

© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

FOURIER TRANSFORMATION AUTOENCODERS FOR ANOMALY DETECTION

Demetris Lappas, Vasileios Argyriou, Dimitrios Makris

School of Computer Science and Mathematics
Kingston University, London, UK

ABSTRACT

Anomaly detection is a challenging problem, mainly due to the lack of a sufficient set of abnormal samples that represents every possible anomaly. Therefore unsupervised methods are employed to model normality and anomaly is detected as an outlier to such models. This paper introduces Fourier Transforms into AutoEncoders to demonstrate how the inclusion of a frequency domain presents less noisy features for a deep learning network to detect anomalies. Comparing our results to the state of the art on a variety of datasets, we show how the proposed method can provide competitive results.

Index Terms— Anomaly detection, Auto-encoder, Fourier Transform

1. INTRODUCTION

Anomaly detection is the practice of detecting rare patterns in data, which deviate beyond a particular distribution of expected behaviour. Anomalies by definition are rare, automated mechanisms that are able to detect anomalies would be valuable as they have the potential to significantly reduce human operators' time and therefore minimise errors and costs.

Classic machine learning approaches tackle the problem by measuring the distance between data points with respect to the area occupied by the majority of normal data; examples of such approaches are Isolation Forests [1], Local Outlier Factor [2, 3] and Gaussian Mixture Models [4].

Fourier Transform, which transforms any sequence from time domain to a frequency domain, has been used before for anomaly detection in classic machine learning approaches [5, 6]. Therefore, if a particular data instance appears to have frequent content that is significant different from the expected frequencies, it is likely to be an anomaly [7, 8, 9].

Deep learning methods appear to have overtaken classic machine learning in recent years. AutoEncoders, which allow for data such as images and videos to be reduced to a low dimensional latent space, have attracted the interest of researchers in anomaly detection [10, 11, 12]. This process seems to remove most redundant noise before data is reconstructed. Others have attempted to expand traditional deep learning by performing classic machine learning on the latent space [13].

An evolved version of AutoEncoders is the Variational AutoEncoders [14, 15], which randomly sample from a normal distribution on the latent space. This allows for fewer gaps in the latent space and trains the network to reconstruct areas which may not be in the training data.

Approaches using semi-supervised learning have also been explored [16, 17, 18] which incorporate Generative Adversarial Networks (GANs), consisting of a Generator and a Discriminator. EGBAD [17] and AnoGAN [18] use GANs to train a Discriminator to distinguish the difference between real and fake samples, under the logical assumption that anomalous samples would resemble fake samples. GANomaly [16] utilizes GANs differently to AnoGAN and EGBAD; GANomaly uses GANs to better train an AutoEncoder by checking whether the Discriminator can distinguish the differences between the input and output. This method seems to be effective as it outperforms AnoGAN and EGBAD [19].

Our contribution bridges the gap between using Fourier Transforms to detect anomalous features and modern deep learning methods. Anomaly detection in images is not without noisy data, often RGB features may not be enough to train a network to distinguish features, which differentiate normal samples from anomalous ones. However, by incorporating Fourier Transform features into our model and applying a combined loss function of Kullback–Leibler (KL) divergence and Binary Cross-entropy, we show that Fourier Transform features improve the ability of a deep network to detect anomalies.

The rest of this paper is organized as following: section 2 describes the proposed deep neural network model, section 3 presents the evaluation results, analysis and comparison and finally section 4 concludes the findings of this research.

2. FOURIER AUTO-ENCODER FOR ANOMALY DETECTION

An anomaly detection dataset can be abstractly described by a set D containing normal and anomalous samples X and \hat{X} respectfully, such that $X \subset D$, $\hat{X} \subset D$, $X \cap \hat{X} = \emptyset$, $X \cup \hat{X} = D$ and $|X| \gg |\hat{X}|$.

