Deep Learning for Image Authentication: A Comparative Study on Real and AI-Generated Image Classification

Gaye Ediboglu Bartos
Óbudai Egyetem
6 PUBLICATIONS **8** CITATIONS

Serel Özmen Akyol
Kütahya Health Sciences University
15 PUBLICATIONS **18** CITATIONS

Deep Learning for Image Authentication: A Comparative Study on Real and AI-Generated Image Classification

Gaye Ediboglu Bartos
Alba Regia Technical Faculty
Obuda University
Szekesfehervar, Hungary
gaye.ediboglu@uni-obuda.hu

Serel Akyol
Faculty of Engineering and Natural Sciences
Kütahya Health Sciences University
Kütahya, Turkey
serel.akyol@ksbu.edu.tr

***Abstract*— The rapid advancement of AI-based image generation techniques has led to the increasing need for methods to distinguish between real and AI-generated images. In response, this study applies two popular deep learning algorithms namely Residual Networks (ResNets) and Variational Autoencoders (VAEs) to recognize AI-generated synthetic images. The algorithms adopted in the study approach the problem differently: ResNets aim to distinguish between real and fake images by analyzing high-level features and structural patterns within the images, while VAEs approach the problem as an anomaly detection task. The performance of the abovementioned algorithms is assessed using the CIFAKE image dataset containing 120000 images (60,000 synthetically generated images and 60,000 real images). The experimental results indicated that ResNets outperform VAEs in accurately classifying real and AI-generated images. VAEs, approaching image classification from an anomaly detection point of view, yielded insufficient performance in the case of the CIFAKE dataset. The findings underscore that the model selection, training objective, dataset attributes, and the specific task are all significant factors influencing a model's performance.**

Keywords*— deep learning, image authentication, real image, ai-generated image, fake image, synthetic image, ResNet, Residual Network, VEAs, variational** autoencoder, ***anomaly detection, feature extraction, image classification.

I. INTRODUCTION

The rapid evolution of artificial intelligence (AI) has led to an era where AI-generated images have become increasingly difficult to distinguish from authentic photographs. As the boundaries between real and AI generated synthetic content blur, it is crucial to develop effective methods for discerning between images created by AI algorithms (such as Generative Adversarial Networks [1] or Diffusion Models [2]) and those originating from the real world. Common applications of synthetic images include generating synthetic handwriting [3][4] and photorealistic faces [5]. With an effort to overcome the abovementioned challenge, this research seeks to employ two powerful deep learning architectures: Residual Networks (ResNet) and Variational Autoencoders (VAEs) [6][7]. Our objective is to develop and evaluate separate classification models based on ResNet and VAEs for discriminating between authentic real-world images and AI generated ones. CIFAKE image database consisting of 60000 real and 60000 fake images is used in training and evaluation of the models giving [8].

Detecting whether an image is a real image or an AI-generated one is a challenging task. To address this challenge, various approaches can be employed, including discriminator networks, anomaly detection, and feature extraction. Discriminator Networks are used as a part of generative models together with a generative network. The discriminator is responsible for distinguishing the real data from the data created by the generator. Therefore, discriminator networks such as in GANs can be used to detect fake images. Another approach to this challenge is to treat the synthetic images as anomalies based on how different they are from real images and use anomaly detection algorithms such as autoencoders and Variational Autoencoders (VAEs). Another common approach is to use a pre-trained deep convolutional neural network (CNN) [9] such as VGG, ResNet, or Inception to extract features from the images and then classify them according to the distinct features. Besides the abovementioned methods, combination of predictions from multiple models might improve performance. In this study, two separate algorithms were employed namely ResNet using feature extraction and VAEs using anomaly detection approach to classify real and fake images from the CIFAKE database. It is also worth mentioning that certain modifications on both architectures were also deployed to find the most suitable model for the target input.

ResNet, renowned for its prowess in image classification tasks, has been chosen as one of the primary models. Its capability to capture intricate visual features and patterns has made it a standard choice for various computer vision tasks. This study explores the potential of ResNet in addressing the critical challenge of distinguishing AI-generated images from real-world photographs, capitalizing on its discriminative abilities.

Conversely, VAEs, primarily designed for generative tasks, offer unique potential for detecting anomalies in image data. By encoding and decoding images, VAEs can reconstruct input data, making them valuable for anomaly detection. In our research, we harness the anomaly-detection capabilities of VAEs to create a distinct classification model. The objective is to assess whether VAEs can excel at identifying AI-generated images by detecting deviations from learned real-world data distributions. It should be noted that VAEs usually demand more computational power during training compared to models like ResNet for image classification due to their generative nature. The following section provides an overview of relevant research in the field.

