" In: *International Journal of Computer Vision* 28.3. Publisher: Springer, pp. 245–260.

Benfold, Ben and Ian Reid (2011). "Stable multi-target tracking in real-time surveillance video". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. IEEE, pp. 3457–3464.

Bhola, Punit Kumar et al. (2019). "Flood inundation forecasts using validation data generated with the assistance of computer vision". In: *Journal of Hydroinformatics* 21.2, pp. 240–256.

Bilinski, Piotr and Francois Bremond (2016). "Human violence recognition and detection in surveillance videos". In: *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 30–36.

Bloom, Victoria, Vasileios Argyriou, and Dimitrios Makris (2017). "Linear latent low dimensional space for online early action recognition and prediction". In: *Pattern Recognition* 72, pp. 532–547.

Bloom, Victoria, Dimitrios Makris, and Vasileios Argyriou (2014). "Clustered Spatio-temporal Manifolds for Online Action Recognition". In: *Proceedings of the 22nd International Conference on Pattern Recognition*. 2014 22nd International Conference on Pattern Recognition. ISSN: 1051-4651, pp. 3963–3968.

Boominathan, Lokesh, Srinivas SS Kruthiventi, and R Venkatesh Babu (2016). "Crowdnet: A deep convolutional network for dense crowd counting". In: *Proceedings of the 24th ACM international conference on Multimedia*. ACM, pp. 640–644.

Bourke, Paul (2005). "Spherical mirror: a new approach to hemispherical dome projection". In: *Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*. GRAPHITE '05. New York, NY, USA: Association for Computing Machinery, pp. 281–284.

Bouwmans, Thierry, Lucia Maddalena, and Alfredo Petrosino (2017). "Scene background initialization: A taxonomy". In: *Pattern Recognition Letters* 96, pp. 3–11.

Butler, Daniel J., Jonas Wulff, and Michael J. Stanley Garrett B.and Black (2012). "A Naturalistic Open Source Movie for Optical Flow Evaluation". In: *Computer Vision – ECCV 2012*. Ed. by Andrew Fitzgibbon et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 611–625.

Cao, Xinkun et al. (2018). "Scale aggregation network for accurate and efficient crowd counting". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 734–750.

Cass, Stephen (2020). "Nvidia makes it easy to embed AI: The Jetson nano packs a lot of machine-learning power into DIY projects - [Hands on]". In: *IEEE Spectrum* 57.7. Conference Name: IEEE Spectrum, pp. 14–16.

Cerri, Pietro et al. (2010). "Day and night pedestrian detection using cascade AdaBoost system". In: *13th International IEEE Conference on Intelligent Transportation Systems*. 13th International IEEE Conference on Intelligent Transportation Systems. ISSN: 2153-0017, pp. 1843–1848.

Chan, Antoni B, Zhang-Sheng John Liang, and Nuno Vasconcelos (2008). "Privacy preserving crowd monitoring: Counting people without people models or tracking". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1–7.

Chan, Antoni B and Nuno Vasconcelos (2009). "Bayesian poisson regression for crowd counting". In: *Proceeding of the IEEE 12th International Conference on Computer Vision*. IEEE, pp. 545–551.

— (2011). "Counting people with low-level features and Bayesian regression". In: *IEEE Transactions on Image Processing* 21.4, pp. 2160–2177.

— (2008). "Modeling, Clustering, and Segmenting Video with Mixtures of Dynamic Textures". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.5. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 909–926.

Chan, Tony F and Luminita A Vese (2001). "Active contours without edges". In: *IEEE Transactions on image processing* 10.2, pp. 266–277.

Chang, Chung-Liang and Kuan-Ming Lin (2018). "Smart Agricultural Machine with a Computer Vision-Based Weeding and Variable-Rate Irrigation Scheme". In: *Robotics* 7.3, p. 38.

Change Loy, Chen, Shaogang Gong, and Tao Xiang (2013). "From semi-supervised to transfer counting of crowds". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV'13)*, pp. 2256–2263.

Chen, Jun-Cheng et al. (2016). "A cascaded convolutional neural network for age estimation of unconstrained faces". In: *Proceedings of the IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS'16)*. IEEE, pp. 1–8.

Chen, Ke et al. (2012). "Feature mining for localised crowd counting." In: *Proceeding of British Machine Vision Conference BMVC*. Vol. 1. Issue: 2, p. 3.

Chen, W. et al. (2016). "Synthesizing Training Images for Boosting Human 3D Pose Estimation". In: *Proceedings of the Fourth International Conference on 3D Vision (3DV)'16*, pp. 479–488.

Cheng, Zhi-Qi et al. (2019). "Learning spatial awareness to improve crowd counting". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6152–6161.

Cheung, Ernest et al. (2018). "Mixedpeds: pedestrian detection in unannotated videos using synthetically generated human-agents for training". In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

Chollet, François (2018). *Deep learning with Python*. OCLC: ocn982650571. Shelter Island, New York: Manning Publications Co. 361 pp.

Chow, W. K. and Candy M. Y. Ng (2008). "Waiting time in emergency evacuation of crowded public transport terminals". In: *Safety Science* 46.5, pp. 844–857.

Ciregan, Dan, Ueli Meier, and Jürgen Schmidhuber (2012). "Multi-column deep neural networks for image classification". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR'12)*. IEEE, pp. 3642–3649.

Conte, D. et al. (2010). "A Method Based on the Indirect Approach for Counting People in Crowded Scenes". In: *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 111–118.

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks". In: *Machine Learning* 20.3, pp. 273–297.

Cust, Emily E et al. (2019). "Machine and deep learning for sport-specific movement recognition: a systematic review of model development and performance". In: *Journal of Sports Sciences* 37.5, pp. 568–600.

