Detection of Deepfake images using CNN

Tanvir Y. Sarker (tys180001)

Jared Seifert (jws170000)

Ikhlaq Ahmad (ixa190000)

Wednesday 27th November, 2024

Abstract

In recent years the emergence the number of deepfakes published has risen as a result of the advent of AI models has been published in the internet with the intention of causing drama related to a topic, farming reputation in websites like YouTube and Reddit, and damaging the reputation of a person and putting their livelihood at risk. This model intends helps to combat this behavior of ensuring a legitimacy of a photo by using machine learning model which employs various techniques to build a generalized model to detect these deepfake images.

In this age of information, humans are becoming more informed than ever before. However, with the sophisticated use of machine learning techniques, and with the help of large amounts of data, the authenticity integrity of information is becoming seriously questionable. Machine generated text, images, and videos have been circulating over the internet. However, the level of realism has now dramatically increased due to access to more computational power and vast amounts of data. Digital images have recently become a prime source of concern about false information. Additionally, with every new improvement in the machine training algorithms and models, it is now becoming harder to distinguish between real and fake images. Social media platform users are regularly being exposed to enormous number of machine-generated images and videos. Many engineers and computer scientists are working to develop and train models that can detect machine generated imagery. In this project, we have developed one such model to accurately distinguish between real and deepfake images by using custom convolutional neural networks (CNNs) and Resnet50, a pre-trained Convolutional Neural Network (CNN) to extract the deep textural features as a base model to leverage transfer learning. Among the models, a customized CNN was developed, incorporating additional layers such as a dense layer, MaxPooling, and a dropout layer. This model underwent a comprehensive process including frame extraction, image feature extraction, data preprocessing, and classification. The results of this study highlights the potential of deep learning techniques, particularly CNN-based approaches, in effectively identifying deepfakes and enhancing security in digital media. Extensive experimental analysis has been done and the results show an improved detection accuracy compared to other methods.

1 Introduction

a History of Deepfakes

Image forgery has existed since the earliest instances of art. While the earlier methods of image tampering remained purely mathematics based, for example, as noted by Chakraborty [1], the early classification of objects using mathematics appeared in the late 19th century. A formal method of classifying faces was first proposed by Francis Galton in 1888. Since then, the work in face recognition remained largely dormant until the 1980s. With Artificial Intelligence gaining momentum in computing in the early 1990s, the invention of computer-based image manipulations has given the AI field a new dimension, computer vision. Since the 1990's, research interest in face recognition has grown significantly [1].

In terms of technology, deepfakes are the product of Generative Adversarial Networks (GANs) that appeared in 2010s. There are namely two artificial neural networks working together to create real-looking media. These two networks called 'the generator' and 'the discriminator' are trained on the same dataset of images, videos, or sounds [2]. The first then tries to create new samples that are good enough to trick the second network, which works to determine whether the new media it sees is real. That way, they drive each other to improve. A GAN can look at thousands of photos of a person and produce a new portrait that approximates those photos without being an exact copy of any one of them. Soon, GANs will be trained on less information and be able to swap heads, whole bodies, and voices [2].

Developing a computational model of face recognition is quite difficult, because faces are complex, multi-dimensional visual stimuli. After three decades of research effort, the Eigenface approach merged as the first real successful demonstration of automatic human face recognition. The low-dimensional representation of faces in the Eigenface approach is derived by applying Principle Component Analysis (PCA) to a representative dataset of images of faces. The system functions by projecting face images onto a feature space that spans significant variations among known face images. These significant features are termed "Eigenfaces" because they are the principal components of the set of training face images. This method can be classified as an appearance-based method that uses the whole face region as the raw input to a recognition system [1].

Being able to swap the whole face and body in images with realism was predicted in 2017 by Westerlund [2], but it has already become a new norm. In 2018, Khudeyer [3], researchers used GANs to create fake faces with multiple resolutions and sizes, then used different DCNN models to detect fake images. They apply a deep-face recognition system to transfer weights, and the network is fine-tuned using real or fake images in the [name not provided] AI Challenge [3].

