DF-GANeXt: Implementation of ConvNeXt Models for GAN Deepfake Detection

Axel Pribadi
School of Computing and Technology
Asia Pacific University of Technology and Innovation
Kuala Lumpur, Malaysia
tp061236@mail.apu.edu.my

***Abstract*—Deepfakes which are digitally manipulated synthetic videos or images, have increased in popularity and usage in the past half decade. Developments in Generative Adversarial Networks (GANs) have significantly improved to a point where deepfakes are synonymous to real images. When maliciously used, deepfakes can result in identity theft or harm on the internet. To combat this issue, the author proposes the use of a ConvNeXt model which is a ResNet model enhanced by adding Transformer-like features to it while preserving the convolutional neural network (CNN) architecture has been implemented. Additionally, an autoencoder model for anomaly detection has been employed for comparison purposes. Preliminary results demonstrate that the ConvNeXt-T model significantly outperformed the autoencoder achieving an astounding 99.50% accuracy in distinguishing GAN-generated deepfake images from the Flickr Faces High-Quality (FFHQ) dataset. By applying Gradient-weighted Class Activation Mapping (Grad-CAM) on the images, it was discovered that the facial regions are key features in the classification. Notably, we observed that fake images tended to have more widespread and less intense class activation maps.**

Keywords—ConvNeXt, convolutional neural network, deepfake, deepfake detection, Grad-CAM

I. INTRODUCTION

Computing technologies of the 21st century have played a huge role in what creative works people are able to create and post on the internet. Deepfakes, stemming from the words "deep learning" and "fake", are forms of hyper realistic synthetic media which have been digitally manipulated. These media usually refer to videos, images and even audios. Ever since deepfakes boomed in Reddit in 2017 [1], they have grown in popularity and the technologies behind them have improved as well. Especially in the realm of deepfake images, the development of generative adversarial networks (GANs) [2] have allowed deepfakes to be created without the need of a source face. The fascinating part is that GAN-generated images have reached a point where they are eerily identical to real faces which makes them tough to distinguish.

With the ever-increasing accessibility to manipulation tools on the internet, deepfakes can essentially be created by anyone without any necessary technical skills required. A prime example would be "this-person-does-not-exist.com" where with a click of a button, a GAN image would be generated and can be downloaded. As bad people can simply upload deepfake media on social media where it is available for anybody to consume, these posts can explode in virality especially with the growing prevalence of social media usage in our daily lives. Look no further to the 2020 US election where there was a deepfake about one of the candidates that was shared by many of the American political leaders at that time and was viewed over 2.5 million times on Facebook [3]. Because of this, deepfakes pose a threat of identity theft, damaged reputations, and other detrimental consequences when in the wrong hands. Thus, requiring countermeasures which can detect deepfake media.

In combatting the rise of deepfakes there are two options: watermarking techniques or deepfake detection. The former fundamentally embeds a media with a "watermark" to verify its authenticity. To an extent, it would be able to involve blockchains. Nevertheless, these watermarks can easily be removed by the attackers which would prevent the deepfakes from being detected. The latter is one that is more effective in combatting deepfakes. Detection models would often exploit a certain feature to be able to distinguish real images from fake ones. While there are varying categorizations of these detection models by different researchers and experts [4,5], it can be classified into two methods which are the feature-based and machine learning detection methods.

Feature-based detection refers to the detectors that exposes and exploits certain features of the deepfake. Often times, they are the biometric [6], media [7], or generative model [8] features. These methods would usually have to do some processing to extract those features to be detected.

Machine learning detection on the other hand applies traditional machine learning algorithms (e.g., Support Vector Machines (SVMs) and Random Forests) or deep learning ones (e.g., Convolutional Neural Networks (CNNs) [9] and Autoencoders [10]) to the deepfake problem.

The recent advancement in machine learning models have seen Transformers go into computer vision tasks with Vision Transformers (ViTs). The hierarchical ViTs (e.g., Swin Transformers) were the ones that reintroduced convolution to Transformers though CNNs already had that [11]. However, those ViTs have surpassed the performance of CNNs. For those reasons, Liu et al. [12] introduced a model dubbed ConvNeXt which took a ResNet model and "modernized" with Transformer-like features while retaining the architecture of a CNN. This ConvNeXt model provided better results to the Swin Transformers on the ImageNet dataset.