2.1. Related concepts and methodologies

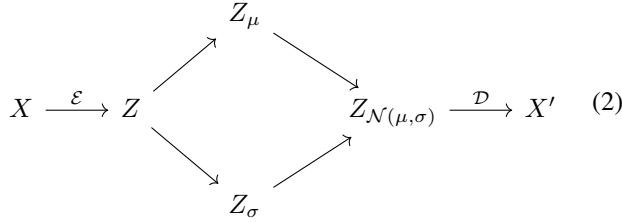
One of the modern methods for detecting anomalies is AutoEncoders (AEs). An AE can be conceptually composed of an Encoder followed by a Decoder, $\mathcal{E} : X \rightarrow Z$ and $\mathcal{D} : Z \rightarrow X'$ respectfully.

$$X \xrightarrow{\mathcal{E}} Z \xrightarrow{\mathcal{D}} X' \quad (1)$$

The Encoder would map any $x \in X$ to $\mathcal{E}(x) \in Z$, where $X \subseteq \mathbb{R}^m$, $Z \subseteq \mathbb{R}^n$ and $m > n$. The dimensionally reduced space Z is formally known as the latent space. The Decoder then maps $\mathcal{E}(x) \in Z$ to $\mathcal{D}(\mathcal{E}(x)) \in X'$, where $X' \subseteq \mathbb{R}^m$ is similar to X . If $|x - \mathcal{D}(\mathcal{E}(x))|$ exceeds a certain threshold the sample is considered anomalous.

The presumption is that because AEs are trained on normal data, they will be unable to reconstruct anomalous data, therefore the AE output will differ significantly to the AE input.

An extension of an AE is the Variational AutoEncoder (VAE). VAEs are similar to AEs except they randomly sample from a normal distribution on the latent space, often within one standard deviation of $\mathcal{E}(x)$ so that even small differences within samples may be mapped.



For many cases, it has been shown that VAEs perform better than AEs.

A Fourier Transform is a transformation which maps any sequence on the time domain in \mathbb{R}^n to a frequency domain on the complex plane, \mathbb{C}^n with $\mathbb{C}^n = \mathbb{R}^{2n}$. Furthermore, a Fourier Transform is an isomorphism, let's initially define it as $\mathcal{F} : \mathbb{R}^n \rightarrow \mathbb{C}^n$. An isomorphism is an injective, surjective and homomorphic map.

- Injective - For every unique value $x \in \mathbb{R}^n$ there exists a unique value $\mathcal{F}(x) \in \mathbb{C}^n$, which implies \mathcal{F} is invertable.
- Surjective - \mathcal{F} maps the space \mathbb{R}^n to the entire space \mathbb{C}^n .
- Homomorphic - For any two abstract binary operations (say \circ and $*$) and any $a, b \in \mathbb{R}^n$,
 $\mathcal{F}(a \circ b) = \mathcal{F}(a) * \mathcal{F}(b)$.

This means calculations can be made on a Fourier Transform $\mathcal{F}(X)$ as though they were being made on X , while still retaining all of the information. Fourier Transform offers an enhanced representation of abnormal points and therefore it can improve the performance of anomaly detection approaches.

2.2. Fourier Models for Anomaly Detection

The proposed method takes advantage of Fourier Transform's representation to identify anomalies by combine them with AEs and VAEs. Firstly, a function \mathcal{F} is defined which maps any dataset X to its Fourier Transform in the complex plane, $\mathcal{F} : \mathbb{R}^n \rightarrow \mathbb{C}^n$. Since Fourier Transform is isomorphic, the values of $\mathcal{F}(X)$ could be used to perfectly represent the values of X . As a result a conceptually valid AE is shown in equation (3).

$$X \xrightarrow{\mathcal{F}} X_{\mathcal{F}} \xrightarrow{\mathcal{E}} Z \xrightarrow{\mathcal{D}} X'_{\mathcal{F}} \xrightarrow{\mathcal{F}^{-1}} X' \quad (3)$$

However, \mathcal{E} (i.e. Encoder) and \mathcal{D} (i.e. Decoder) for the model in (3) would need to have complex weights. Furthermore, we also recall that $\mathbb{C}^n = \mathbb{R}^{2n}$, which means $\mathcal{F}(X)$ can be split into two for each $\mathbb{R}^n \subset \mathbb{R}^{2n} = \mathbb{C}^n$. Let us define the functions $\mathcal{F}_{\mathbb{R}}$ and $\mathcal{F}_{\mathbb{I}}$ which map any dataset X to their Fourier Transform Real Value and Fourier Transform Imaginary Value datasets respectfully as.