According to Remya [4], advancement of data-driven and ML techniques like deep learning using Convolutional Neural Networks (CNNs) have shown exceptional results in general image classification problems. These CNNs are capable of learning rich feature representations directly from images. The layer activations of the pretrained CNNs models can be used as feature extractors for numerous applications in the field of computer vision [4].

b Current Methods

Deep Fakes are created using a combination of two algorithms, one that creates a synthetic image or video, and another that detects if the media is real or AI-generated. Custom approaches to CNNs, such as LSTM or Spatial Attention have revolutionized synthetic image generation. A person can now leverage pre-trained base models like ResNet, ImageNet, and DenseNet to reduce computational time. Custom Models and LoRa (Low-Rank Adaptation) weights can be found in sites like HuggingFace and Civitai. Readily available online tools exist to generate images using different models like Stable Diffusion, Midjourney, and DALL·E 3. These are easily trainable on a local machine or online and quickly deployable.

Many machine learning models are simply created from scratch and trained on relevant data. However, with the increase in dataset size and required computational power, many models can be built using pre-trained models on massive data. Many tech giants, such as Google and IBM, have been working to develop these base models to overcome deepfake detection, such as UADFV, DeepFake-TIMIT, and Celeb-DF. In addition, some companies released large-scale datasets to support researchers against this phenomenon. Google created a large-scale dataset, called DeepFakeDetection, to promote researchers in developing new methods for deepfake detection. Furthermore, Facebook and Microsoft presented a new dataset that has been included in the Deepfake Detection Challenge (DFDC) [5].

The manipulation detection methods presented until now focus on the detection of a specific form of targeted manipulation. This limits the application range of such methods because for creating a doctored image, several different processing operations can be applied to obtain a visually convincing result. For instance, in the case of splicing falsification, the forged part of the image can go through one or several basic operations such as rescaling, contrast enhancement and median filtering. Therefore, it is of great importance to develop general-purpose strategies that can detect different kinds of image manipulation operations [6].

Recently, deep-learning networks have proven more effective than traditional methods for multiple manipulation and operator chain detection in JPEG and non-JPEG domains. Towards this goal, some CNN frameworks are proposed to identify manipulation, adaptive forensics, and processing history. Unlike typical computer vision tasks, image forensics involves elaborating on features associated with tiny fingerprints left by particular image alterations. Several CNN-based forensic systems focus on constructing a single pre-processing layer (or initial layer) to better detect and counteract tampering [7].

c Impact of Deepfakes

As technology has advanced and deep-fake images have grown more sophisticated, the effects of these images have also grown in magnitude. In 2004, an edited image that depicted Jane Fonda next to presidential candidate John Kerry speaking to a crowd of Vietnam Veterans was circulated in order to convince people that John Kerry shared Fonda's controversial anti-war views. In reality, this was the combination of two different photographs that depicted the two subjects separately. The mass distribution of this image very likely could have had an impact on Kerry's campaign. [8] The term deepfakes originated in 2017 from the Reddit user 'deepfakes'. The user was later banned

in 2018 for sharing involuntary pornography, where they would transfer the likeness of an unsuspecting woman onto the body of another woman engaging in sexually explicit acts on video. [9] The psychological and social harm inflicted through this practice is impossible to either quantify or understate. However, this remains the most common use of this technology to this day, with women comprising the overwhelming majority of victims.

2 Methodology

a Dataset

We got our data from the dataset **AI vs Human Generated Images by Shirshak Acharya** which was retrieved from Kaggle. The dataset consisted of 720K files of real and fake images which were evenly distributed. The total size of the data on disk was over 70 GB. The images were of 20 different types of objects. These object types were airplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train, and tv monitor. There were over 18,000 each of real and AI-generated images for each object type. The images were originally 256x256 pixels in size in a .png format.

a.1 Exploratory Data Analysis

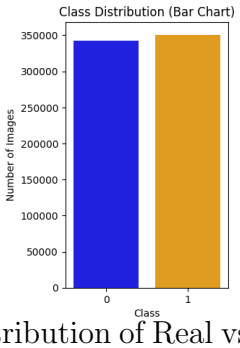

Figure 1: Distribution of Real vs Fake Images

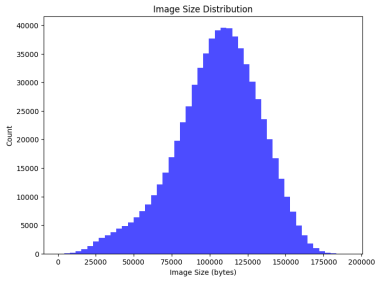

Figure 2: Distribution of Image Sizes

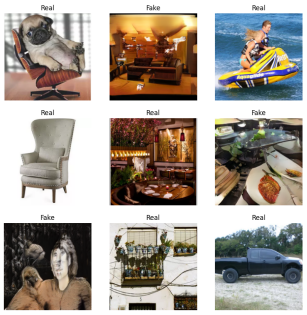

Figure 3: Sample data from the dataset

b Data Preprocessing

b.1 Classification Directories

The image files had to be first rearranged since they were originally categorized into sub-folders for 20 different categories, airplanes, etc. First, we had to rename all the files by prefixes denoting their object type to differentiate them and then aggregate them all into folders named real and fake.