D. Souza, C. R. et al. (2017). "Procedural Generation of Videos to Train Deep Action Recognition Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, pp. 2594–2604.

Dai, Jifeng et al. (2016). "R-fcn: Object detection via region-based fully convolutional networks". In: *Advances in neural information processing systems*, pp. 379–387.

Dalal, N. and B. Triggs (2005). "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 1. ISSN: 1063-6919, 886–893 vol. 1.

Dee, Hannah M. and Alice Caplier (2010). "Crowd behaviour analysis using histograms of motion direction". In: *Proceedings of the IEEE International Conference on Image Processing (ICIP'10)*. Proceedings of the IEEE International Conference on Image Processing. ISSN: 2381-8549, pp. 1545–1548.

Deng, Li (2014). "A tutorial survey of architectures, algorithms, and applications for deep learning". In: *APSIPA Transactions on Signal and Information Processing* 3. Publisher: Cambridge University Press.

Deshpande, Renuka G, Lata L Ragha, and Satyendra Kumar Sharma (2018). "Video Quality Assessment through PSNR Estimation for Different Compression Standards". In: *Indonesian Journal of Electrical Engineering and Computer Science* 11.3, pp. 918–924.

Dey, Sandipan (2018). *Hands-On Image Processing with Python: Expert techniques for advanced image analysis and effective interpretation of image data*. Packt Publishing Ltd.

Dollar, Piotr et al. (2011). "Pedestrian detection: An evaluation of the state of the art". In: *IEEE transactions on pattern analysis and machine intelligence* 34.4, pp. 743–761.

Dosovitskiy, Alexey et al. (2015). "FlowNet: Learning Optical Flow with Convolutional Networks". In: *2015 IEEE International Conference on Computer Vision ICCV*, pp. 2758–2766.

Dupre, Rob and Vasileios Argyriou (2019). "A human and group behavior simulation evaluation framework utilizing composition and video analysis". In: *Computer Animation and Virtual Worlds* 30.1. e1844 cav.1844, e1844.

Errami, Mounir and Mohammed Rziza (2016). "Improving Pedestrian Detection Using Support Vector Regression". In: *Proceedings of the 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV'16)*. 2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV), pp. 156–160.

Esteban, Carlos Hernández, George Vogiatzis, and Roberto Cipolla (2008). "Multiview Photometric Stereo". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, pp. 548–554.

Fang, Weili et al. (2018). "Falls from heights: A computer vision-based approach for safety harness detection". In: *Automation in Construction* 91, pp. 53–61.

Felzenszwalb, Pedro F et al. (2009). "Object detection with discriminatively trained part-based models". In: *IEEE transactions on pattern analysis and machine intelligence* 32.9, pp. 1627–1645.

Foley, James D. et al. (1996). *Computer Graphics: Principles and Practice*. Addison-Wesley Professional. 1294 pp.

Frankot, Robert T. and Rama Chellappa (1988). "A method for enforcing integrability in shape from shading algorithms". In: *IEEE Transactions on pattern analysis and machine intelligence* 10.4. Publisher: IEEE, pp. 439–451.

Freeman, William T. and Michal Roth (1995). "Orientation histograms for hand gesture recognition". In: *Proceedings of the International workshop on automatic face and gesture recognition*. Vol. 12, pp. 296–301.

French, Geoffrey et al. (Sept. 10, 2015). "Convolutional Neural Networks for Counting Fish in Fisheries Surveillance Video". In: *Proceedings of the Machine Vision of Animals and their Behaviour (MVAB)*. Ed. by Robert Fisher. Vol. 7.1-7.10. Conference Name: Workshop on Machine Vision of Animals and their Behaviour Meeting Name: Workshop on Machine Vision of Animals and their Behaviour. GBR: BMVA Press.

Friedman, Jerome H (2001). "Greedy function approximation: a gradient boosting machine". In: *Annals of statistics*, pp. 1189–1232.

Fu, Min et al. (2015). "Fast crowd density estimation with convolutional neural networks". In: *Engineering Applications of Artificial Intelligence* 43, pp. 81–88.

Gao, Junyu, Tianzhu Zhang, and Changsheng Xu (2017). "A Unified Personalized Video Recommendation via Dynamic Recurrent Neural Networks". In: *Proceedings of the*

25th ACM international conference on Multimedia. MM '17. Mountain View, California, USA: Association for Computing Machinery, pp. 127–135.

Ge, Weina and Robert T Collins (2009). "Marked point processes for crowd counting". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2913–2920.

Ge, Weina, Robert T. Collins, and R. Barry Ruback (2012). "Vision-Based Analysis of Small Groups in Pedestrian Crowds". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 34.5. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1003–1016.

Goodfellow, Ian et al. (2014). "Generative adversarial nets". In: *Advances in neural information processing systems*, pp. 2672–2680.

Gotovac, Sven, Vladan Papić, and Željko Marušić (2016). "Analysis of saliency object detection algorithms for search and rescue operations". In: *2016 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. 2016 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM). ISSN: 1847-358X, pp. 1–6.

Grant, Jason M. and Patrick J. Flynn (2017). "Crowd Scene Understanding from Video: A Survey". In: *ACM Transactions on Multimedia Computing, Communications, and Applications* 13.2, 19:1–19:23.

Guo, Yangyang, Dongjian He, and Lilong Chai (2020). "A Machine Vision-Based Method for Monitoring Scene-Interactive Behaviors of Dairy Calf". In: *Animals* 10.2. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, p. 190.

Hattori, Hironori et al. (2015). "Learning scene-specific pedestrian detectors without real data". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*, pp. 3819–3827.