To produce a better solution for detecting GAN-generated deepfake images, we propose the implementation of a ConvNeXt model in this paper. Seeing the recent success of

ResNets in detecting deepfakes [7], the use of ConvNeXts should be able to perform beyond the mark as they are built upon ResNets. Having outperformed ViTs in ImageNet, we believe ConvNeXts may exhibit similar groundbreaking performances on the distinguishment of deepfake [12]. We also utilize Gradient-weighted Class Activation Mapping (Grad-CAM) to visualize the decisions behind the classification done by the ConvNeXt model.

II. Related Works

A. GAN Deepfake Detection

State-of-the-art deepfake videos or images are often generated using GANs as these algorithms' generators are specialized to create an output that would fool the discriminator [2]. While a variety of other deepfake generation techniques are available through face swaps or autoencoders, the superior quality of GAN deepfakes eclipses that of their counterparts. Thus, making the GAN deepfakes extremely difficult to distinguish from real faces. For that reason, there are multiple attempts made by other researchers to create a detector that can accurately detect the GAN images or videos with their respective methods.

Li et al. [13] worked on a detector which utilized the abnormal eye blinking, in which deepfakes often lacked the blinking of the eyes, effectively discriminating between real and fake videos. Using the CEW dataset, they trained a VGG16 model to detect eye blinking patterns in the deepfakes. Employing an 80:20 train test split, the model achieved an AUC score of 0.99. Though the limitation of the model [13] is that it only accounts for the blinking cues and no other aspects like the dynamic pattern of eyes blinking.

Pu et al. [8] also implemented a model to differentiate real faces from GAN deepfake faces. Their work leveraged the unique patterns left behind by the generative models. NoiseScope takes a blind detection approach, meaning that it has no prior training on GAN images. This detection model is based on the attack and defense model. By extracting, classifying, and finally detecting the noise residuals and the generator's fingerprint, NoiseScope is able to achieve more than 92% in the F1 score for StyleGAN, BigGAN, PGGAN, and CycleGAN. Its best performance was on the BigGAN-BurgLV dataset with 99.68% F1 score.

By dynamically augmenting the image data, Das et al. [9] were able to improve the performance of the detection models. Their method of dynamic face augmentation was to cutout some parts on the face to help the model attend to the relevant regions of the image. Using established models in EffNet-B4 and XceptionNet, they subjected the augmented images to testing on the models and yielded AUC scores of 95.44% and 95.66%. According to their research [9], the models' accuracy increased by 15.2% to 35.3% when applied on existing architectures.

B. Anomaly Detection

As deepfake detection is frequently done with a binary classification model, this method would have two classes of "real" and "fake". Anomaly detection does not follow in this concept as it instead detects data which deviates significantly from the norm [14]. This means that there is technically only one class in the training stage. Models that apply anomaly detection assumes that the data supplied in training all belong to the same class. When those "anomalies" are fed into the model during testing, certain thresholds would calculate that the anomaly does not fit in with the rest of the data. This method has been adopted in fraud detection and cybersecurity to say the least.

The concept of anomaly detection can also be deployed in deepfake detection by treating deepfakes as the anomalies [10], although this is still a relatively unexplored area in deepfake detection. Khalid and Woo [10] implemented a Variational Autoencoder (VAE) to tackle the problem of limited deepfake data being released. The approach taken is founded on the understanding that the VAE would be able to learn to reconstruct real images or videos. This assumes that the latent space of deepfakes is different to those of real ones and so the model would not be able to reconstruct the deepfakes properly. The reconstruction error threshold defines the fine line between real and fake images or videos. Testing on NeuralTextures dataset, their one-class VAE achieved 97.5% accuracy. Their claim is that this model can be further explored to be used on GAN deepfakes images.

III. Methods

A. Dataset

As of now, there are many deepfake datasets available on the internet which may fulfil the criteria to be utilized for the research done in this paper but several constraints that limited our access prevented us from making use of a variety of deepfake datasets.

This paper's work would implement the use of the Flickr Face High-Quality (FFHQ) dataset as the source of the deepfake images. The FFHQ deepfake images are generated using GANs, specifically using the StyleGAN technique, with the intentions to further GAN development [15]. The images found in the dataset provide more demographic diversity with better coverages on accessories which makes it a great source of deepfake data. Whilst the real images are sourced from Flickr's real face dataset. The ratio of real to fake images collected is at a 1:1 scale.