II. RELATED WORK

The classification of real and AI-generated images, along with the detection of fake images, has become a crucial research area in recent years. The rapid growth of generative models resulted in the creation of realistic content such as images and text. In response to the advancements in fake content generation, techniques to detect such content have also emerged. In terms of AI generated image detection, most of the research focuses on the detection of deepfakes which are synthetic images of human faces. However, this research does not only focus on detection of fake human faces but also any other image containing skyline, animals, vehicles etc. thus it appears to be a more challenging task than detection of deepfake images. The challenge derives from the fact that animals and vehicles have a wide variety of shapes, sizes, and appearances, making it challenging to create a comprehensive detection model that can identify all possible variations accurately. Detection of fake human faces as a task is simpler because facial structures have certain consistent features and symmetry, allowing for the development of more reliable algorithms that can pinpoint inconsistencies associated with deepfake manipulations. In this section, the focus will be the research conducted on detecting AI generated rather than deepfake images. In the next section, the database and the adopted CNN architectures are explained in detail.

In 2021, Goebel et. al proposed a method to detect, attribute and localize GAN generated/manipulated images [10]. The proposed model was then evaluated using 2.76 million images (1.69 real images and 1.07 million fake images from 5 different GAN databases). The performance of the proposed method was compared with eight other CNN architectures such as ResNet, VGG16 [11] and InceptionV3[12] and it yielded promising results in detecting GAN generated images. In 2023, Dogoulis et. al. published a study to address the challenge of generalization for fake image detection in the cross-concept scenario caused by training the model on samples of random AI-generated images [13]. Their proposal includes a probabilistic method to evaluate the quality of generated images and training image detectors with those higher-quality images instead of randomly picked ones. The results indicate that using more realistic images to train the detector can significantly improve the generalization ability.

Xi et. al. proposed an ai-generated image detection method consisting of a dual-stream network comprised of a residual stream and a content stream [14] in 2023. To be able to test the capabilities of the proposed model, the authors created two databases consisting of AI-generated images using text to image AI generators. The performance of the proposed classifier was compared with seven other popular classifiers such as ResNet and CGNet [15]. The performance of the proposed fake image detector outperformed all seven classifiers on both the generated database and excelled when applied to mainstream computer graphics detection datasets. Zhu et al. put forward the GenImage database consisting of over 1 million pairs of AI-generated fake images and collected real image generated by the state-of-the-art diffusion models and GANs in 2023 [16]. The database does not focus on a specific class of AI generated images, it includes general images from 1000 image classes. In their study, in order to evaluate the GenImage dataset, different CNN architectures and image transformers were used.

In the subsequent sections, we demonstrate the methodologies, hardware and software employed, the datasets utilized, and the experimental results obtained through the application of ResNet and VAEs in image classification. Furthermore, we will conduct a comparative analysis to clarify the variations in performance between the two models.

III. EXPERIMENTS

In this section, we present the experimental setup, procedures conducted to assess the efficacy of deep learning models namely Variational Autoencoders (VAEs) and Residual Networks (ResNet) for image authentication. Our objective was to investigate and compare the models' performance in distinguishing real images from AI-generated images using the CIFAKE dataset.

The following sections put forward the details on the hardware and software used for the experiments, the dataset used, the modifications on partitioning of the dataset and image loading phase and finally the changes made on the deep learning models consequently.

A. *Experimental Hardware and Software*

The neural networks used for detecting AI-generated images were constructed using Python and key deep learning libraries such as TensorFlow and PyTorch [18][19]. All TensorFlow and PyTorch random seeds were initialized to ensure result reproducibility. The Variational Autoencoder (VAEs) and ResNet models were implemented using the resources provided by Google Colab within the Jupyter Notebook environment [20]. However, we encountered difficulties due to the substantial size of the input dataset, which consisted of 120,000 images. The challenge arose from the necessity to load and preprocess (resize) this extensive dataset before training. Consequently, we had to revise our initial approach to loading and processing images, incorporating optimizations to handle the large dataset efficiently.
The VAEs model was trained leveraging the GPU allocated by Google Colab, as was the ResNet model. All image data was processed at a resolution of 64x64 pixels. Google Colab provided the necessary environment and GPU resources for efficient computation and analysis of the models.