b.2 Other preprocessing

We used other various methodologies to further process our data after the initial manual sorting of the data. The methods we used are listed here:

1. Creating image paths and labels into lists to create our training and validation splitting

2. Image normalization a method used to normalize images either between 0 and 1.

3. Image decoding - we decoded the image into 3 channels R, G, B

4. Resizing image to standardize the images to we learn on which was $224 \cdot 224$

5. Custom augmentations to generate more data diversity and randomness only on the training data to make it more generalizable. Augmentations that were done are:

   (a) Flipping image either left right up or down

   (b) Increasing or decreasing the brightness

   (c) Increasing or decreasing the contrast

   (d) Increasing or decreasing the hue with a max delta of 0.1

   (e) Increasing or decreasing the saturation

6. Autotune is used to optimize the the number of threads used to parallel processing of data

7. Batching training data into 32 images per batch

8. Prefetch being used to overlap the preprocessing and model execution of a training step

c Model

c.1 Environment

The model was trained using Graphic Processing Unit (GPU). Specifically, we utilized Nvidia GeForce RTX 3050 8GB DDR5. We installed cuDNN and CUDA libraries from Nvidia to be able to run the model.

c.2 ResNet50

The ResNet-50 model is a deep convolutional neural network architecture designed to tackle the vanishing gradient problem in deep neural networks. It achieves this by introducing residual connections (skip connections) that allow gradients to flow through the network more effectively. The following code customizes ResNet-50 using transfer learning and fine-tuning, combined with preprocessing, data augmentation, and training, to classify images into two classes: Real and Fake.

c.3 Key Components of the Code

1. Major Imports and Dependencies:

   The code relies on the following key libraries:

   - TensorFlow/Keras: For the deep learning framework and model implementation.
   - Matplotlib: For visualizing data and training performance.
   - Pillow (PIL): For image loading and manipulation.
   - NumPy: For efficient data processing.
   - Pandas: For data handling.
   - Sk-learn: For data splitting and meterics
   - OS: To handle file paths and directory traversal.

2. Data Loading and Preprocessing:

   Input Data:
   image_paths contains paths to image files, and labels are corresponding class labels (0 for Real and 1 for Fake).

   Data Preprocessing:

   Images are loaded using lists for simple and efficient pipeline handling. Then, all the data is split using sk-learn into 80/20 split. 80 percent for training and 20 percent for testing.

   Preprocessing Function:
   The preprocess_input function from keras.applications.resnet50 normalizes the images according to ResNet-50's expected input, ensuring compatibility.

3. Model Architecture:

   Transfer Learning:

   The pre-trained ResNet-50 base model is imported using keras.applications.ResNet50. The include_top=False argument excludes the fully connected (classification) head, allowing customization for binary classification. The input_shape parameter ensures images are resized to (224, 224, 3) to match ResNet-50's input requirements.

   Custom Top Layers:

   A GlobalAveragePooling2D layer replaces the dense layers from the original architecture, condensing the feature maps while reducing overfitting. A dense layer with 128 units and ReLU activation is added, followed by dropout for regularization. The final dense layer has 1 unit with sigmoid activation for binary classification.

4. Fine-Tuning:

   The layers of ResNet-50 are frozen initially to train only the custom head layers. During fine-tuning, the top layers of ResNet-50 (layers after conv5_block3_out, typically layer 150) are unfrozen to allow training of deeper network layers with a smaller learning rate.

5. Data Augmentation:

   Custom data augmentation method is used to augment the training data, including:

   - Random Flips
   - Rotations
   - Shifts
   - Brightness Changes
   - Contrast
   - Hue

   This increases better data processing and reduces over and under fitting.

6. Training and Callbacks:

   Training Pipeline:

   train_dataset and val_dataset are batched and prefetched to optimize training performance. A batch size of 32 is used.