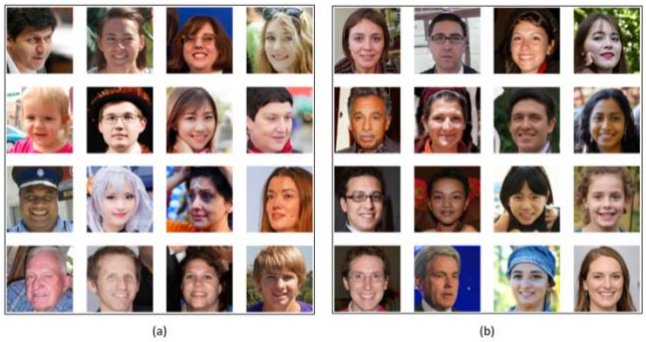

Fig. 1. Random sample of (a) real images and (b) FFHQ StyleGAN deepfake images.

Both the real and fake images were resized from 1024x1024 to 256x256 pixels. The size reduction was done to reduce the computation complexity during the processes in the model's training and testing. In an effort to reduce the complexity, we sampled 12,000 images from the 140,000 images available from the dataset. This sample was split into

a 5:1 train to test ratio in which both consist of a balanced number of real and fake images.

B. Data Pre-Processing

Drawing inspiration from the principles of the dynamic face augmentation work [9], we also conducted image augmentation and transformation on the images as it should be able to build a more robust model after training. The artificial augmentation of images allows the model to extract more information, supplying the model with more variation of data, while also reducing the possibility of overfitting [16].

In augmenting our set of images, we did horizontal, vertical, and rotational flips. Adding distortions and color jitters helps to increase the diversity of images. The images were also normalized to means [0.485, 0.456, 0.406] and standard deviations [0.229, 0.224, 0.225] on the three color channels. These were done on the train set. The test set virtually had minimal augmentations of resizing and normalization to mimic the actual images in the real-world scenario.

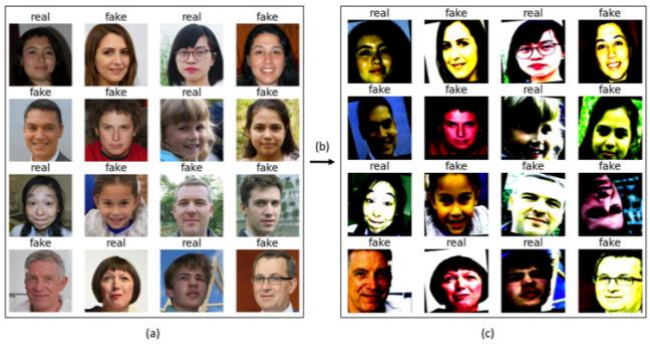

Fig. 2. (a) Train images undergo the (b) augmentations which includes the flipping of images, color, and distortions that results in the (c) augmented images.

We have also implemented a random seed to ensure the reproducibility of the experiments done during the entire process.

C. ConvNeXt Model

The ConvNeXt model, based on the ResNet model, has undergone several modifications in the CNN architecture. There are several notable changes that includes increasing the kernel size to a 7x7 dimension allowing the convolution to read off more data in the locality, and using a computing ratio of 1:1:3:1 which imitates the Swin Transformer computing ratio [17]. ConvNeXts also have an inverted bottleneck design in each convolution block which reduces the total computations. Basing off of the ConvNeXt model by Liu et al. [12], there are several sub-models varying in their sizes where the smallest ConvNeXt has 29 million parameters while the largest has 350 million parameters. This work employs the smallest version of ConvNeXt, ConvNeXt-T, to detect the deepfakes.

As the ConvNeXt has been pre-trained on ImageNet, the work needed to be done is a lot lesser as the model had been built and trained on voluminous data. Therefore, minimal editing of the model was needed during this process of work. Mainly, the major change was to replace the output nodes to fit the classification problem.

