CNN Generated Image Detection: A Study in Machine Learning Approaches

Md. Mujahidul Islam Mridha
Computer Science & Engineering
BRAC University
Dhaka, Bangladesh
tamim18016@gmail.com

Abstract

The progress in CNN technology has led to the creation of hyper-realistic synthetic images. As the line between real and synthetic images keeps getting blurrier, the development of reliable detection methods continues to become increasingly important. This research focuses on the application of CNN to detect synthetic images using a diverse dataset and evaluates the effectiveness of this approach. The findings aim to support the detection techniques in the face of the rapid evolution of generative technologies.

Keywords

CNN-generated Images, Detection, AI-generated Images, Neural Networks, Image Classification, Synthetic Images

ACM Reference Format:
Md. Mujahidul Islam Mridha. 2024. CNN Generated Image Detection: A Study in Machine Learning Approaches. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The advent of Convolutional Neural Networks (CNNs) has led to rapid development of generative technologies capable of producing hyper-realistic synthetic images. These images are getting harder to differentiate from real images with just the human eye. A recent study found that humans only classified 61% percent of synthetic images correctly [5]. These synthetic images even created outrage by winning art competitions [7]. The development of methods to differentiate between real and ai generated images has become very crucial at this point in time.

Generative Adversarial Models (GANs) and Latent Diffusion Models (LDMs) are the leading technologies in synthetic image generation [6, 9]. These technologies use high-dimensional data representations to create images that contain intricate details like real images. The quality of the images generated by these CNN based models has reached the point that traditional approaches are becoming redundant [9].

CNN based detection methods have emerged as a promising method in the field of image classification. CNNs specialize at recognizing subtle patterns and features that are not visible to the human eye, making them perfect for the task of classification. They are well suited to detect inconsistencies and artifacts in synthetic images. By training them on a large dataset comprising of both real and synthetic images we can use CNNs to classify real and synthetic images [9].

This research aims to find out the effectiveness of CNNs in the detection of synthetic images by using a comprehensive and diverse dataset. This research explores how CNN architectures can be optimized for this classification task and assess their robustness against the advancements of generative models. By addressing this challenge, this research seeks to contribute to the development of reliable detection methods of CNN generated images, safeguarding people against the risks of synthetic media.

2 Related Works

Many recent studies have raised concern regarding the problem of detecting images generated by CNNs. Cozzolino et al. [3] found that forensic classifiers failed to transfer their learned capabilities to detect CNN manipulations and often times performed close to a random choice. They proposed a new learning-based method to improve generalization.

Researchers found that the images generated by CNNs often leave artifacts. Many of these artifacts are generated in the up-sampling or down-sampling process. One such case is the checkerboard effect left by the deconvolutional layers [4]. Wang et al.[9] proposed that it may be possible create a universal detector to classify synthetic images. This means that once trained the model will be able to detect synthetic images regardless of the model used to generate or the dataset used to test. Their research finds that a standard image classifier trained on a single model (ProGAN) is able to generalize over a range of various generators and the results are fairly satisfactory. This discovery implies that all contemporary CNN based image generators share systematic flaws that prevent them from generating images indistinguishable from real images. These flaws might stem from systematic biases in the training data or architectural limitations of currently existing models. Data pre-processing in the form of common image post-processing operations such as JPEG compression, blurring, resizing etc are extremely critical for generalization.

Latent Diffusion Models are the state of the art technology in the modern image generation field [6]. Bird & Lotfi also suggest using image classifiers and proper tuning of them to detect images generated by LDMs[1].