B. *The Dataset*

In this study, the CIFAKE image dataset is used to train and test the methods used for classification. CIFAKE dataset is a balanced dataset and includes 120000 images from two different classes namely real images (60000 instances) and AI-generated synthetic images (60000 instances) [8]. The real images are from the CIFAR-10[17] dataset and the synthetic images are generated using Latent Diffusion Model [2]. The generated images belong to the class of general images, they are not on a specific content such as human face or art. Samples belonging to real images category are shown in the Fig. 1 and samples from the synthetic image category are represented in Fig. 2 below. The partitioning of the dataset into training and test set is provided in the next section.

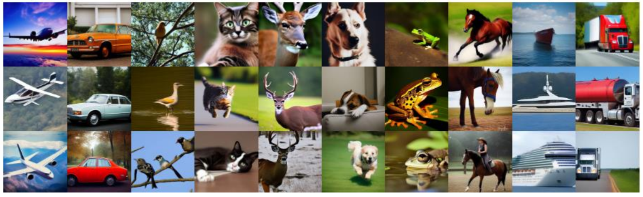


Fig. 1 Examples of AI-generated images from the dataset [8]

Fig. 2 Examples of real images from the dataset [8]

C. *Modifications on the Image Loading and Partitioning Phase:*

In this section the partitioning of the dataset and the changes on the image loading and resizing are provided. As previously highlighted, we encountered significant challenges stemming from the extensive size of our input dataset and the constraints posed by limited computational resources for image processing. Consequently, we devised specific modifications to mitigate this issue. This section provides an in-depth discussion of the challenges we faced, along with the solutions we implemented.

In the earlier attempts, all 120000 images from CIFAKE dataset were loaded and resized one by one without using explicit batch processing. The images were loaded and processed individually inside the for loops that iterated over the image files in the directories. However, it was exhausting for the system therefore to reduce the load on the GPU and optimize the code for handling many images, a few optimizations were implemented namely batch loading and parallel processing. Batch loading helps us to load the images in batches instead of loading all images at once. This process helps in managing memory more efficiently. At the same time, using parallel processing helps us load images concurrently thus speeding up the image loading process.

Since the experiments were carried out in the Google Collaboratory using python programming language, the “ThreadPoolExecutor” class is utilized to create a pool of threads that execute the function that loads the images concurrently, loading images in parallel for each batch instead of the for loop in the original code. This parallelism helps speed up the image loading process by utilizing multiple threads. Each thread processes a batch of images independently, improving overall efficiency. Another change to weigh down the code was made by using a more efficient image loading library. The “opencv” library for loading images was applied instead of the initial “PIL” library to try and improve the performance.

After several experiments, we further optimized the code by reducing the batch size. At first, a batch size of 32 was implemented, however, experiments showed that batch site 16 worked the best for our needs.

Additionally, the partitioning ratio was also determined at this step. Since ResNet model took less time to train, in order to determine the best portioning was decided using ResNet with 50 epochs and the batch size of 64. The table below shows that the partitioning of 75% training set and 25% test set provided the highest classification accuracy. Therefore, the abovementioned partition is used to train and test the models in the next section. The next section puts forward the changes made to the models in details.

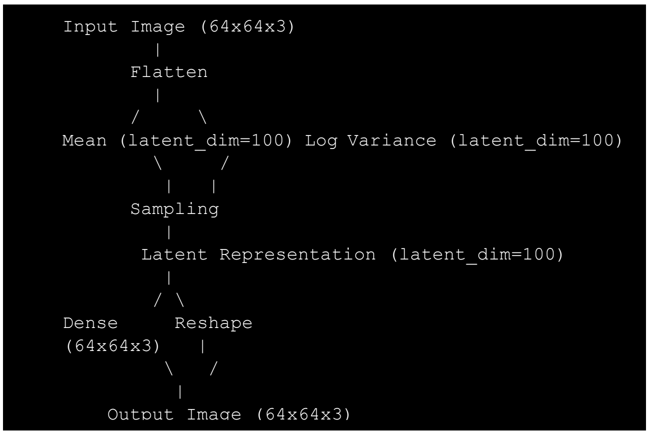
TABLE 1 PERFORMANCE OF DIFFERENT PARTITIONING OF DATASET

Size of Training Set (%)	Size of Test Set (%)	Accuracy
70%	30%	0.85
75%	25%	0.89
80%	20%	0.94