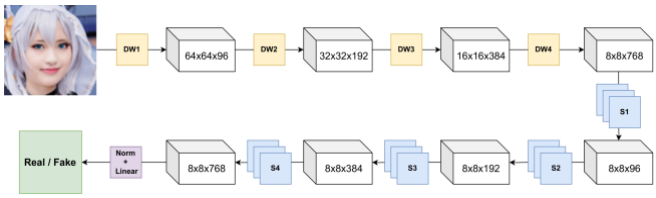

Fig. 3. ConvNeXt-T step-by-step process of detecting GAN deepfakes.

Fig. 3 above shows the model at its core having four downsampling layers, four stages of convolution. Each stage has 3, 3, 9, and 3 blocks of convolution respectively. The output is then normalized and passed to the linear layer for prediction.

D. Grad-CAM

In this realm of computer vision, CNNs have been a reliable tool that can convolve images and learn the important features to do image classification. Though understanding the rationale behind their decisions has been a formidable task until Class Activation Mapping (CAM) was introduced to solve that problem. CAMs implement the global average pooling to obtain object localization which is represents the predicted class scores and is visualized with the help of a heat map [18]. Grad-CAM improves on CAM by introducing the use of gradients of the target class to localize the features, resulting in a better mapping of important features [19].

The implementation of Grad-CAM in this paper is to see and understand what parts of the image factors in majorly in the prediction made. The heat map of Grad-CAMs is interpreted just like any other heat maps where the colors of blue to red references low to high importance. Using this, we would be able to come up with the understanding of how deepfakes are different in terms of how the model interprets them to be.

IV. Results

A. Model Performance

Having prepared the training and test data by augmenting them, we could set the hyperparameters of the ConvNeXt-T model. We started by setting the learning rate of the AdamW optimizer to 0.0001 and introduced a scheduler to update the learning rate. The decay of the learning rate would decay every 2 steps at a rate of gamma 0.1. Having the learning rate of the model decay every now and then allows the model to improve in learning those complex patterns found in the set of training images [20]. Training the model on 10 epochs, it was visible that the model performed brilliantly from the very start with ConvNeXt-T reaching the pinnacle of its performance during the fourth iteration. At its best, the model was able to perform at a 99.50% accuracy and had 99.49% F1 score.

In experimenting with the model, we also compared when the loss function was replaced from cross entropy loss to binary cross entropy loss. The reason for this was because this was a binary classification problem in which it should perform similarly. Subsequently some parts of the model were readjusted to fit for the binary cross entropy loss function. In reality, the change in loss function was not ideal as it basically predicted like a flip of a coin.

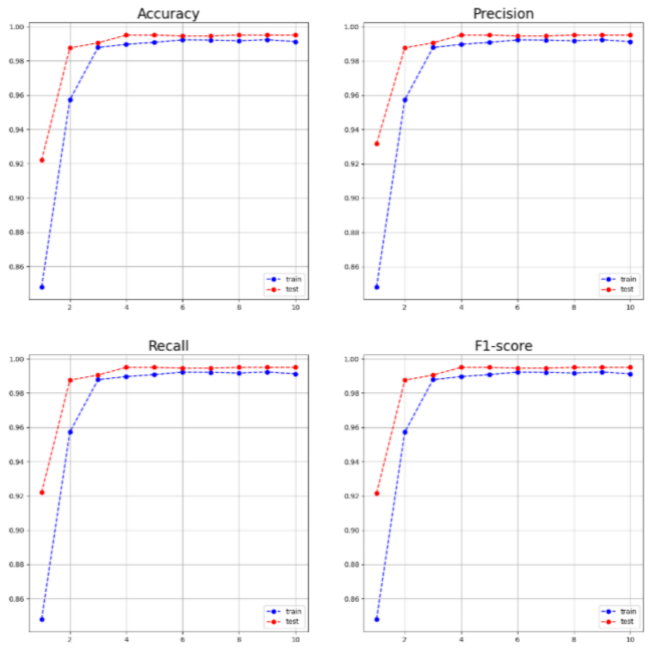

Fig. 4. Performance of the ConvNeXt-T model illustrated by the accuracy, precision, recall, and F1 score by iteration. The x-axis shows the epoch while the y-axis shows the score of each metric.

