Scientific Computing, Modeling & Simulation
Savitribai Phule Pune University

Master of Technology (M.Tech.)
Programme in Modeling and Simulation

Mini Project Report

Fake Image Detection

Gaurav Prakash Lute
MT2313

Academic Year 2023-25

Scientific Computing, Modeling & Simulation
Savitribai Phule Pune University

Certificate

This is certify that this report titled

Fake Image Detection

authored by

Gaurav Prakash Lute (MT2313)

describes the project work carried out by the author under our supervision during the period from 15 October 2024 to 16 November 2024. This work represents the project component of the Master of Technology (M.Tech.) Programme in Modeling and Simulation at the Department of Scientific Computing, Modeling & Simulation, Savitribai Phule Pune University.

Mihir Arjunwadkar, Professor
SCMS-SPPU, Pune, India

Arun Banpurkar, Head
SCMS-SPPU, Pune, India

Scientific Computing, Modeling & Simulation
Savitribai Phule Pune University

Author's Declaration

This document titled

Fake Image Detection

authored by me is an authentic report of the project work carried out by me as part of the Master of Technology (M.Tech.) Programme in Modeling and Simulation at the Department of Scientific Computing, Modeling & Simulation, Savitribai Phule Pune University. In writing this report, I have taken reasonable and adequate care to ensure that material borrowed from sources such as books, research papers, internet, etc., is acknowledged as per accepted academic norms and practices in this regard. I have read and understood the University's policy on plagiarism (`http://unipune.ac.in/administration_files/pdf/Plagiarism_Policy_University_14-5-12.pdf`).

Gaurav Prakash Lute
MT2313

Abstract

This report addresses the challenge of distinguishing between real and AI-generated (fake) images using deep learning techniques. We developed a Convolutional Neural Network (CNN) from scratch to classify images into real or fake categories, achieving notable accuracy. This document outlines the problem domain, previous attempts at solutions, data exploration, modeling methods, results, and future directions.

Acknowledgements

I am deeply indebted to **Prof. Mihir Arjunwadkar** for his invaluable guidance throughout the course of this project. His expertise and daily support in the analysis and design of algorithms greatly contributed to the success of this work. I am also grateful for the comprehensive reference materials he shared, which have empowered me to undertake and complete such research independently in the future.

I extend my sincere gratitude to **Prof. Mihir Arjunwadkar** for serving as my internal guide and to all individuals who provided assistance, directly or indirectly, during this project.

Finally, I wish to thank my colleague for their constant encouragement and support during my time at the Department of Scientific Computing, Modeling, and Simulation(CMS).

Contents

Chapter 1

Introduction

Artificial intelligence(AI) has become very powerful in recent years, especially in creating new things. This has led to many useful and creative applications,but it has also created a problem. It's becoming harder to tell real content from fake content made by AI. This has made it important to develop tools that can identify and categorize images created by AI.

A real image is an accurate visual representation captured using a camera or sensor. It directly depicts the subject as it appears in the real world.

A fake image is one that is been changed or completely made up, usually through editing software or advanced AI tools. People often edit photos by altering backgrounds or adding things that weren't there, like making someone appear in a place they weren't. Deepfakes are a more high-tech version of this, where AI is used to create very realistic but completely fake videos or photos, sometimes to make it look like someone is doing or saying something they never actually did. Fake images can also be just photos taken out of context or altered in ways that make them seem real, even though they are not.

1.1 Problem Domain

This project focuses on computer vision and deep learning, specifically targeting the detection and classification of images as either real or fake(AI-generated). Training Convolution Neural Network (CNN) models to be robust and generalize well enough for unseen data is the core challenge. The problem domain thus extends beyond technical aspects, touching on ethical, societal, and security considerations.