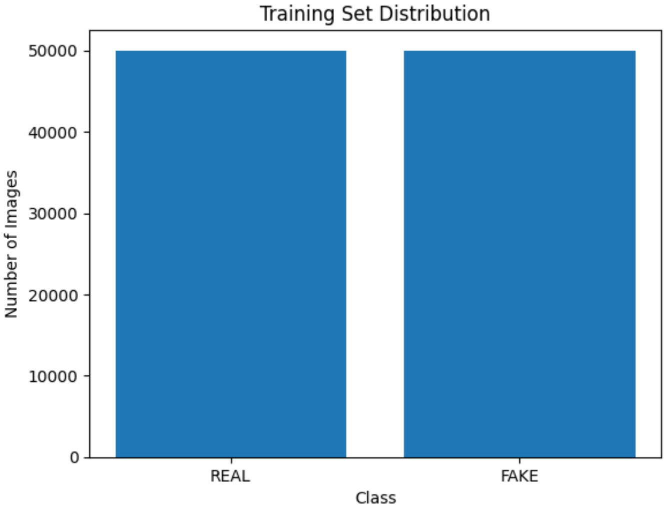
Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

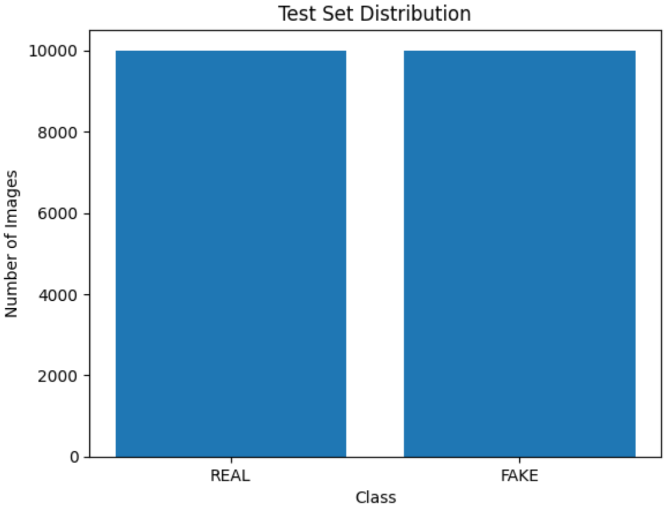
ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

3 Dataset

3.1 Class Distribution

For this research we are using the dataset 'CIFAKE: Real and AI-Generated Synthetic Images' by Jordan J. Bird [2]. This dataset contains 120,000 files consisting of both real and synthetic images.





The images are distributed for Training and Test purposes. The Training Set contains a total of 100,000. The images are distributed between 'REAL' and 'FAKE' classes. Each class contains 50,000 images. The Test Set contains a total of 20,000. The images are distributed between 'REAL' and 'FAKE' classes. Each class contains 10,000 images. The real images are from the CIFAR-10 dataset and the synthetic images are generated using Stable Diffusion version 1.4 We can see that data is equally distributed between classes which makes sure that our model will not be biased towards a single class.

3.2 Image Size Distribution

The images are all of uniform size. They share the following size:

Height	32px
Width	32px
Aspect Ratio	1:1

Table 1: Size Distribution

3.3 Sample

Below are sample images from both the "REAL" and "FAKE" classes

4 Methodology & Techniques

This section is dedicated to methodology. We will see how our data is augmented to make it suitable for the task at hand. Then we will discuss the training and validation process of our CNN models and finally discuss the testing method. Our data pre-processing methods are common image post-processing techniques. Then we will be splitting our dataset for training and validation. This ensures reliable model testing. Two common image classifiers, ResNet-50 and InceptionV3 are used for the classification process of real and synthetic images. This is a complete and concise guide, documenting each step of the process.

4.1 Data Pre-processing

All the data augmentation done here are common image post-processing methods. This is to increase the model's ability to generalize. First we are normalizing our pixel values. We are normalizing the pixel values of the input images from the range[0,255] to [0,1]. This ensures consistent input scale for the neural network which consequently results in faster convergences during training because it prevents large gradient updates. Then we randomly flipped images horizontally to introduce variation. Introducing variation helps the model learn invariant features regardless of the orientation of objects. Next we randomly applied upto 20% zoom to the images. This process simulates changes in object size and introduces scale variation. This enhances the model's robustness to scale variations. This is helpful in scenarios where objects may appear closer or further during testing. Finally, the dataset is split 80:20 for training and validation. This ensures that the model can be validated on unseen data during training. This helps us detect overfitting before entering testing phase. These augmentations help us increase the diversity of the training data and increase its ability to generalize over new and unseen data.