D. *Modifications on Algorithms:*

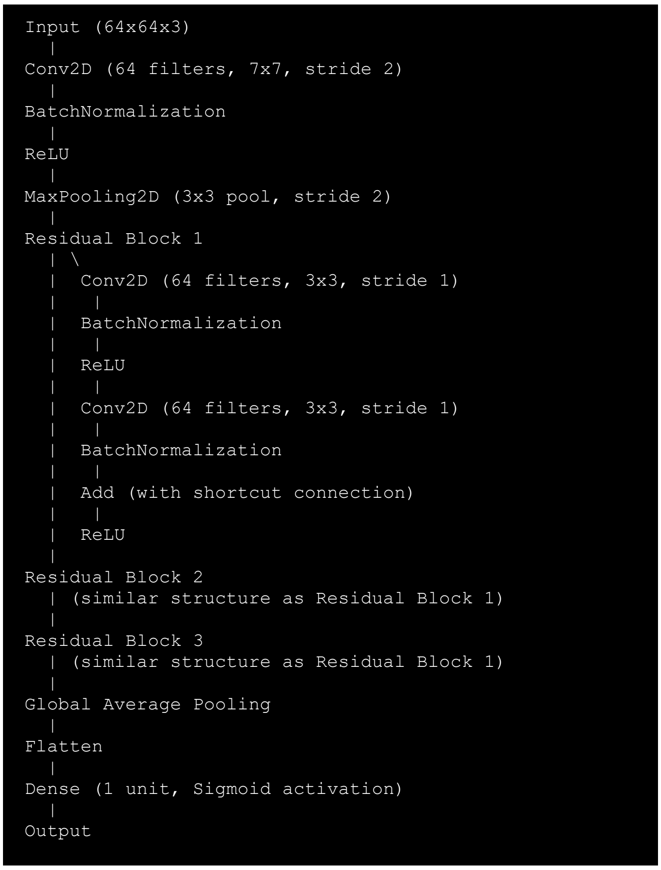
In this study two deep learning algorithms were adopted for the image authentication purpose namely VAEs and ResNet. The model architectures of the algorithms can be seen in Fig. 3 and Fig. 4 below [6][7].

Fine-tuning a VAEs involves making modifications to the architecture, loss function, or training process to better suit a specific task or dataset. In our study, due to the limited computational power, the fine tuning applied on VAEs was changing the batch size and number of epochs. We varied the batch size between 16 and 128 to assess the trade-off between computational efficiency and gradient accuracy. Additionally, experiments with different numbers of epochs were carried out. The number of epochs ranged from 40 to 80, to understand the model's convergence behavior and generalization.



```
    Input Image (64x64x3)
            |
          Flatten
            |
          /     \
    Mean (latent_dim=100) Log Variance (latent_dim=100)
            \      /
             |    |
          Sampling
             |
           Latent Representation (latent_dim=100)
             |
            / \
    Dense     Reshape
    (64x64x3)    |
             \    /
              |
        Output Image (64x64x3)
```

Fig. 3 Variational Auto Encoder Architecture



```
Input (64x64x3)
  |
Conv2D (64 filters, 7x7, stride 2)
  |
BatchNormalization
  |
ReLU
  |
MaxPooling2D (3x3 pool, stride 2)
  |
Residual Block 1
  | \
  |  Conv2D (64 filters, 3x3, stride 1)
  |    |
  |  BatchNormalization
  |    |
  |  ReLU
  |    |
  |  Conv2D (64 filters, 3x3, stride 1)
  |    |
  |  BatchNormalization
  |    |
  |  Add (with shortcut connection)
  |    |
  |  ReLU
  |
Residual Block 2
  | (similar structure as Residual Block 1)
  |
Residual Block 3
  | (similar structure as Residual Block 1)
  |
Global Average Pooling
  |
Flatten
  |
Dense (1 unit, Sigmoid activation)
  |
Output
```

Fig. 4 Residual Network Architecture

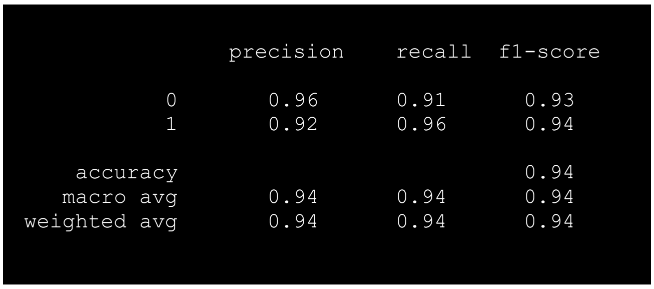
Common fine tuning performed on ResNet includes using a pre-trained ResNet Model, modification of the last classification layer and fine tuning hyperparameters such as learning rates and batch sizes. Similar to the changes on VAE, we tuned the batch size, exploring values from 16 to 128, to balance computation efficiency and gradient accuracy. In addition to those, the number of training epochs varied from 40 to 80, seeking an optimal convergence point and avoiding overfitting.