7. Callbacks:

- EarlyStopping: Halts training when validation loss stops improving.
- ModelCheckpoint: Saves the best-performing model during training.
- ReduceLROnPlateau: Reduces the learning rate if validation loss plateaus.

8. Metrics:

   Binary Cross-Entropy Loss: Optimized for binary classification tasks.

   Accuracy: Monitors the proportion of correctly classified samples.

   Additional Metrics:

   Precision, recall, F1-score, mean squared error (MSE), and root mean squared error (RMSE) are calculated to assess model performance further.

9. Visualization:

   Images are visualized before training to verify data loading and augmentation.

   Training history (accuracy and loss) is plotted to monitor model performance.

3 Results

Our model showed near perfect accuracy when presented with real images, and 95% accuracy when presented with fake images. This suggests that our model is very good at identifying when an image is authentic but is still being fooled by a small number of highly realistic deepfake images.

Accuracy				
Class	Accuracy			
Real	1.00			
Fake	0.95			
Classification Report				
	Precision	**Recall**	**F1-Score**	**Support**
Real	0.95	1.00	0.97	68411
Fake	1.00	0.95	0.97	70154
Accuracy			0.97	138565
Macro Average	0.97	0.97	0.97	138565
Weighted Average	0.97	0.97	0.97	138565