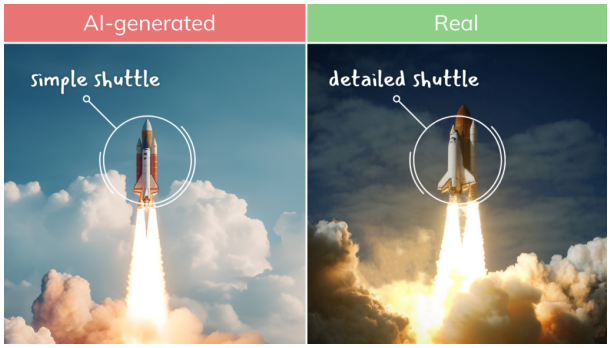

Figure 1.1: Real vs fake image

1.1.1 Our Goals

The primary objective of this project is to develop a deep learning model capable of accurately classifying images as real or fake. The target is to achieve a validation accuracy of at least 70%. The model must generalize effectively, minimizing false positives and false negatives to ensure reliability.

1.1.2 Relevance of the Problem

This problem is not just a theoretical issue, it has real-world consequences. With the rise of deepfakes and other AI-generated media, there's a growing concern about fake news and misinformation. These fake images and videos can be used to deceive people, spread false information, and damage trust in news sources. In today's digital world, where we're constantly bombarded with information, it is useful to have tools that can distinguish between real and fake content to ensure the accuracy of information.

1.1.3 The History of Fake Image Detection

Detecting fake images has been a challenge for a long time. In the past, people relied on manual checks and simple image analysis techniques to spot fake images. While these methods were helpful, they couldn't keep up with the increasing sophistication of AI-generated images.

With the development of machine learning, especially Convolutional Neural Networks (CNNs), we've made significant progress. CNNs are very good at recognizing patterns in images, making them ideal for image classification. Techniques like XceptionNet and VGGNet were used to improve the accuracy of detecting fake images. Researchers also combined CNNs with other advanced techniques to further enhance performance.

However, there are still challenges. One issue is that models can become too specialized, performing well on specific datasets but struggling with newer or more diverse types of generated images. This means we need to constantly update our models and training data to stay ahead of the evolving techniques used to create fake images.

Approach and Innovation

We plan to create a model using a type of artificial intelligence called a Convolutional Neural Network (CNN) to distinguish between real and fake images. We'll optimize the model's design, training process, and evaluation methods to make it as accurate and versatile as possible. To prevent the model from becoming too specialized and to improve its reliability, we will use techniques like data augmentation, transfer learning, and regularization. We willl test the model on a wide range of images to ensure it can handle different types of AI-generated images. By addressing the limitations of existing models, we aim to create a more effective tool for detecting fake images, which has significant implications for technology, ethics, and society.

By overcoming the limitations of previous models, such as over fitting, bias in the training data, and limited adaptability to new types of AI-generated images, we aim to create a more effective tool for detecting fake images. This will have positive impacts on technology, ethics, and society.

Chapter 2

Data Preprocessing

2.1 Dataset

The CIFAKE dataset is a valuable tool for researchers developing methods to identify fake images.
It contains a collection of images specifically designed to train computers to distinguish real photos from AI-generated ones.

What's in the Dataset?

CIFAKE has a total of 120,000 images divided into two categories:

- **Real Photos (60,000):**

  These come from a popular dataset called CIFAR-10, which features everyday objects like airplanes and dogs.

- **AI-Generated Images (60,000):**

  These were created using a powerful AI tool called Stable Diffusion. They're designed to look just like real photos, making it a tough challenge for computers.

The dataset is also split into two parts for training and testing:

- **Training Set (100,000 images):**

  Used by researchers to train their models on identifying real and fake images.

- **Testing Set (20,000 images):**

  Used to evaluate how well the trained models perform on unseen images.

The dataset was carefully curated to maintain balance between the two classes, allowing the model to learn equally from both types of images without being biased towards one category.

2.2 Exploratory Data Analysis (EDA)

2.2.1 Understanding the Dataset Distribution

- **Class Balance:**

  The first thing I will check if my dataset had a good balance between real and fake images. This is really important because if one class is much larger than the other, our model might end up favoring the majority class, leading to biased predictions. It was great to see that the dataset was balanced, with roughly equal numbers of real and fake images.