IV. RESULTS

In this section, we present the results of our comparative study on image authentication using VAEs and ResNet. We conducted experiments to evaluate the performance of these models on classifying real and AI-generated images using a dataset specifically curated for this research, namely the CIFAKE dataset. The performance metrics were calculated using "sklearn.metrics" module and they include precision, recall, f1 score, accuracy as well as weighted average and macro average of those metrics.

We first fine-tuned both the VAEs and ResNet models using different batch sizes and number of epochs. As anticipated, VAEs needed a lot more computational power during training compared to ResNet for image classification due to their generative nature. In contrast, models like ResNet do not have the added complexity of generative tasks thus resulting in less computational requirements during training. Training VAEs model on CIFAKE took at least 0.5 times more time than training ResNet model at almost every experiment using the same hardware. The table below shows the accuracy of both models using different hyperparameters.

In terms of the number of epochs tested were 40, 50, 60,70 and 80. However by watching the learning curve and performance metrics closely, it could be seen that 50 epochs were ideal for our domain. The classification accuracy did not change in a noteworthy way after 50 epochs. Additionally, after 40 epochs the validation loss reached a plateau state and the decrease in the training loss was insignificant. Therefore, for the future experiments and epoch number 50 was set. As can be seen in the table below, the batch size of 64 resulted in the highest accuracy in both models.



```
              precision    recall  f1-score

           0       0.96      0.91      0.93
           1       0.92      0.96      0.94

    accuracy                           0.94
   macro avg       0.94      0.94      0.94
weighted avg       0.94      0.94      0.94
```

Fig. 5 Performance Parameters of the best performing ResNet

TABLE 2 EXPERIMENTAL RESULTS

Batch Size	Accuracy ResNet	Accuracy VAEs
16	0.92	0.69
32	0.91	0.69
64	0.94	0.71
128	0.93	0.68

An attempt to further improve the classification performance of the VAEs was made by carrying out an additional experiment. The experiment included changing the loss function from 'binary_crossentropy' to 'mean_squared_error' and changing the learning rate parameter (0.001). Additionally, an early stopping was also adopted to prevent overfitting. However, the classification accuracy remained at 68.5% after the changes.

The results of our experiments demonstrated that ResNet achieved superior classification performance compared to VAEs in distinguishing real images from AI-generated images. ResNet achieved a maximum accuracy of 94% while VAEs achieved an accuracy of maximum 71%. The detailed performance metrics of the experiment yielding in the best performance can be in the Fig. 5 below. The difference in accuracy and efficiency suggests that ResNet, with its powerful feature extraction capabilities, is more effective for image authentication tasks on the CIFAKE dataset. Generative models such as VAEs and GANS take longer to train yet the VAEs in this domain yielded a lot lower performance. It is also important to note that the hyperparameter modifications have an impact on performance of both models. Fine-tuning these parameters improved the classification accuracy for both VAEs and ResNet, reinforcing the importance of hyperparameter tuning for optimizing model performance.

As mentioned earlier, VAEs are originally generative models however they can be used to classify real and synthetic images by trying to detect the anomalies within fake images. However, in the case of the CIFAKE dataset, the VAE model adopted had a significantly lower performance compared to the other model. The cause of it may be their generative and unsupervised learning nature, not having the same level of discriminative capacity as ResNet. Another reason for the poor performance may be its shallow architecture compared to the deep model of ResNet.

In summary, it is worth pointing out that the choice of model, training objective, dataset characteristics, and the task at hand all play crucial roles in determining the performance of a model. Overall, our findings highlight the efficacy of deep learning models, particularly ResNet, for image authentication tasks, emphasizing the necessity of tailoring hyperparameters to achieve optimal results.

V. CONCLUSION

This study put forward a comparative study on how to classify real and AI-generated images using deep learning. We employed two powerful deep learning architectures, Variational Autoencoders (VAEs) and Residual Networks (ResNet) and performed a comparative analysis using the CIFAKE dataset. VAEs are generative models, and they simulate real images and identify anomalies in AI-generated ones. ResNet are discriminative models, and they excel in feature extraction and are vital for distinguishing real from fake images. Both models used in the experiments were then optimized by changing batch sizes and training with different number of epochs.