4.2 Training & Validation

For Training and validation we implemented transfer learning. We are using the ResNet-50 and InceptionV3 architecture pre-trained on the existing ImageNet dataset. The models are initialized excluding the original fully connected layers. This allows us to customize the layers according to our task. The inputs shape was set to (224,224,3). To optimize training, we froze all the except the last 50. This preserves the pre-trained features in the earlier layers and allows us to fine tune the deeper layers. Then a custom classification head is added that includes a Global Average Pooling layer that reduces feature maps to a single value per feature. Then a dense layer with 1024 neurons and ReLU activation is added. A dropout layer with a rate of 0.3 is added. This layer is an effective method to reduce overfitting. A final dense layer with Sigmoid activation is added for binary classification. The models are compiled using using the Adam optimizer for efficient gradient descent. Since this is a task of binary classification, Binary Cross-Entropy Loss is used as the loss function. Batch size was set to 64 and steps per epoch was set to 300. The models were trained for 5 epochs on a gpu to accelerate computation. The training history is recorded for analysis and evaluation.

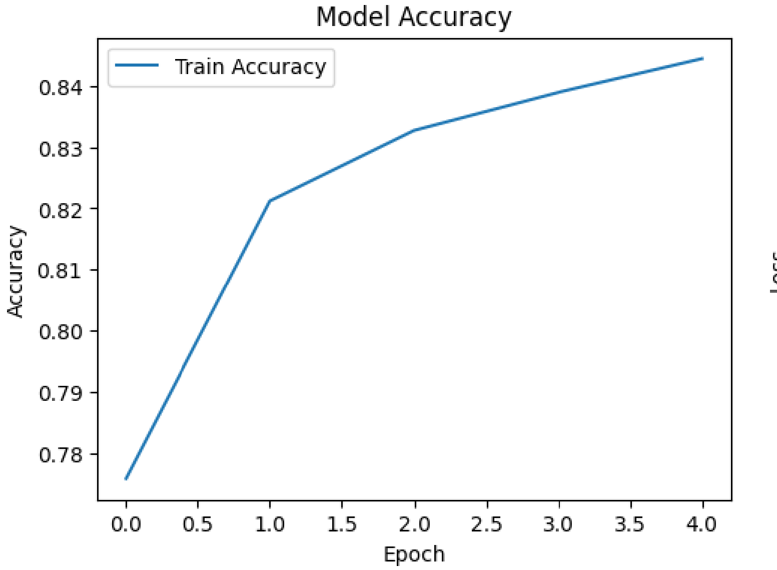
4.3 Testing

After the Training & Validation steps are complete we tested the models using the Test set and generated the confusion matrix. We analyzed the models using the confusion matrix. O indicates FAKE class and 1 indicates REAL class.

5 Results

5.1 Training & Validation Data

The training data for the ResNet-50 model are as follows:

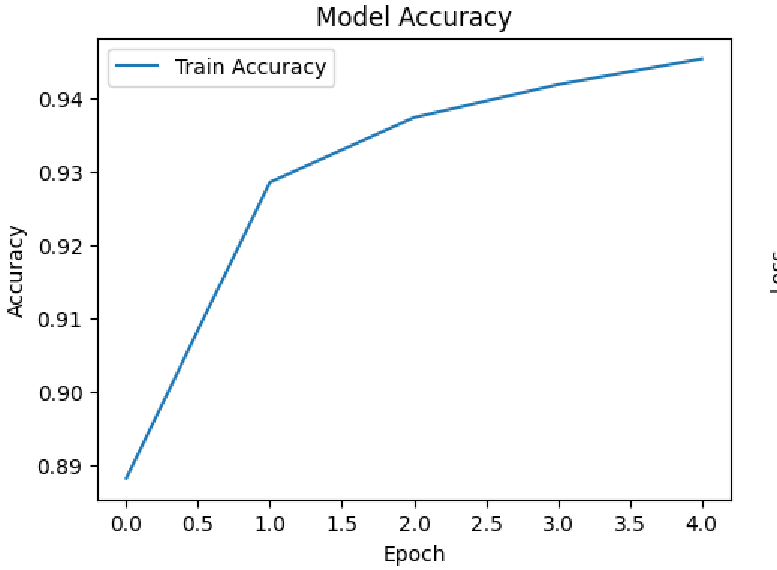


Epochs	Accuracy	Model Loss
1	0.7758	0.4933
2	0.8212	0.4063
3	0.8327	0.3823
4	0.8389	0.3726
5	0.8444	0.3663

Table 2: Training Data for ResNet-50

Accuracy for the validation set is: 51.20%

The training data for the InceptionV3 model are as follows:



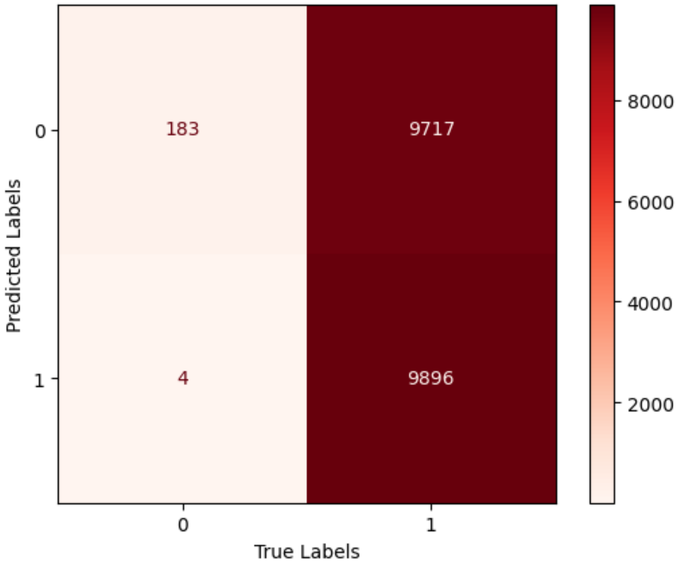
Epochs	Accuracy	Model Loss
1	0.8883	0.2781
2	0.9286	0.1841
3	0.9374	0.1632
4	0.9419	0.1484
5	0.9454	0.1393

Table 3: Training Data for InceptionV3

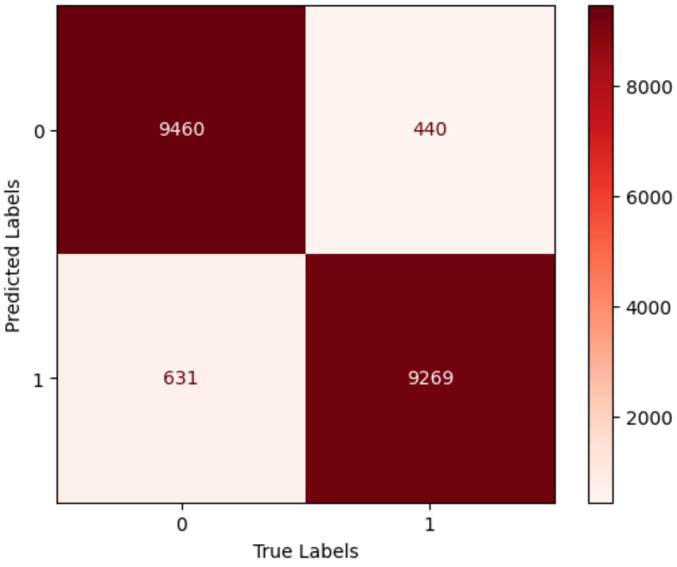
Accuracy for the validation set is: 94.78%

5.2 Test Data

Confusion matrix generated using Test Set by ResNet-50:



Confusion matrix generated using Test Set by InceptionV3:



Metric	ResNet-50	InceptionV3
Accuracy	50.90%	94.59%
Precision	50.46%	95.47%
Recall	99.96%	93.63%
Specificity	1.85%	95.56%
F1-Score	67.06%	94.54%

Table 4: Performance metrics comparison between ResNet-50 and InceptionV3.