2.2.2 Checking Image Quality

- **Pixel Intensity Analysis:**

  Next, we dove into the average pixel intensity for both real and fake images. This analysis helped us identify any differences in color composition or brightness that could be significant for training the model. For example, fake images sometimes showed slight inconsistencies in brightness or color that might serve as distinguishing features for the model to learn.

- **Resolution Consistency:**

  we ensured that all images were the same size (32x32 pixels). This consistency was essential to avoid problems during training and kept the input data uniform, making the model's learning process smoother. This standardization prevented potential issues related to differing image sizes, which could have led to distortions or misalignment during feature extraction.

2.2.3 Preprocessing the Data:

- **Resize Images:**

  All images were resized to 32x32 pixels, a resolution that provided a good trade-off between computational efficiency and detail. We also normalized pixel values to a range between 0 and 1 by dividing by 255. This normalization step helps the training process by ensuring that gradient updates remain stable and prevents the model from being influenced by large pixel value differences

- **Final dataset**

  After completing the above process, a new DataFrame was created that includes two columns. The first column contains the file path for each image, while the second column provides a label indicating whether the image is fake 0 or real 1.

This detailed EDA provided a strong understanding of the dataset and helped inform crucial decisions for preprocessing and model building. By taking the time to explore the data thoroughly, we were able to spot potential challenges and address them before moving on to model development. This groundwork ensures a more reliable and well-prepared path for training a model that can effectively distinguish between real and fake images.

Chapter 3

Modeling Method

3.1 Convolution Neural Network (CNN)

The main method used for detecting fake images was a Convolutional Neural Network (CNN). This deep learning model was chosen because it works well with image data and can learn patterns effectively. The details below explain the model in depth.

3.1.1 Convolutional Layers

Convolutional layers were used to pick up important features from the input images. Small 2×2 filters moved across the images, detecting things like edges and textures. These layers created feature maps that highlighted key details in the images.

3.1.2 Activation Function

After each convolutional layer, the ReLU (Rectified Linear Unit) activation function was applied. This function adds non-linearity, helping the network learn complex patterns. The ReLU function sets all negative values to zero, making the network faster and preventing certain training issues.

3.1.3 Pooling Layers

Max pooling layers were added to make the feature maps smaller while keeping important details. Using a 2×2 window, the max pooling operation selected the largest value in each section, which helped reduce the size of the data and made the model more flexible to small changes in the input images.

3.1.4 Dropout Layers

Dropout layers were included to avoid overfitting. These layers randomly dropped some units during training, which forced the model to learn stronger, more general features. A common dropout rate of 0.5 was used, meaning half of the units were turned off during training.

3.1.5 Dense Layers

After the convolutional and pooling layers, a fully connected layer with 1,500 units was used. This dense layer, with ReLU activation, combined all the learned features and passed them on to the output layer.

3.1.6 Output Layer

The final output layer used a softmax function for binary classification, meaning it determined the likelihood of an image being real or fake. The softmax function produces a probability for each class, making it easy to interpret the results.

3.2 Model Compilation

The model compiled with the following settings.

3.2.1 Optimizer

The Adam optimizer was used for efficient training because it adjusts the learning rate automatically.

3.2.2 Loss Function

Binary cross-entropy was used since the task involved two classes: real and fake images. It measures how well the predicted probabilities match the actual labels.

3.3 Model Training

3.3.1 Batch Size

A batch size of 32 was used, balancing speed and memory usage.

3.3.2 Epochs

The model was trained for 50 epochs, with early stopping to prevent overfitting by monitoring the validation loss.

3.3.3 Early Stopping

To avoid overfitting, early stopping was applied. This technique monitors the validation loss during training, and if the loss stops improving for a set number of epochs, training stops early. This helps prevent the model from learning noise from the data.

3.3.4 Data Augmentation

Data augmentation was applied to make the training data more diverse and improve the model's generalization. Techniques such as random image flips, rotations, and zooms were used. This helps the model learn to recognize images even when they appear in slightly different ways.

3.4 Mathematical Model

Given an input image X represented as a 3D matrix of dimensions $H \times W \times C$ (height, width, and channels), the model's task is to compute an output $\hat{y}$, where $\hat{y} \in [0,1]$ represents the probability that the image is fake.