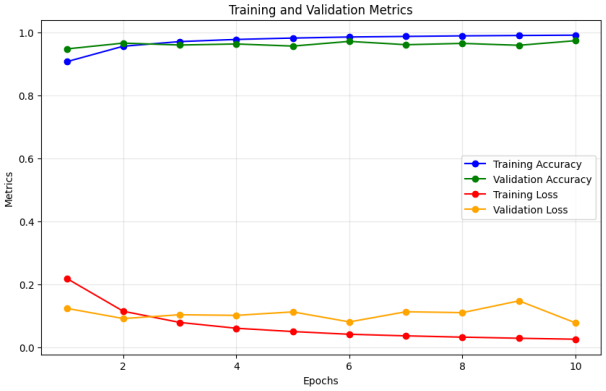

Figure 4: Training and Validation Metrics

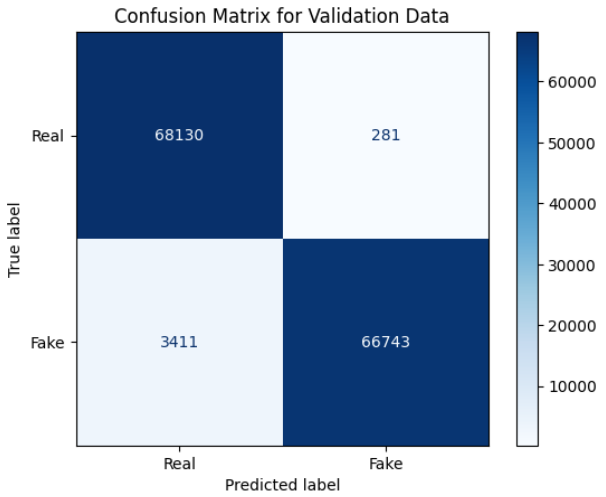

Figure 5: Confusion Matrix

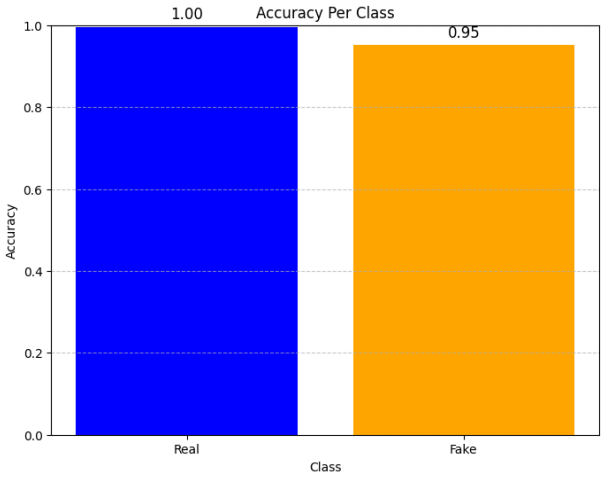

Figure 6: Accuracy per Class

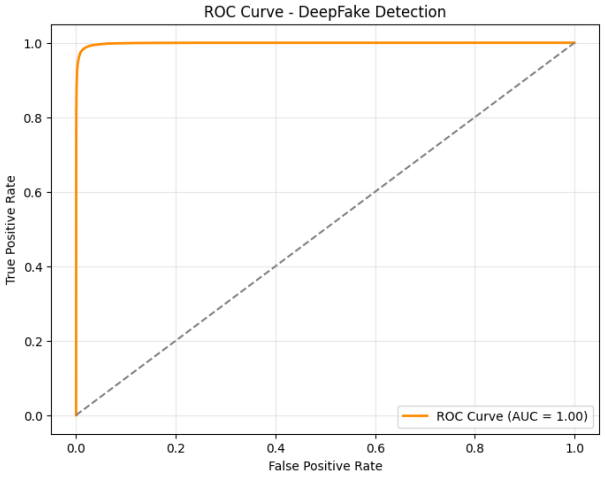

Figure 7: ROC

Metric	**Value**
Mean Squared Error (MSE)	0.0266
Root Mean Squared Error (RMSE)	0.1632
Accuracy	0.9734
Precision	0.9958
Recall	0.9514
F1 Score	0.9731

Table 1: Model Evaluation Metrics

4 Conclusion

Our model displays very high accuracy in detecting whether an image is real or synthetically generated using AI models. This high degree of accuracy allows us to reliably use this model as a tool for differentiating real versus generated media in a wide range of settings. The model could be further trained in order to determine if a piece of artwork was crafted by the hand of an artist, or if an image of a politician is genuine or being used as misinformation.

While our dataset included a wide range of objects, it did not contain any images with humans as the primary subject. The most damaging deep-fakes are those that imitate a person's likeness to defame their character or inflict psychological harm to them. Further training is needed in order to more accurately identify whether an image depicting a person is legitimate or AI-generated. The main limitation in our study was the computing power required to train our model on such vast amounts of image data. Our model could be further improved with additional training data and time if we had access to dedicated cloud computing resources. Additionally, about 30 thousand images were lost from our dataset due to an error made when rebuilding our file directory. Recovering this data and retraining our model might yield slightly improved results.

The next steps for this model would be to further train it on face data until it can accurately determine if an image of a person is authentic. Then, training the model on even more object types would allow for a more generalized model that could determine the legitimacy of images of all kinds. We would then like to implement our model into a mobile application that could determine the authenticity of images seen while scrolling

through various social media platforms or websites, to empower the user against the growing threat of misinformation and disinformation.

References

[1] D. Chakraborty, S. K. Saha, and M. A.-A. Bhuiyan, "Face recognition using eigenvector and principle component analysis," *International Journal of Computer Applications*, vol. 50, no. 10, pp. 42–49, Jul 2012.

[2] M. Westerlund, "The emergence of deepfake technology: A review," *Technology innovation management review*, vol. 9, no. 11, 2019.

[3] R. S. Khudeyer and N. M. Almoosawi, "Fake image detection using deep learning," *Informatica*, vol. 47, no. 7, 2023.

[4] K. R. Revi and M. Wilscy, "Image forgery detection using deep textural features from local binary pattern map," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 5, pp. 6391–6401, May 2020.

[5] S. Yavuzkilic, A. Sengur, Z. Akhtar, and K. Siddique, "Spotting deepfakes and face manipulations by fusing features from multi-stream cnns models," *Symmetry*, vol. 13, no. 8, p. 1352, Aug 2021.

[6] I. C. Camacho and K. Wang, "A comprehensive review of deep-learning-based methods for image forensics," *Journal of Imaging*, vol. 7, no. 4, p. 69, Apr 2021.

[7] V. Kadha, V. V. N. J. S. L. Nandikattu, S. Bakshi, and S. K. Das, "Forensic analysis of manipulation chains: A deep residual network for detecting jpeg-manipulation-jpeg," *Forensic Science International: Digital Investigation*, vol. 47, p. 301623, Dec 2023.

[8] C. Shen *et al.*, "Fake images: The effects of source, intermediary, and digital media literacy on contextual assessment of image credibility online," *SSRN Electronic Journal*, 2018. [Online]. Available: https://doi.org/10.2139/ssrn.3234129

[9] T. Kirchengast, "Deepfakes and image manipulation: Criminalisation and control," *Information & Communications Technology Law*, vol. 29, no. 3, pp. 308–323, 2020.