$$\begin{aligned} \mathcal{F}_{\mathbb{R}} : X &\rightarrow X_{\mathbb{R}} \\ \mathcal{F}_{\mathbb{I}} : X &\rightarrow X_{\mathbb{I}} \end{aligned} \quad (4)$$

The proposed method builds three Encoders \mathcal{E} , $\mathcal{E}_{\mathbb{R}}$, $\mathcal{E}_{\mathbb{I}}$ which map X , $X_{\mathbb{R}}$, $X_{\mathbb{I}}$ to their latent spaces Z , $Z_{\mathbb{R}}$, $Z_{\mathbb{I}}$ respectfully. Each Encoder represents a sequence of Convolution layers, Activation Function layers and Batch Normalization layers - that are denoted as a Sequence of Neural Network Layers (SNNL).

$$\begin{aligned} \mathcal{E} : X &\rightarrow Z \\ \mathcal{E}_{\mathbb{R}} : X_{\mathbb{R}} &\rightarrow Z_{\mathbb{R}} \\ \mathcal{E}_{\mathbb{I}} : X_{\mathbb{I}} &\rightarrow Z_{\mathbb{I}} \end{aligned} \quad (5)$$

We define $K_{[Z, Z_{\mathbb{R}}, Z_{\mathbb{I}}]}$ to be the concatenation of Z , $Z_{\mathbb{R}}$, $Z_{\mathbb{I}}$ as our combined latent space. The intuition behind this method is that the latent space $K_{[Z, Z_{\mathbb{R}}, Z_{\mathbb{I}}]}$ should have features of lower dimensionality from both the original data X and its Fourier Transform $\mathcal{F}(X)$. The latent space $K_{[Z, Z_{\mathbb{R}}, Z_{\mathbb{I}}]}$ is then decoded by \mathcal{D} to a space Y , where \mathcal{D} also represents a SNNL.

$$\mathcal{D} : K_{[Z, Z_{\mathbb{R}}, Z_{\mathbb{I}}]} \rightarrow Y \quad (6)$$

Once $K_{[Z, Z_{\mathbb{R}}, Z_{\mathbb{I}}]}$ is decoded, Y is split by two maps, one that represents real values $Y_{\mathbb{R}}$ and the other the imaginary ones $Y_{\mathbb{I}}$. Both maps in equation (7) are a SNNL.

$$\begin{aligned} Y &\rightarrow Y_{\mathbb{R}} \\ Y &\rightarrow Y_{\mathbb{I}} \end{aligned} \quad (7)$$

Since for both of them we have $Y_{\mathbb{R}} \subset \mathbb{R}^n$ and $Y_{\mathbb{I}} \subset \mathbb{R}^n$, they are combined to represent the real and imaginary parts of $Y_{\mathbb{C}} \subset \mathbb{C}^n$, respectively.