Hayakawa, Hideki (1994). "Photometric stereo under a light source with arbitrary motion". In: *JOSA A* 11.11. Publisher: Optical Society of America, pp. 3079–3089.

He, Kaiming et al. (2015). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

— (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR'16)*, pp. 770–778.

Heaton, Jeff (2015). *Artificial intelligence for humans. volume 3 volume 3*. OCLC: 964654220. St. Louis, MO, USA: Heaton Research, Inc.

Heimberger, Markus et al. (2017). "Computer vision in automated parking systems: Design, implementation and challenges". In: *Image and Vision Computing*. Automotive Vision: Challenges, Trends, Technologies and Systems for Vision-Based Intelligent Vehicles 68, pp. 88–101.

Hekler, Achim et al. (2019). "Deep learning outperformed 11 pathologists in the classification of histopathological melanoma images". In: *European Journal of Cancer* 118, pp. 91–96.

Herbort, Steffen and Christian Wöhler (2011). "An introduction to image-based 3D surface reconstruction and a survey of photometric stereo methods". In: *3D Research* 2.3, p. 4.

Hestness, Joel et al. (2017). "Deep Learning Scaling is Predictable, Empirically". In: *ArXiv* abs/1712.00409.

Hicks, M. D. et al. (2011). "A photometric function for analysis of lunar images in the visual and infrared based on Moon Mineralogy Mapper observations". In: *Journal of*

Geophysical Research: Planets 116 (E6). _eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2010JE003733.

Ho, George To Sum et al. (2019). "A Computer Vision-Based Roadside Occupation Surveillance System for Intelligent Transport in Smart Cities". In: *Sensors* 19.8, p. 1796.

Hoang, Toan et al. (2016). "Road Lane Detection by Discriminating Dashed and Solid Road Lanes Using a Visible Light Camera Sensor". In: *Sensors* 16.8, p. 1313.

Hope, Tom, Yehezkel S Resheff, and Itay Lieder (2017). *Learning TensorFlow: a guide to building deep learning systems*. OCLC: 1001371602. Beijing: O'Reilly.

Horn, B.K.P. (1977). "Understanding Image Intensities". In: *Artificial Intelligence* 8.11, pp. 201–231.

Hossain, Mohammad et al. (2019). "Crowd Counting Using Scale-Aware Attention Networks". In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, pp. 1280–1288.

Hosseini, Mohammad-Parsa et al. (2020). "Deep Learning Architectures". In: *Deep Learning: Concepts and Architectures*. Ed. by Witold Pedrycz and Shyi-Ming Chen. Studies in Computational Intelligence. Cham: Springer International Publishing, pp. 1–24.

Hua, Kai-Lung et al. (2018). "Background Extraction Using Random Walk Image Fusion". In: *IEEE Transactions on Cybernetics* 48, pp. 423–435.

Huang, Siyu et al. (2020). "Stacked Pooling for Boosting Scale Invariance of Crowd Counting". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'20)*, pp. 2578–2582.

Huang, Zhanyuan et al. (2018). "Video-based Fall Detection for Seniors with Human Pose Estimation". In: *2018 4th International Conference on Universal Village (UV)*. 2018 4th International Conference on Universal Village (UV), pp. 1–4.

Idrees, Haroon, Imran Saleemi, et al. (2013). "Multi-source multi-scale counting in extremely dense crowd images". In: *Proceedings of the IEEE conference on Computer Vision and pattern Recognition (CVPR'13)*, pp. 2547–2554.

Idrees, Haroon, Muhmmad Tayyab, et al. (2018). "Composition loss for counting, density map estimation and localization in dense crowds". In: *Proceedings of the European Conference on Computer Vision (ECCV)'18*, pp. 532–546.

Ikeuchi, Katsushi (1981). "Determining Surface Orientations of Specular Surfaces by Using the Photometric Stereo Method". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-3, pp. 661–669.

Illiyas, Faisel T. et al. (2013). "Human stampedes during religious festivals: A comparative review of mass gathering emergencies in India". In: *International Journal of Disaster Risk Reduction* 5, pp. 10–18.

Immel, David S., Michael F. Cohen, and Donald P. Greenberg (1986). "A radiosity method for non-diffuse environments". In: *Proceedings of the SIGGRAPH,'86*.

Insider (2020). *Why 'The Mandalorian' Uses Virtual Sets Over Green Screen | Movies Insider*. URL: https://www.youtube.com/watch?v=Ufp8weYYDE8 (visited on 02/06/2021).

Isola, Phillip et al. (2017). "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE conference on Computer Vision and pattern Recognition (CVPR'17)*, pp. 1125–1134.

Jarosz, Wojciech, Henrik Wann Jensen, and Craig Donner (2008). "Advanced global illumination using photon mapping". In: *ACM SIGGRAPH 2008 classes*. Proceddings

of the SIGGRAPH '08. New York, NY, USA: Association for Computing Machinery, pp. 1–112.

Jiang, S. et al. (2020). "Mask-Aware Networks for Crowd Counting". In: *IEEE Transactions on Circuits and Systems for Video Technology* 30.9. Conference Name: IEEE Transactions on Circuits and Systems for Video Technology, pp. 3119–3129.

Jin, Zhixing and Bir Bhanu (2012). "Single camera multi-person tracking based on crowd simulation". In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR'12)*. Proceedings of the 21st International Conference on Pattern Recognition (ICPR'12). ISSN: 1051-4651, pp. 3660–3663.

Joshi, Ameet V. (2020). "Perceptron and Neural Networks". In: *Machine Learning and Artificial Intelligence*. Ed. by Ameet V Joshi. Cham: Springer International Publishing, pp. 43–51.