TABLE I. COMPARISON OF PERFORMANCE OF THE MODELS

Model Name	***Variations***	***Image Size***	***Accuracy***
ConvNeXt-T	Cross Entropy Loss Function	256x256	99.50%
ConvNeXt-T	Binary Cross Entropy Loss Function	256x256	50.00%
Autoencoder	0.0035 Reconstruction Threshold No Augmented Images	256x256	55.20%
Autoencoder	0.0025 Reconstruction Threshold	152x152	51.60%

[a.] The autoencoder models were solely trained on real images and the testing included the deepfakes and real images to find the reconstruction error.

Table 1 proves that the ConvNeXt-T with cross entropy has outperformed all the models that we have experimented on by a remarkable margin. The anomaly detecting autoencoders were trained on real images, but they could not detect the GAN deepfakes nevertheless. Understanding the reconstruction error, the deepfakes' reconstruction error was falling in the ranges of the real images' ones. This made it tough to set the threshold between real and fake images. We also see that the image size has effects to the performance of deepfake detection as the smaller images had lower accuracy which was seen in the autoencoder. Lowering the image size is a trade-off as it relieves the CPU which in then lowers the performance.

B. Interpretation of the ConvNeXt Model

Selecting the best model from the experiment, the utilization of Grad-CAM let us visualize how the CNN discerned between the two classes of images. Grad-CAM was applied after the final convolution in the fourth stage since that part contains all the information necessary for the model to make its decision.

Fig. 5. Grad-CAM on random sample of real images

Fig. 6. Grad-CAM on random sample of GAN deepfake images

Based on the Grad-CAMs in Fig. 5 and 6, they illustrate the features of the images that have an impact on the deepfake detection. Elucidating the interpretations of the real images, the Grad-CAM technique highlighted that the important features were mostly found around the facial regions. These activating features also expanded around the face in some cases. The map showed that there was high intensity of the features indicated by the prominent presence of redness on the map.

Regarding deepfake faces, we observed that the activation features were notably more spread out and less intense when compared to the class activation map of the real images. The mapping of important features in these fake images were also more variative as few images had more weight in the facial regions while others did not.

A particular observation is that the propensity of deepfake images having a spread-out map may be correlated to the density of the images themselves. The distribution of deepfake density suggests that the latent space between the datapoints in the fake images were reflected in the convolutions done in the ConvNeXt-T model when visualized with Grad-CAM.

TABLE II. DENSITY DISTRIBUTION OF IMAGES OF AUTOENCODERS

Autoencoder Image Density	**Image Type**	
	Real	***Deepfake***
Density Average	11303.9314	10160.6407
Density Standard Deviation	8.1763×10^{-12}	1175.9394

[b.] The average and standard deviation of the density were extracted from the fourth experiment in Table 1. This density distribution trend follows in the other autoencoder experiment conducted.

Table 2 shows that the standard deviation of real images is very low; implying they are mostly within the proximity of the average density stated. However, the deepfake density shows a rather different understanding than the average says. Instead of a gaussian distribution, the averages of the deepfake density were in two groups of values nearing 9,000 or those nearing 11,000. Averaging those numbers would get the density stated in Table 2. Perhaps this could be the reasoning as to why deepfake images show a varied Grad-CAM which at times resemble those of real images whilst other instances exhibit low intensity and scattered regions of activation on the map.

V. CONCLUSION

In this work, we have proposed the use of a ConvNeXt model to differentiate between real and GAN generated deepfakes. With 18 blocks of convolution variants, 4 downsampling convolution layers, and over 4.5 billion operations, the model was able to reach and supersede performances of ResNets and Transformers in detection of forged imageries by achieving an accuracy of 99.50%. We also delved into anomaly detection which had autoencoders decode and encode the images which we used the reconstruction error to rebuild the image. This method of anomaly detection that was experimented on unfortunately did not result in a satisfactory result, closing the doors on anomaly detection for GAN deepfake detection [10].

Taking the ConvNeXt-T as the model to dissect, the logic behind the model decisions were visualized using Grad-CAM in which we summarized that the real and fake images had clear differences. Real images tended to have more activation features in the facial region. These features were also more intense with more red found on the map. Fake images did not follow in this step in which they had a diverse feature map. Due to the scattered features on the map, it led to a more widespread and less intense Grad-CAM while sometimes they were identical with the real images in terms of Grad-CAM.

Strikingly, we presumably found a correlation between the density of deepfake images from autoencoders and the Grad-CAM results of deepfake images. The disparity between the Grad-CAM results of deepfake images is reflected in the distribution of density in deepfakes.