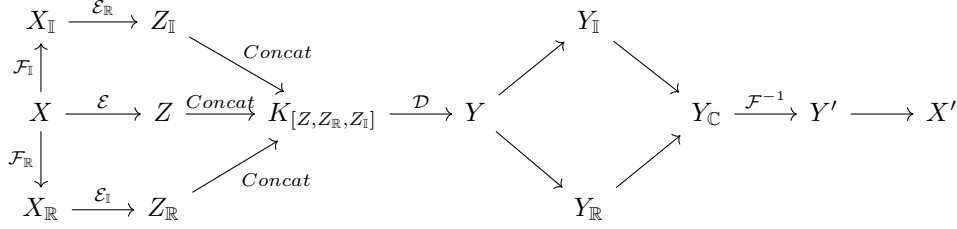


Fig. 1. FAE Complete Model

3. RESULTS

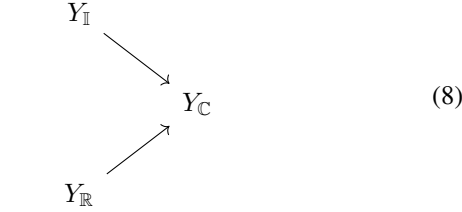
3.1. Datasets

We tested FAE and FVAE on 3 different image datasets - MNIST [20], CIFAR10 [21] and Concrete Cracks [22]. For MNIST and CIFAR10 we used a one-versus-the-rest approach [19]; meaning one class would be considered normal and the rest would be considered anomalous. The Concrete Cracks dataset is evenly split between two classes. The MNIST dataset consists of 60,000 28x28 grayscale hand written training images and 10,000 test images. Both the training and testing dataset have evenly distributed classes between 0 to 9. The CIFAR10 dataset consists of 50,000 32x32 RGB training images and 10,000 test images. Both the training and testing datasets have 10 evenly distributed classes of the following categories; plane, car, bird, cat, deer, frog, horse, ship, truck, dog. The Concrete Cracks dataset consists of 40,000 227x227 RGB images of concrete. Half of the dataset consists of concrete images without cracks (normal class) and the other half consists of concrete images with cracks (anomalous class).

3.2. Evaluation

Let us define \mathcal{A} , a set of models to be trained. The set \mathcal{A} will consist of a FAE, a FVAE, a Vanilla AE and a Vanilla VAE. Both FVAE and the Vanilla VAE used a combined loss function of binary cross-entropy and KL divergence, while the FAE Vanilla AE only used binary cross-entropy.

Each model in \mathcal{A} was trained on each class in MNIST and CIFAR10, where the training dataset consisted of only a single class (considered to be the normal samples). Therefore, a total of 40 models were trained on the MNIST and 40 on the CIFAR10 datasets. Each of the 80 models scored an Area Under Curve (AUC) value on their respective test data. The results are shown in Figures 4 and 5, comparing the Fourier learning [19] and displaying a mean AUC for each one. In Figures 4 and 5 all the results are summarised and the best cases are highlighted. On the MNIST dataset the results of the Fourier Models tended to outperform or were relatively close to GANomaly's performance. Something similar can be observed for the CIFAR10 dataset with the exception of

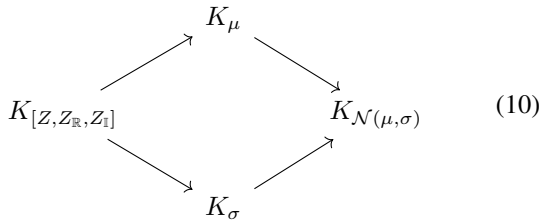


$$(8)$$

An Inverse Fourier Transform is then applied to the space $Y_C \subset \mathbb{C}^n$ to attain $Y' \subset \mathbb{R}^n$. Lastly we add a final mapping from Y' to X' which is also a SNNL, where X' is our output and should be relatively similar to our input X . A full diagram of the FAE methodology can be found under Figure 1.

$$Y' \rightarrow X' \quad (9)$$

Additionally to the above mentioned FAE we also proposed a Fourier enabled Variational AutoEncoder. The FVAE is similar to the FAE, except it samples from the latent space following the architecture of a classic VAE, before being passed to the decoder, as it is shown in equation (10). The complete FVAE is shown in Figure 2.



$$(10)$$

2.3. Architecture of the Proposed Model

Each Encoder (\mathcal{E} , \mathcal{E}_R , \mathcal{E}_I) is made of 10 layers consisting of *Conv2D*, *Max-Pooling* and *Batch Normalization Layers*. The Decoder (\mathcal{D}) is made of 11 layers consisting of *Conv2D Transpose*, *Up-Sampling2D* and *Batch Normalization Layers*. To build the Fourier Transform Layers we used *Lambda Layers* combined with the inverse and forward Fourier Transforms. Each model was trained for 100 epochs, with a batch size of 200 using an Adam Optimizer. The loss function for the Fourier Models were the combined sum of Binary Cross-entropy and KL Divergence.