Kajiya, James T. (1986). "The rendering equation". In: *Proceedings of the 13th annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, pp. 143–150.

Kamble, P. R., A. G. Keskar, and K. M. Bhurchandi (2019). "A deep learning ball tracking system in soccer videos". In: *Opto-Electronics Review* 27.1, pp. 58–69.

Kamilaris, Andreas and Francesc X. Prenafeta-Boldú (2018). "Disaster Monitoring using Unmanned Aerial Vehicles and Deep Learning". In: *EnviroInfo*.

Kang, Di and Antoni Chan (2018). "Crowd counting by adaptively fusing predictions from an image pyramid". In: *29th British Machine Vision Conference (BMVC'18)*.

Kang, Di, Zheng Ma, and Antoni B Chan (2018). "Beyond Counting: Comparisons of Density Maps for Crowd Analysis Tasks—Counting, Detection, and Tracking". In: *IEEE Transactions on Circuits and Systems for Video Technology* 29.5, pp. 1408–1422.

Karamouzas, Ioannis et al. (2009). "A predictive collision avoidance model for pedestrian simulation". In: *Proceeding of the International workshop on motion in games*. Springer, pp. 41–52.

Khan, Salman et al. (2018). *A Guide to Convolutional Neural Networks for Computer Vision*. Morgan & Claypool.

Kim, Chloe Eunhyang et al. (2018). "A comparison of embedded deep learning methods for person detection". In: *arXiv preprint arXiv:1812.03451*.

Kim, Daehum et al. (Sept. 2012). "Crowd Density Estimation Using Multi-class Adaboost". In: *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*. 2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance, pp. 447–451.

Kingma, Diederik P. and Jimmy Ba (2015). "Adam: a method for stochastic optimization". In: *3rd International Conference for Learning Representations*.

Kok, V. J. and C. S. Chan (2017). "GrCS: Granular Computing-Based Crowd Segmentation". In: *IEEE Transactions on Cybernetics* 47.5, pp. 1157–1168.

Krausz, Barbara and Christian Bauckhage (2012). "Loveparade 2010: Automatic video analysis of a crowd disaster". In: *Computer Vision and Image Understanding*. Special issue on Semantic Understanding of Human Behaviors in Image Sequences 116.3, pp. 307–319.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Proceedings of the Advances in neural information processing systems*, pp. 1097–1105.

Kumagai, Shohei, Kazuhiro Hotta, and Takio Kurita (2017). "Mixture of counting cnns: Adaptive integration of cnns specialized to specific appearance for crowd counting". In: *arXiv preprint arXiv:1703.09393*.

Kuricheti, Gayatri and P Supriya (2019). "Computer Vision Based Turmeric Leaf Disease Detection and Classification: A Step to Smart Agriculture". In: *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), pp. 545–549.

Kuvaev, Alexander and Roman Khudorozhkov (2020). "An Attention-Based CNN for ECG Classification". In: *Advances in Computer Vision*. Ed. by Kohei Arai and Supriya Kapoor. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, pp. 671–677.

Lan, Wenbo et al. (2018). "Pedestrian Detection Based on YOLO Network Model". In: *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*. 2018 IEEE International Conference on Mechatronics and Automation (ICMA). ISSN: 2152-744X, pp. 1547–1551.

Laroca, Rayson et al. (2018). "A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector". In: *2018 International Joint Conference on Neural Networks (IJCNN)*. 2018 International Joint Conference on Neural Networks (IJCNN). ISSN: 2161-4407, pp. 1–10.

Lecun, Y. et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11. Conference Name: Proceedings of the IEEE, pp. 2278–2324.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *Nature* 521.7553. Number: 7553 Publisher: Nature Publishing Group, pp. 436–444.

Lempitsky, Victor and Andrew Zisserman (2010). "Learning To Count Objects in Images". In: *Advances in Neural Information Processing Systems 23*. Ed. by J. D. Lafferty et al. Curran Associates, Inc., pp. 1324–1332.

Leo, Marco et al. (2020). "Analysis of Facial Information for Healthcare Applications: A Survey on Computer Vision-Based Approaches". In: *Information* 11.3. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute, p. 128.

Li, Gang and Chunyu Li (2020). "Learning skeleton information for human action analysis using Kinect". In: *Signal Processing: Image Communication* 84, p. 115814.

Li, Guanbin and Yizhou Yu (2015). "Visual saliency based on multiscale deep features". In: *Proceedings of the IEEE conference on Computer Vision and pattern Recognition (CVPR'15)*, pp. 5455–5463.

Li, Hanhui et al. (2018). "Structured inhomogeneous density map learning for crowd counting". In: *arXiv preprint arXiv:1801.06642*.

Li, Min, Guanghua Xu, et al. (2018). "Pre-Impact Fall Detection Based on a Modified Zero Moment Point Criterion Using Data From Kinect Sensors". In: *IEEE Sensors Journal* 18.13. Conference Name: IEEE Sensors Journal, pp. 5522–5531.

Li, Min, Zhaoxiang Zhang, et al. (2008). "Estimating the number of people in crowded scenes by MID based foreground segmentation and head-shoulder detection". In: *2008 19th International Conference on Pattern Recognition*. Proceeding in the 19th International Conference on Pattern Recognition (ICPR'08). ISSN: 1051-4651, pp. 1–4.

Li, Yuhong, Xiaofan Zhang, and Deming Chen (2018). "CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes". In: *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'18), pp. 1091–1100.

Lin, Jia-Ping and Min-Te Sun (2018). "A YOLO-Based Traffic Counting System". In: *2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*. ISSN: 2376-6824, pp. 82–85.