6 Interpretation & Discussion

In the previous section we showed all the data we got from our models. Now we will interpret the data and discuss the implications. In the training phase, ResNet-50 had accuracy as high as 84.44%. We see model loss drop which indicates the model is learning. However, in the validation phase we see accuracy drop to 51.20% which further dropped to 50.90% in the Testing Phase. We see very high Recall (99.96%) and extremely low Specificity (1.85%). Precision is 50.46% and F1-Score is 67.06%. The performance of this model is not satisfactory. InceptionV3 on the other hand, had accuracy as high as 94.54% in the test phase and it was 94.78% in the validation phase. Accuracy in the final Test set is 94.59%. Precision, Recall, Specificity are 95.47%, 93.63%, 95.76% respectively. F1-Score is 94.54%. All these metrics are very satisfactory.

Here we can see that the InceptionV3 model produced really Promising results. However, the results for ResNet-50 are not satisfactory at all. High recall and Low Specificty indicates that This model is accurately labeling Real classes but it is struggling to label False Classes. This indicates class bias. The primary reason for class bias in imbalanced data. However, our dataset is extremely balanced, containing equal amounts of sample for both classes. This indicates a problem with the design philosophy of ResNet-50. InceptionV3 captures feature at various levels which makes it adept at identifying subtle artifact [8]. On the other hand, ResNet-50 uses residual networks which is more suited for complex artifacts [8]. Wang et al.[9] used ResNet-50 in their study on GAN based synthetic images and got satisfactory results. However, as time has progressed and LDMs were introduced, the artifacts left behind by the CNNs became much more subtle and became undetectable by ResNet-50.

Neural Network technology can be reliably used in the detection of synthetic images. However, as Generative technology improves we must understand the underlying patterns, use effective models and tuning to effectively detect synthetic images.

References

[1] Jordan J. Bird and Ahmad Lotfi. 2024. CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images. *IEEE Access* 12 (2024), 15642–15650. https://doi.org/10.1109/ACCESS.2024.3356122

[2] Birdy654. 2021. CIFake: Real and AI-Generated Synthetic Images. https://www.kaggle.com/datasets/birdy654/cifake-real-and-ai-generated-synthetic-images Accessed: 2025-01-02.

[3] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva. 2018. ForensicTransfer: Weakly-Supervised Domain Adaptation for Forgery Detection. *arXiv preprint arXiv:1812.02510* (2018). https://arxiv.org/abs/1812.02510

[4] Augustus Odena, Vincent Dumoulin, and Chris Olah. 2016. Deconvolution and Checkerboard Artifacts. *Distill* (2016). https://doi.org/10.23915/distill.00003

[5] A. Pocol, L. Istead, S. Siu, S. Mokhtari, and S. Kodeiri. 2024. Seeing is No Longer Believing: A Survey on the State of Deepfakes, AI-Generated Humans, and Other Nonveridical Media. In *Advances in Computer Graphics. CGI 2023*, B. Sheng, L. Bi, J. Kim, N. Magnenat-Thalmann, and D. Thalmann (Eds.). Lecture Notes in Computer Science, Vol. 14496. Springer, Cham. https://doi.org/10.1007/978-3-031-50072-5_34

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

[7] Kevin Roose. 2022. An AI-Generated Picture Won an Art Prize. Artists Arent Happy. (2022).

[8] S. R. Shah, S. Qadri, H. Bibi, S. M. W. Shah, M. I. Sharif, and F. Marinello. 2023. Comparing Inception V3, VGG 16, VGG 19, CNN, and ResNet 50: A Case Study on Early Detection of a Rice Disease. *Agronomy* 13, 6 (2023), 1633. https://doi.org/10.3390/agronomy13061633

[9] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. 2019. CNN-generated images are surprisingly easy to spot... for now. *arXiv preprint arXiv:1912.11035* (2019). https://doi.org/10.48550/arXiv.1912.11035