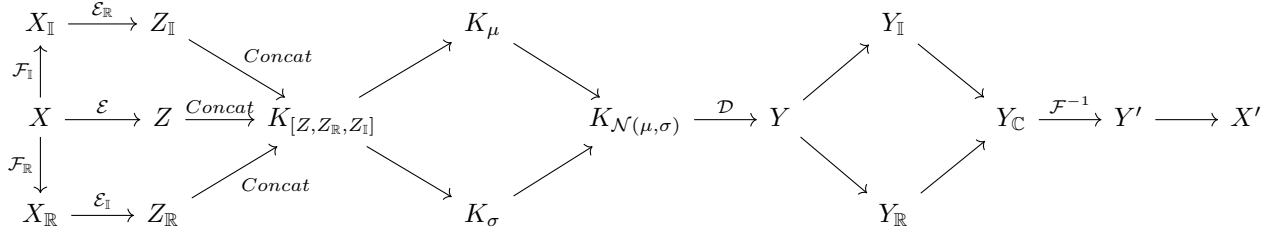


Fig. 2. FVAE Complete Model

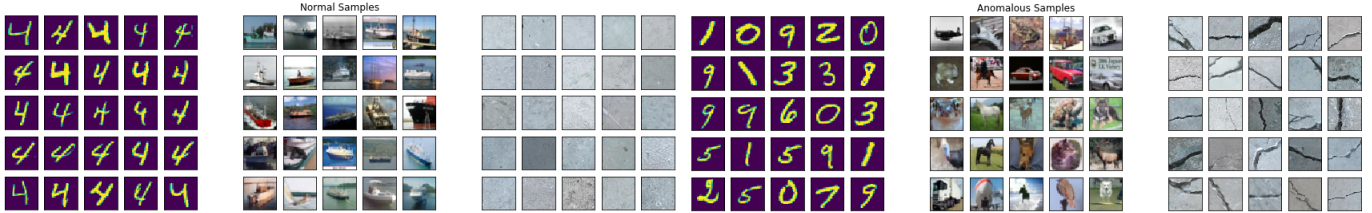


Fig. 3. Samples Split between normal (left) and abnormal (right)

Model	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9	mean
GANomaly	0.881	0.675	0.953	0.801	0.827	0.864	0.849	0.682	0.856	0.558	0.795
AnoGAN	0.623	0.31	0.521	0.458	0.442	0.431	0.492	0.401	0.392	0.368	0.444
EGBAD	0.783	0.294	0.523	0.506	0.453	0.436	0.593	0.398	0.523	0.358	0.487
Vanilla VAE	0.652	0.968	0.571	0.606	0.715	0.648	0.739	0.77	0.602	0.683	0.695
Vanilla AE	0.647	0.987	0.905	0.677	0.913	0.611	0.71	0.83	0.72	0.76	0.776
Fourier VAE	0.951	0.996	0.814	0.609	0.858	0.8	0.917	0.921	0.756	0.716	0.834
Fourier AE	0.895	0.998	0.576	0.693	0.775	0.502	0.723	0.899	0.592	0.871	0.752

Fig. 4. MNIST Results

Model	plane	car	bird	cat	deer	frog	horse	ship	truck	dog	mean	Cracks
GANomaly	0.633	0.631	0.51	0.585	0.593	0.683	0.605	0.616	0.617	0.628	0.61	0.86
AnoGAN	0.516	0.492	0.411	0.399	0.335	0.321	0.399	0.567	0.511	0.393	0.43	na
EGBAD	0.577	0.514	0.383	0.448	0.374	0.353	0.526	0.413	0.555	0.481	0.46	na
Vanilla VAE	0.651	0.436	0.529	0.476	0.484	0.47	0.476	0.596	0.499	0.508	0.51	0.79
Vanilla AE	0.713	0.407	0.679	0.566	0.735	0.638	0.494	0.702	0.41	0.549	0.59	0.99
Fourier VAE	0.678	0.443	0.52	0.5	0.481	0.477	0.485	0.638	0.51	0.51	0.52	0.85
Fourier AE	0.667	0.428	0.656	0.608	0.724	0.661	0.55	0.744	0.428	0.61	0.61	0.99

Fig. 5. CIFAR10 and Concrete Cracks Results

‘cars’ and ‘trucks’, which are quite similar and their Fourier Transform may not enhance distinct differences. Similar results were obtained for the Concrete Cracks dataset and figure 5 shows the results of the proposed FAE models and the state of the art GANomaly [19] method.

4. CONCLUSION

In this work we proposed a novel Anomaly Detection approach utilizing Fourier Transforms in AutoEncoders and

Variational Autoencoders, allowing the latent space to extract more relevant feature representations. We evaluated the suggested methodology on three datasets (MNIST, CIFAR10, Cracks) and the proposed Fourier-based models achieved similar or better results than the current state of the art.