Lin, Tsung-Yi et al. (2014). "Microsoft COCO: Common Objects in Context". In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 740–755.

Lira, Gustavo et al. (2016). "A computer-vision approach to traffic analysis over intersections". In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). ISSN: 2153-0017, pp. 47–53.

Liu, Jiang et al. (2018). "Decidenet: Counting varying density crowds through attention guided detection and density estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*, pp. 5197–5206.

Liu, L. et al. (2020). "DENet: A Universal Network for Counting Crowd with Varying Densities and Scales". In: *IEEE Transactions on Multimedia*. Conference Name: IEEE Transactions on Multimedia, pp. 1–1.

Liu, Lingbo, Zhilin Qiu, et al. (2019). "Crowd Counting With Deep Structured Scale Integration Network". In: Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR'19), pp. 1774–1783.

Liu, Lingbo, Hongjun Wang, et al. (2018). "Crowd counting using deep recurrent spatial-aware network". In: *27th International Joint Conference on Artificial Intelligence (IJCAI'18)*.

Liu, Ning et al. (2019). "ADCrowdNet: An Attention-injective Deformable Convolutional Network for Crowd Understanding". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'19)*, pp. 3225–3234.

Liu, Weizhe, Mathieu Salzmann, and Pascal Fua (2019). "Context-Aware Crowd Counting". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5099–5108.

Liu, Xialei, Joost van de Weijer, and Andrew D Bagdanov (2018). "Leveraging unlabeled data for crowd counting by learning to rank". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*, pp. 7661–7669.

Lowe, D.G. (1999). "Object recognition from local scale-invariant features". In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Proceedings of the Seventh IEEE International Conference on Computer Vision. Vol. 2, 1150–1157 vol.2.

Loy, Chen Change et al. (2013). "Crowd counting and profiling: Methodology and evaluation". In: *Modeling, simulation and visual analysis of crowds*. Springer, pp. 347–382.

Luo, Liangchen et al. (2019). "Adaptive Gradient Methods with Dynamic Bound of Learning Rate". In: *arXiv:1902.09843 [cs, stat]*. arXiv: 1902.09843.

Manfredi, Marco et al. (2014). "Detection of static groups and crowds gathered in open spaces by texture classification". In: *Pattern Recognition Letters*. Pattern Recognition and Crowd Analysis 44, pp. 39–48.

Marín, Javier et al. (2010). "Learning appearance in virtual scenarios for pedestrian detection". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 137–144.

Marsden, Mark et al. (2016). "Fully convolutional crowd counting on highly congested scenes". In: *12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*.

— (2017). "Resnetcrowd: A residual deep learning architecture for crowd counting, violent behaviour detection and crowd density level classification". In: *Proceedings of the 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS'17)*. IEEE, pp. 1–7.

McCormac, John et al. (2016). "Scenenet rgb-d: 5m photorealistic images of synthetic indoor trajectories with ground truth". In: *arXiv preprint arXiv:1612.05079*.

Müller, Andreas Christian and Sarah Guido (2018). *Introduction to machine learning with Python a guide for data scientists*. OCLC: 1107796731.

Nair, Vinod and Geoffrey E Hinton (2010). "Rectified Linear Units Improve Restricted Boltzmann Machines". In: p. 8.

Nayar, Shree K., Katsushi Ikeuchi, and Takeo Kanade (1990). "Shape from interreflections". In: *International Journal of Computer Vision* 6, pp. 173–195.

Niedenthal, Simon (2002). "Learning from the Cornell Box". In: *Leonardo* 35, pp. 249–254.

Nishani, Eralda and Betim Çiço (2017). "Computer vision approaches based on deep learning and neural networks: Deep neural networks for video analysis of human pose estimation". In: *2017 6th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1–4.

Oh, Min-hwan, Peder Olsen, and Karthikeyan Natesan Ramamurthy (2020). "Crowd Counting with Decomposed Uncertainty". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.7. Number: 07, pp. 11799–11806.

Olmschenk, Greg, Hao Tang, and Zhigang Zhu (2019). "Improving Dense Crowd Counting Convolutional Neural Networks using Inverse k-Nearest Neighbor Maps and Multiscale Upsampling". In: *arXiv preprint arXiv:1902.05379*.

Onoro-Rubio, Daniel and Roberto J López-Sastre (2016). "Towards perspective-free object counting with deep learning". In: *European Conference on Computer Vision*. Springer, pp. 615–629.

Oren, M. et al. (1997). "Pedestrian detection using wavelet templates". In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition). ISSN: 1063-6919, pp. 193–199.

ou, Vasileios and Maria Petrou (2009). "Chapter 1 Photometric Stereo: An Overview". In: *Advances in Imaging and Electron Physics*. Ed. by Peter W. Hawkes. Vol. 156. Imaging and Electron Physics. Elsevier, pp. 1–54.

Ozcan, Koray et al. (2020). "Road Weather Condition Estimation Using Fixed and Mobile Based Cameras". In: *Advances in Computer Vision*. Ed. by Kohei Arai and Supriya Kapoor. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, pp. 192–204.

Patrício, Diego Inácio and Rafael Rieder (2018). "Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review". In: *Computers and Electronics in Agriculture* 153, pp. 69–81.

Pelechano, Nuria et al. (2005). *Crowd simulation incorporating agent psychological models, roles and communication*. Tech. rep. Pennsylvania, United States Center for Human Modeling and Simulation.

Perez, Hugo et al. (2016). "Task-based crowd simulation for heterogeneous architectures". In: *Innovative Research and Applications in Next-Generation High Performance Computing*. IGI Global, pp. 194–219.

Pham, Viet-Quoc et al. (2015). "Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV'15)*, pp. 3253–3261.