Our experiments were facilitated within the Google Colab environment, providing us with the necessary computational resources, including GPU support. This allowed us to efficiently train and evaluate our models on a significant dataset. However, as mentioned earlier, the default GPUs provided by Google Colab were insufficient in several cases during our experiments. We encountered challenges due to the sheer size of the dataset, necessitating adaptations in our approach to image loading and processing. These modifications were crucial in optimizing the training process and overcoming resource limitations.

The results of our study revealed intriguing insights into the capabilities of VAEs and ResNet in the domain of image authentication. ResNet model exhibited commendable performance, showcasing the potential of deep learning in distinguishing real images from their AI-generated counterparts. However, the classification performance of VAEs was far from ideal and due to their generative nature, VAEs demanded more computational resources to operate optimally. These findings emphasize that it is crucial to pick the optimal approach for specific image classification tasks.

In conclusion, this study contributes to the evolving field of image authentication by providing insights into the efficacy of deep learning models approaching the same problem differently (generative model and discriminative model). The insights gained from this study also underscore the need for careful consideration in selecting an appropriate architecture based on specific application requirements. Nevertheless, there exist promising avenues for future exploration and enhancement of this study. One way could be to extend the scope of fine-tuning the algorithms to improve the performance. However, it should be noted that, such task would require substantial computational resources beyond what Google Colab could provide in our case.

VI. REFERENCES

[1] I. Goodfellow et al., “Generative adversarial networks,” Commun. ACM, vol. 63, no. 11, pp. 139–144, 2020.

[2] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-Resolution Image Synthesis with Latent Diffusion Models,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2022-June, pp. 10674–10685, 2022.

[3] T. Luhman and E. Luhman, “Diffusion models for Handwriting Generation,” pp. 1–17, 2020.

[4] G. Ediboğlu Bartos, Y. Hoscan, A. Kauer, and É. Hajnal, “A Multilingual Handwritten Character Dataset: T-H-E Dataset,” Acta Polytech. Hungarica J. Appl. Sci., vol. 17, no. 9, 2020.

[5] M. Kim, F. Liu, A. Jain, and X. Liu, “DCFace: Synthetic Face Generation with Dual Condition Diffusion Model,” pp. 12715–12725, 2023.

[6] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 770–778, 2016.

[7] D. P. Kingma and M. Welling, “An introduction to variational autoencoders,” Foundations and Trends in Machine Learning, vol. 12, no. 4. pp. 307–392, 2019.

[8] J. J. Bird and A. Lotfi, “CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images,” 2023.

[9] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nat. Methods, vol. 13, no. 1, p. 35, 2015.

[10] M. Goebel, L. Nataraj, T. Nanjundaswamy, T. M. Mohammed, S. Chandrasekaran, and B. S. Manjunath, “Detection, attribution and localization of GAN generated images,” IS T Int. Symp. Electron. Imaging Sci. Technol., vol. 2021, no. 4, 2021.

[11] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in International Conference on Learning Representations, 2015, pp. 1–14.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 2818–2826, 2016.

[13] P. Dogoulis, G. Kordopatis-Zilos, I. Kompatsiaris, and S. Papadopoulos, Improving Synthetically Generated Image Detection in Cross-Concept Settings, vol. 1, no. 1. Association for Computing Machinery, 2023.

[14] Z. Xi, W. Huang, K. Wei, W. Luo, and P. Zheng, “AI-Generated Image Detection using a Cross-Attention Enhanced Dual-Stream Network,” 2023.

[15] Y. Yao, Z. Zhang, X. Ni, Z. Shen, L. Chen, and D. Xu, “CGNet: Detecting computer-generated images based on transfer learning with attention module,” Signal Process. Image Commun., vol. 105, no. March, p. 116692, 2022.

[16] M. Zhu et al., “GenImage: A Million-Scale Benchmark for Detecting AI-Generated Image,” 2023.

[17] A. Krizhevsky, “Learning Multiple Layers of Features from Tiny Images,” 2009.

[18] TensorFlow. 2023. “An Open-Source Machine Learning Framework for Everyone.” https://www.tensorflow.org/.

[19] PyTorch. 2023. “An Open-Source Deep Learning Platform That Provides a Seamless Path from Research Prototyping to Production Deployment.” https://pytorch.org/.

[20] Google. 2017. “What Is Colaboratory.” https://colab.research.google.com/notebooks/intro.ipynb#recent=true.

[21] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. Deep Learning. MIT Press. https://www.deeplearningbook.org/.