REFERENCES

[1] M. Somers, "Deepfakes, explained," *MIT Sloan*, Jul. 21, 2020. https://mitsloan.mit.edu/ideas-made-to-matter/deepfakes-explained

[2] I. J. Goodfellow *et al.*, "Generative Adversarial Networks." arXiv, Jun. 10, 2014. Accessed: Jul. 09, 2023. [Online]. Available: http://arxiv.org/abs/1406.2661

[3] S. Greengard, "Will deepfakes do deep damage?," *Commun. ACM*, vol. 63, no. 1, pp. 17–19, Dec. 2019, doi: 10.1145/3371409.

[4] T. Zhang, "Deepfake generation and detection, a survey," *Multimed Tools Appl*, vol. 81, no. 5, pp. 6259–6276, Feb. 2022, doi: 10.1007/s11042-021-11733-y.

[5] R. Tahir *et al.*, "Seeing is Believing: Exploring Perceptual Differences in DeepFake Videos," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, Yokohama Japan: ACM, May 2021, pp. 1–16. doi: 10.1145/3411764.3445699.

[6] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, "Multi-attentional Deepfake Detection." arXiv, Mar. 08, 2021. Accessed: Jul. 09, 2023. [Online]. Available: http://arxiv.org/abs/2103.02406

[7] Y. Li and S. Lyu, "Exposing DeepFake Videos By Detecting Face Warping Artifacts." arXiv, May 22, 2019. Accessed: Jul. 09, 2023. [Online]. Available: http://arxiv.org/abs/1811.00656

[8] J. Pu, N. Mangaokar, B. Wang, C. K. Reddy, and B. Viswanath, "NoiseScope: Detecting Deepfake Images in a Blind Setting," in *Annual Computer Security Applications Conference*, Austin USA: ACM, Dec. 2020, pp. 913–927. doi: 10.1145/3427228.3427285.

[9] S. Das, S. Seferbekov, A. Datta, M. S. Islam, and M. R. Amin, "Towards Solving the DeepFake Problem : An Analysis on Improving DeepFake Detection using Dynamic Face Augmentation." arXiv, Aug. 25, 2021. Accessed: Jul. 09, 2023. [Online]. Available: http://arxiv.org/abs/2102.09603

[10] H. Khalid and S. S. Woo, "OC-FakeDect: Classifying Deepfakes Using One-class Variational Autoencoder," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 2794–2803. doi: 10.1109/CVPRW50498.2020.00336.

[11] Z. Liu *et al.*, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows." arXiv, Aug. 17, 2021. Accessed: Jul. 18, 2023. [Online]. Available: http://arxiv.org/abs/2103.14030

[12] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s." arXiv, Mar. 02, 2022. Accessed: Jul. 02, 2023. [Online]. Available: http://arxiv.org/abs/2201.03545

[13] Y. Li, M.-C. Chang, and S. Lyu, "In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking." arXiv, Jun. 11, 2018. Accessed: Jul. 09, 2023. [Online]. Available: http://arxiv.org/abs/1806.02877

[14] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep Learning for Anomaly Detection: A Review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, Mar. 2022, doi: 10.1145/3439950.

[15] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks." arXiv, Mar. 29, 2019. Accessed: Jul. 04, 2023. [Online]. Available: http://arxiv.org/abs/1812.04948

[16] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[17] A. Singh, "ConvNext: The Return Of Convolution Networks," *Towards Data Science*, Feb. 10, 2022. https://medium.com/augmented-startups/convnext-the-return-of-convolution-networks-e70cbe8dabcc

[18] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning Deep Features for Discriminative Localization." arXiv, Dec. 13, 2015. Accessed: Jul. 20, 2023. [Online]. Available: http://arxiv.org/abs/1512.04150

[19] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," *Int J Comput Vis*, vol. 128, no. 2, pp. 336–359, Dec. 2019, doi: 10.1007/s11263-019-01228-7.

[20] K. You, M. Long, J. Wang, and M. I. Jordan, "How Does Learning Rate Decay Help Modern Neural Networks?" arXiv, Sep. 26, 2019. Accessed: Jul. 19, 2023. [Online]. Available: http://arxiv.org/abs/1908.01878