Pishchulin, Leonid, Mykhaylo Andriluka, et al. (2013). "Poselet Conditioned Pictorial Structures". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 588–595.

Pishchulin, Leonid, Arjun Jain, et al. (2011). "Learning people detection models from few training samples". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*, pp. 1473–1480.

Pouyanfar, Samira et al. (2018). "A Survey on Deep Learning: Algorithms, Techniques, and Applications". In: *ACM Computing Surveys* 51.5, 92:1–92:36.

Qiming, Luo et al. (2017). "The design of intelligent crowd attention detection system based on face detection technology". In: *2017 13th IEEE International Conference on Electronic Measurement Instruments (ICEMI)*. 2017 13th IEEE International Conference on Electronic Measurement Instruments (ICEMI), pp. 310–314.

Qiu, Zhilin et al. (2019). "Crowd counting via multi-view scale aggregation networks". In: *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, pp. 1498–1503.

Quintero, Luis (2019). *Grayscale Photography of People Raising Hands · Free Stock Photo*. URL: https://www.pexels.com/photo/grayscale-photography-of-people-raising-hands-2014775/ (visited on 08/19/2020).

Ranjan, Rajeev, Vishal M Patel, and Rama Chellappa (2017). "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.1, pp. 121–135.

Ranjan, Viresh, Hieu Le, and Minh Hoai (2018). "Iterative crowd counting". In: *Proceedings of the European Conference on Computer Vision (ECCV)'18*, pp. 270–285.

Remondino, Fabio and Sabry El-Hakim (2006). "Image-based 3D modelling: a review". In: *The photogrammetric record* 21.115. Publisher: Wiley Online Library, pp. 269–291.

Ren, Shaoqing et al. (2015). "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*, pp. 91–99.

Rodin, Ivan et al. (2020). "Scene Understanding and Interaction Anticipation from First Person Vision". In: *SMARTPHIL-First Workshop on Personal Health Interfaces, in conjunction with ACM IUI*.

Russakovsky, Olga et al. (2015). "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3, pp. 211–252.

Russell, Stuart J. (1997). "Rationality and intelligence". In: *Artificial intelligence* 94.1. Publisher: Elsevier, pp. 57–77.

Ryan, David et al. (2015). "An evaluation of crowd counting methods, features and regression models". In: *Computer Vision and Image Understanding* 130, pp. 1–17.

Saleh, Sami Abdulla Mohsen, Shahrel Azmin Suandi, and Haidi Ibrahim (2015). "Recent survey on crowd density estimation and counting for visual surveillance". In: *Engineering Applications of Artificial Intelligence* 41, pp. 103–114.

Sam, Deepak Babu and R Venkatesh Babu (2018). "Top-down feedback for crowd counting convolutional neural network". In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

Sam, Deepak Babu, Shiv Surya, and R Venkatesh Babu (2017). "Switching convolutional neural network for crowd counting". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. IEEE, pp. 4031–4039.

Santhosh, K. K., D. P. Dogra, and P. P. Roy (2020). "Anomaly Detection in Road Traffic Using Visual Surveillance: A Survey". In: *ACM Computing Surveys* 53.6, 119:1–119:26.

Sermanet, P., S. Chintala, and Y. LeCun (2012). "Convolutional neural networks applied to house numbers digit classification". In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR'12)*. Proceedings of the 21st International Conference on Pattern Recognition (ICPR'12). ISSN: 1051-4651, pp. 3288–3291.

Shang, Chong, Haizhou Ai, and Bo Bai (2016). "End-to-end crowd counting via joint learning local and global count". In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 1215–1219.

Sharma, Sagar (2017). "Activation functions in neural networks". In: *Towards Data Science* 6.

Sheng, Biyun et al. (2016). "Crowd counting via weighted VLAD on a dense attribute feature map". In: *IEEE Transactions on Circuits and Systems for Video Technology* 28.8, pp. 1788–1797.

Shi, Miaojing et al. (2018). "Perspective-aware CNN for crowd counting". PhD thesis. Inria Rennes-Bretagne Atlantique.

Shi, Xiaohua, Kaicheng Tang, and Hongtao Lu (2020). "Smart library book sorting application with intelligence computer vision technology". In: *Library Hi Tech* ahead-of-print (ahead-of-print).

Shi, Zenglin et al. (2018). "Crowd counting with deep negative correlation learning". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR'18)*, pp. 5382–5390.

Silverman, Barry G et al. (2005). "Crowd simulation incorporating agent psychological models, roles and communication". In:

Sim, Chern-Horng, Ekambaram Rajmadhan, and Surendra Ranganath (2008). "A Two-Step Approach for Detecting Individuals within Dense Crowds". In: *Articulated Motion and Deformable Objects*. Ed. by Francisco J. Perales and Robert B. Fisher. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 166–174.

Simonyan, Karen and Andrew Zisserman (2015). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations (ICLR'15)*.

Sindagi, Vishwanath and Vishal M. Patel (2017a). "A Survey of Recent Advances in CNN-based Single Image Crowd Counting and Density Estimation". In: *Pattern Recognition Letters* 107, pp. 3–16.

— (2017b). "CNN-Based cascaded multi-task learning of high-level prior and density estimation for crowd counting". In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6.

Sindagi, Vishwanath A and Vishal M Patel (2017a). "Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting". In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, pp. 1–6.