3.4.1 Convolutional Layer

Each convolutional layer l applies a set of filters w_l to the input X_l, producing an output:

$$X_{l+1} = \sigma(W_l \star X_l + b_l)$$

Where:

- $\star$ is the convolution operation.
- b_l is the bias term.
- $\sigma(.)$ is the activation function, such as ReLU defined as $\text{ReLU}(x) = \max(0,x)$.

3.4.2 Pooling Layer

A pooling layer reduces the spatial size (height and width) of the feature map to focus on the most significant features.

$$X_l = \text{Pooling}(X_l)$$

3.4.3 Output Layer

The final layer outputs a single value z, which is converted into a probability $\hat{y}$ using the sigmoid function:

$$\hat{y} = \frac{1}{1+e^{-z}}$$

The output range is between 0 and 1.

Chapter 4

Result and discussion

4.1 Result

In this section, I explain my model performance report. For analysis purposes, evaluation metrics such as accuracy and precision were calculated using the testing dataset. These metrics helped assess the models effectiveness.

There are various methods for model evaluation, but since my problem statement involves classification, we use precision, recall, and F1-score as evaluation metrics.

Classification Report

The classification report is used to evaluate the performance of a classification model. This classification report testing dataset this include 20000 images ,that images equally distributed in two class that is real and fake images.

Report observation

- In this report their are two classe 0 and 1 , it means fake and real images.
- **Precision**

  The proportion of correctly predicted positive observation out of all predicted positive. 87% of instances predicted as class 0 were actually class 0 and 64% of instances predicted as class 1 were actually class 1.

- **Recall**

  The proportion of correctly predicted positive observations out of all actual positives. Only 49% of actual class0 instances were correctly identified and 93%of actual class 1 instances were correctly identified.

Classification Report :

	precision	recall	f1-score	support
0	0.87	0.49	0.63	10000
1	0.64	0.93	0.76	10000
accuracy			0.71	20000
macro avg	0.76	0.71	0.69	20000
weighted avg	0.76	0.71	0.69	20000

Figure 4.1: classification report

- **F1-score**

  The F1-score balances precision and recall, showing how well the model identifies positives while avoiding false predictions. It's useful when both precision and recall need equal attention.

  The model performs for Class 0 with a score of 0.63, but it is not as strong as its performance for Class 1, which has a higher score of 0.76. This means the model does a better for predicting Class1.

Report Analysis

The model works well for Class 1, with strong recall and F1-score. However, it struggles with Class 0 where the recall is only 49%.

The overall accuracy is 71%, which is good, but their are some improvement in model in future scope.

Chapter 5

Conclusion

This project set out to create a dataset to help detect AI-generated images. We focused on collecting and organizing both real and synthetic images, then processing them to prepare for model training. By using tools like OpenCV and PIL, we resized and formatted the images to ensure consistency, building a solid foundation for deep learning.

Our work began with gathering a balanced collection of real and AI-generated images. We used automated techniques to load and organize these images, labeling them and merging them into a unified training set. Through image processing, we standardized the size and quality of the images, which was crucial for effective model training. This phase laid the groundwork for creating a reliable image detection system.

Looking ahead, there are several promising paths to explore. Training and comparing different models, such as CNNs, could boost detection accuracy. Data augmentation could make the model more robust, allowing it to handle a wider variety of AI-generated images. Adding explainability tools to the model would make its decision-making process clearer and build trust. Expanding the dataset to cover more types of images would improve the model's ability to adapt.

Investigating real-time applications would make the model more practical for detecting fake images as they are uploaded or shared. Finally, considering the social impact of image detection technology is essential to ensure it is used responsibly and ethically. These steps could lead to a more effective system for spotting fake images and promoting trust in digital content.

Reference

- An Introduction to Statistical Learninghttps://www.statlearning.com/
- Identifying Fake Images Through CNN Based Classification Using FIDAChttps://ieeexplore.ieee.org/document/9862034
- Early Stopping of Untrained Convolutional Neural Networkshttps://www.researchgate.net/publication/378011169_Early_Stopping_of_Untrained_Convolutional_Neural_Networks