5. REFERENCES

- [1] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou, “Isolation Forest ICDM08,” *Icdm*.

- [2] Malak Alshwabkeh, Byunghyun Jang, and David Kaeli, "Accelerating the local outlier factor algorithm on a GPU for intrusion detection systems," *International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS*, pp. 104–110, 2010.
- [3] Lin Xu, Yi Ren Yeh, Yuh Jye Lee, and Jing Li, "A hierarchical framework using approximated local outlier factor for efficient anomaly detection," *Procedia Computer Science*, vol. 19, pp. 1174–1181, 2013.
- [4] Dat Tran, W Ma, and Dharmendra Sharma, "Network Anomaly Detection using Fuzzy Gaussian Mixture Models," *International Journal of Future Generation ...*, pp. 37–42, 2006.
- [5] Yinghuan Shi, Yang Gao, and Ruili Wang, "Real-time abnormal event detection in complicated scenes," *Proceedings - International Conference on Pattern Recognition*, , no. i, pp. 3653–3656, 2010.
- [6] Lingyun Yang, Chuang Liu, Jennifer M. Schopf, and Ian Foster, "Anomaly detection and diagnosis in grid environments," *Proceedings of the 2007 ACM/IEEE Conference on Supercomputing, SC'07*, , no. c, 2007.
- [7] Michael Wrinch, Tarek H.M. El-Fouly, and Steven Wong, "Anomaly detection of building systems using energy demand frequency domain analysis," *IEEE Power and Energy Society General Meeting*, vol. 2012, 2012.
- [8] Rafael Ramos Regis Barbosa, Ramin Sadre, and Aiko Pras, "Towards periodicity based anomaly detection in SCADA networks," *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 2012.
- [9] Wei Lu and Ali A. Ghorbani, "Network anomaly detection based on wavelet analysis," *Eurasip Journal on Advances in Signal Processing*, vol. 2009, 2009.
- [10] Yong Shean Chong and Yong Haur Tay, "Abnormal event detection in videos using spatiotemporal autoencoder," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10262 LNCS, pp. 189–196, 2017.
- [11] Mayu Sakurada and Takehisa Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," *ACM International Conference Proceeding Series*, vol. 02-December, pp. 4–11, 2014.
- [12] Chong Zhou and Randy C. Paffenroth, "Anomaly detection with robust deep autoencoders," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. Part F1296, pp. 665–674, 2017.
- [13] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen, "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, pp. 1–19, 2018.
- [14] Yuchen Lu and Peng Xu, "Anomaly Detection for Skin Disease Images Using Variational Autoencoder," 2018, arxiv: 1807.01349.
- [15] L. N. Vostrova, T. A. Voronina, T. L. Karaseva, S. A. Gernega, E. I. Ivanov, A. M. Kirichenko, and M. Yu Totrova, "Synthesis and anticonvulsant activity of benzimidazo[1, 2-c]quinazolines," *Pharmaceutical Chemistry Journal*, vol. 20, no. 6, pp. 404–406, 1986.
- [16] Samet Akcay, Amir Atapour-Abarghouei, and Toby P. Breckon, "GANomaly: Semi-supervised Anomaly Detection via Adversarial Training," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11363 LNCS, pp. 622–637, 2019.
- [17] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar, "Efficient GAN-Based Anomaly Detection," 2018, arxiv: 1802.06222.
- [18] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10265 LNCS, pp. 146–147, 2017.
- [19] Manpreet Singh Minhas and John Zelek, "Anomaly Detection in Images," 2019, arxiv: 1905.13147.
- [20] Y. Bengio Y. LeCun, L. Bottou and P. Haffner, "Gradient-based learning applied to document recognition.," 1998, Proceedings of the IEEE, 86(11):2278-2324.
- [21] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton, "Cifar-10 (canadian institute for advanced research)," <http://www.cs.toronto.edu/~kriz/cifar.html>.
- [22] Yimin Daniel Zhang Lei Zhang, Fan Yang and L. Yang F. Zhang Y. D. Zhu Y. J. Y. J. Z., Zhang, "Road crack detection using deep convolutional neural network.," 2016, In 2016 IEEE International Conference on Image Processing (ICIP).