— (2017b). "Generating high-quality crowd density maps using contextual pyramid cnns". In: *Proceedings of the IEEE International Conference on Computer Vision)*, pp. 1861–1870.
— (2018). "A survey of recent advances in CNN-based single image crowd counting and density estimation". In: *Pattern Recognition Letters*. Video Surveillance-oriented Biometrics 107, pp. 3–16.
Solera, Francesco, Simone Calderara, and Rita Cucchiara (2013). "Structured learning for detection of social groups in crowd". In: *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*. 2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 7–12.
Sugiyama, Masashi (2016). *Introduction to statistical machine learning*. OCLC: 1175927586.
Sun, Hao, Cheng Wang, and Boliang Wang (2011). "Night Vision Pedestrian Detection Using a Forward-Looking Infrared Camera". In: *2011 International Workshop on Multi-Platform/Multi-Sensor Remote Sensing and Mapping*. 2011 International Workshop on Multi-Platform/Multi-Sensor Remote Sensing and Mapping, pp. 1–4.
Sun, J. et al. (2007). "Object surface recovery using a multi-light photometric stereo technique for non-Lambertian surfaces subject to shadows and specularities". In: *Image and Vision Computing* 25.7, pp. 1050–1057.
Sutskever, Ilya et al. (2013). "On the importance of initialization and momentum in deep learning". In: *International conference on machine learning*, pp. 1139–1147.
Szegedy, Christian, Sergey Ioffe, et al. (2017). "Inception-v4, inception-resnet and the impact of residual connections on learning". In: *Thirty-First AAAI Conference on Artificial Intelligence*.
Szegedy, Christian, Wei Liu, et al. (2015). "Going deeper with convolutions". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR'15)*, pp. 1–9.
Takumi, Karasawa et al. (2017). "Multispectral Object Detection for Autonomous Vehicles". In: *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*. Thematic Workshops '17. New York, NY, USA: Association for Computing Machinery, pp. 35–43.
Tan, Ping, Stephen Lin, and Long Quan (2008). "Subpixel Photometric Stereo". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'08)* 30, pp. 1460–1471.
Tankus, Ariel and Nahum Kiryati (2005). "Photometric stereo under perspective projection". In: *10th IEEE International Conference on Computer Vision (ICCV'05)* 1, 611–616 Vol. 1.
Thévenaz, Philippe, Thierry Blu, and Michael Unser (2000). "Image interpolation and resampling". In: *Handbook of medical imaging, processing and analysis* 1.1. Publisher: Citeseer, pp. 393–420.
Tian, Yukun et al. (2019). "Padnet: Pan-density crowd counting". In: *IEEE Transactions on Image Processing* 29. Publisher: IEEE, pp. 2714–2727.
Tombe, Ronald (2020). "Computer Vision for Smart Farming and Sustainable Agriculture". In: *2020 IST-Africa Conference (IST-Africa)*. 2020 IST-Africa Conference (IST-Africa). ISSN: 2576-8581, pp. 1–8.
Valloli, Varun Kannadi and Kinal Mehta (2019). "W-Net: Reinforced U-Net for Density Map Estimation". In: *arXiv preprint arXiv:1903.11249*.

Varior, Rahul Rama et al. (2019). "Scale-Aware Attention Network for Crowd Counting". In: *arXiv preprint arXiv:1901.06026*.

Varol, Gül et al. (2017). "Learning from Synthetic Humans". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, pp. 4627–4635.

Vinyals, Oriol et al. (2015). "Show and tell: A neural image caption generator". In: *Proceedings of the IEEE conference on Computer Vision and pattern Recognition (CVPR'15)*, pp. 3156–3164.

Viola, P. and M. Jones (2001). "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'01*. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR'01. Vol. 1. Kauai, HI, USA: IEEE Comput. Soc, pp. I–511–I–518.

Viola, Paul and Michael J Jones (2004). "Robust real-time face detection". In: *International journal of computer vision* 57.2, pp. 137–154.

Walach, Elad and Lior Wolf (2016). "Learning to count with cnn boosting". In: *European Conference on Computer Vision (ECCV'16)*. Springer, pp. 660–676.

Wang, Chuan et al. (2015). "Deep people counting in extremely dense crowds". In: *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, pp. 1299–1302.

Wang, Haohan and Bhiksha Raj (2017). "On the origin of deep learning". In: *arXiv preprint arXiv:1702.07800*.

Wang, Jinghong et al. (2013). "Risk of Large-Scale Evacuation Based on the Effectiveness of Rescue Strategies Under Different Crowd Densities". In: *Risk Analysis* 33.8, pp. 1553–1563.

Wang, Liming et al. (2007). "Object Detection Combining Recognition and Segmentation". In: *Proceedings of the 8th Asian Conference on Computer Vision (ACCV'17)*. Ed. by Yasushi Yagi et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 189–199.

Wang, Qi et al. (2019a). "Learning From Synthetic Data for Crowd Counting in the Wild". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19), pp. 8198–8207.

— (2019b). "Learning From Synthetic Data for Crowd Counting in the Wild". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19), pp. 8198–8207.

Wang, Yi and Yuexian Zou (2016). "Fast visual object counting via example-based density estimation". In: *Proceedings of the IEEE International Conference on Image Processing (ICIP'16)*. IEEE, pp. 3653–3657.

Wang, Yongzhi et al. (2010). "Pedestrian Detection Using Coarse-to-Fine Method with Haar-Like and Shapelet Features". In: *2010 International Conference on Multimedia Technology*. 2010 International Conference on Multimedia Technology, pp. 1–4.

Wang, Ze et al. (2018). "In defense of single-column networks for crowd counting". In: *arXiv preprint arXiv:1808.06133*.

Wang, Zhou et al. (2004). "Image quality assessment: from error visibility to structural similarity". In: *IEEE transactions on image processing* 13.4, pp. 600–612.

Wei, Yunchao et al. (2016). "Stc: A simple to complex framework for weakly-supervised semantic segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.11, pp. 2314–2320.

Weng, Rongxiang et al. (2020). "Acquiring Knowledge from Pre-Trained Model to Neural Machine Translation". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.5. Number: 05, pp. 9266–9273.

Wilson, Ashia C. et al. (2018). "The Marginal Value of Adaptive Gradient Methods in Machine Learning". In: *arXiv:1705.08292 [cs, stat]*. arXiv: 1705.08292.

Witschel, Tim and Christian Wressnegger (2020). "Aim low, shoot high: evading aimbot detectors by mimicking user behavior". In: *Proceedings of the 13th European workshop on Systems Security*. EuroSec '20. New York, NY, USA: Association for Computing Machinery, pp. 19–24.

Wu, Xingjiao et al. (2019). "Adaptive Scenario Discovery for Crowd Counting". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'19)*. IEEE, pp. 2382–2386.

Xiaohua, Li, Shen Lansun, and Li Huanqin (2006). "Estimation of Crowd Density Based on Wavelet and Support Vector Machine". In: *Transactions of the Institute of Measurement and Control* 28.3, pp. 299–308.

Xie, Shaoci, Xiaohong Zhang, and Jing Cai (2019). "Video crowd detection and abnormal behavior model detection based on machine learning method". In: *Neural Computing and Applications* 31.1, pp. 175–184.

Xiong, Feng, Xingjian Shi, and Dit-Yan Yeung (2017). "Spatiotemporal modeling for crowd counting in videos". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV'17)*, pp. 5151–5159.

Xu, Dan et al. (2017). "Learning deep structured multi-scale features using attention-gated crfs for contour prediction". In: *Advances in Neural Information Processing Systems*, pp. 3961–3970.

Yang, Ying and Danyang Li (2017). "Robust player detection and tracking in broadcast soccer video based on enhanced particle filter". In: *Journal of Visual Communication and Image Representation* 46, pp. 81–94.

Yang, Zhangsihao, Haoliang Jiang, and Lan Zou (2020). "3D Conceptual Design Using Deep Learning". In: *Advances in Computer Vision*. Ed. by Kohei Arai and Supriya Kapoor. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, pp. 16–26.

Yi, Shuai et al. (2016). "L_0 Regularized Stationary-Time Estimation for Crowd Analysis". In: *IEEE transactions on pattern analysis and machine intelligence (TPAMI)* 39.5, pp. 981–994.

Yin, Chuanlong et al. (2017). "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks". In: *IEEE Access* 5. Conference Name: IEEE Access, pp. 21954–21961.

Yogameena, B. and C. Nagananthini (2017). "Computer vision based crowd disaster avoidance system: A survey". In: *International Journal of Disaster Risk Reduction* 22, pp. 95–129.

Yu, Fisher and Vladlen Koltun (2016). "Multi-scale context aggregation by dilated convolutions". In: *4th International Conference on Learning Representations (ICLR 2016)*.

Yuan, Y., J. Fang, and Q. Wang (2015). "Online Anomaly Detection in Crowd Scenes via Structure Analysis". In: *IEEE Transactions on Cybernetics* 45.3, pp. 548–561.

Zeng, Lingke et al. (2017). "Multi-scale convolutional neural networks for crowd counting". In: *Proceedings of the IEEE International Conference on Image Processing (ICIP'17)*. IEEE, pp. 465–469.

Zhan, Beibei et al. (2008). "Crowd analysis: a survey". In: *Machine Vision and Applications* 19.5, pp. 345–357.

Zhang, Cong, Kai Kang, et al. (2016). "Data-Driven Crowd Understanding: A Baseline for a Large-Scale Crowd Dataset". In: *IEEE Transactions on Multimedia* 18.6. Conference Name: IEEE Transactions on Multimedia, pp. 1048–1061.

Zhang, Cong, Hongsheng Li, et al. (2015). "Cross-scene crowd counting via deep convolutional neural networks". In: *Proceedings of the IEEE conference on Computer Vision and pattern Recognition, (CVPR'16)*, pp. 833–841.

Zhang, Lu, Ju Dai, et al. (2018). "A bi-directional message passing model for salient object detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1741–1750.

Zhang, Lu, Miaojing Shi, and Qiaobo Chen (2018). "Crowd counting via scale-adaptive convolutional neural network". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, pp. 1113–1121.

Zhang, Ruiheng et al. (2020). "Multi-camera multi-player tracking with deep player identification in sports video". In: *Pattern Recognition* 102, p. 107260.

Zhang, Shanshan, Jian Yang, and Bernt Schiele (2018). "Occluded Pedestrian Detection Through Guided Attention in CNNs". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR'18), pp. 6995–7003.

Zhang, Yingying et al. (2016). "Single-image crowd counting via multi-column convolutional neural network". In: *Proceedings of the IEEE conference on Computer Vision and pattern Recognition (CVPR'16)*, pp. 589–597.

Zhang, Youmei et al. (2019). "Attention to Head Locations for Crowd Counting". In: *Image and Graphics*. Ed. by Yao Zhao et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 727–737.

Zhou, Y. et al. (2020). "Adversarial Learning for Multiscale Crowd Counting Under Complex Scenes". In: *IEEE Transactions on Cybernetics*, pp. 1–10.

Zhu, Meilu et al. (2019). "Robust Facial Landmark Detection via Occlusion-Adaptive Deep Networks". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19), pp. 3486–3496.

Zhu, Zhe et al. (2016). "Traffic-Sign Detection and Classification in the Wild". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16), pp. 2110–2118.

Zou, Zhikang et al. (2019). "Enhanced 3D convolutional networks for crowd counting". In: *30th British Machine Vision Conference BMVC'19*.