DeepDetect : Detect Fake and AI-generated Images
Mid-Semester Project Report

Himang Chandra Garg Dasari Sai Harsh Nishil Agarwal Piyush Narula
Indraprastha Institute of Information Technology, Delhi
{himang22214, dasari22144, nishil22334, piyush22354}@iiitd.ac.in

Abstract

Our project seeks to address the critical issue of detecting deepfake images and AI-generated content, which have become increasingly sophisticated and challenging to distinguish from real media. The goal is to enhance the ability to discern genuine content from fabricated images, thereby reducing the potential for misinformation.

1. Introduction

1.1. Problem Statement

The recent rise in AI content, particularly deepfake images and videos, has raised serious concerns about the integrity of information online. This synthetic media is often indistinguishable from real content, hence making it very difficult for an average person to tell the truth—an aspect with large implications in public trust, privacy, and security.

2. Literature Survey

Detecting fake images with machine learning is a difficult and growing research field because image editing tools are becoming more accessible and user-friendly. Recent studies have focused on creating automated systems utilizing machine-learning methods in order to identify counterfeit images. This review of literature outlines the latest research in this area and addresses the difficulties and future paths ahead using the following 2 papers.

2.1. Detecting Fake Images Using Machine Learning

The paper ”Detecting Fake Images Using Machine Learning” by Mr. Akash K, Miss. Ahalya K, Mr. Dhinesh N, Miss. Diya Shereef delves into strategies for detecting manipulated images using both traditional image processing and machine learning approaches. The writers suggest a combination model that uses CNNs to extract features like color patterns, texture, and statistical characteristics from images. Next, these characteristics are inputted into a classifier that differentiates between authentic and altered images. The suggested model attains high accuracy and can be utilized in social media content moderation, news verification, and forensic investigations.

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

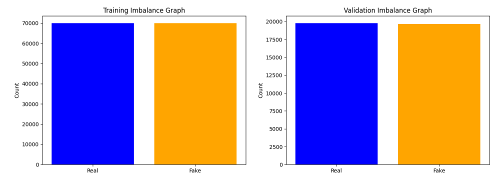
The effectiveness of deep learning models in differentiating real images from AI-generated ones is investigated in the paper ”Deep Learning for Image Authentication: A Comparative Study on Real and AI-Generated Image Classification” by Gaye Ediboglu Bartos and Serel Akyol. The research is centered around two models: Residual Networks (ResNet) and Variational Autoencoders (VAEs). ResNet, renowned for its strong feature extraction abilities, reached an impressive 94% accuracy in distinguishing between real and synthetic images in the CIFAKE dataset, comprising 60,000 real and 60,000 AI-generated images. On the other hand, VAEs, taking an anomaly detection stance, had a lower accuracy of 71% because of their generative nature and less discriminative architecture. The paper highlights how crucial it is to tune hyperparameters, such as batch sizes and epochs, in order to improve model performance.

3. Dataset: Exploratory Data Analysis

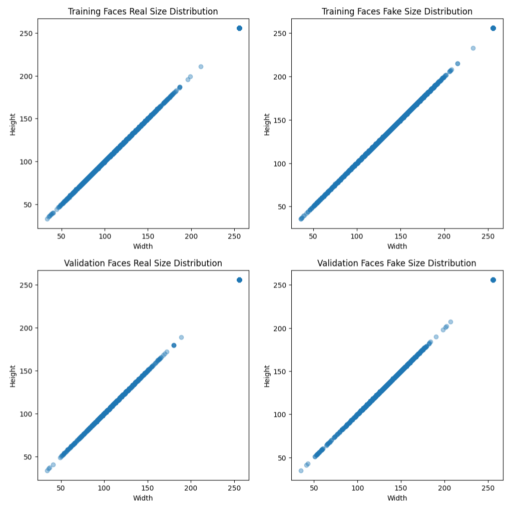
We have used this dataset.This dataset contains manipulated images and real images. The manipulated images are the faces which are created by various means. Each image is a 256 X 256 jpg image of human face either real or fake Exploratory Data Analysis (EDA) is crucial in this project for comprehending the features of both authentic and counterfeit images utilized in detection. By using bar graphs, scatter plots, and violin plots, we analyze the similarities in image size, color distribution, and saturation variances between authentic and AI-produced images. This examination assists in recognizing possible disparities in class, variations in features, and discrepancies, offering crucial understanding for successful model development and assessment.

3.1. Visualization of Dataset

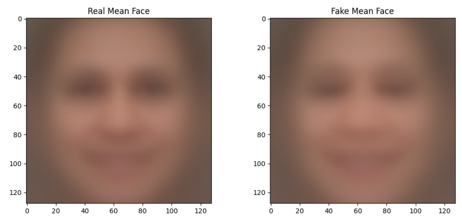
- **Class Imbalance Graphs:** In order to avoid overfitting, we analyzed the amount of images per category in both the training and validation sets. If there were a large imbalance in the number of images in a class, we would have conducted undersampling to achieve a more equal distribution in the dataset. We found no significant class imbalance.



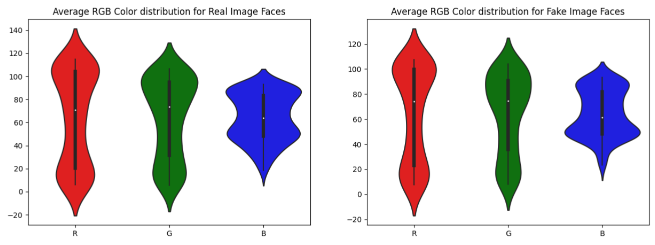
- **Scatter plot of face dimensions:** Faces were extracted from images for deepfake detection. A width vs height scatterplot was created but did not show a difference between real and fake image dimensions, suggesting it may not help distinguish them.



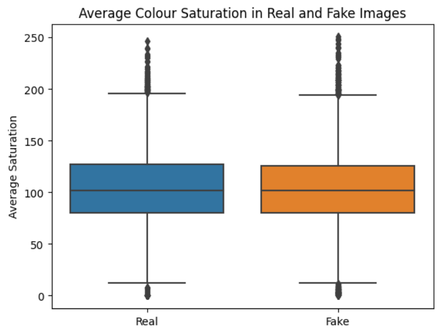
- **Mean Real and Fake Face images:** We created a visual representation of the average face extracted from both the real and fake image datasets in order to compare the mean face images of each type. We noticed distinct discrepancies since they did not resemble each other.



- **RGB Intensity Violin Plots:** RGB intensities were extracted using colour histograms from real and fake face datasets. Average frequency of intensity pixels in both datasets was calculated. Violin plots were created to compare RGB differences between real and fake images. Some differences were observed but they do not seem that significant.



- **Box plot for Colour Saturation:** We extracted colour saturation from real and fake face datasets. We generated box plots on basis of the saturation data. We observed no significant differences.



4. Methodology: Preprocessing and Model

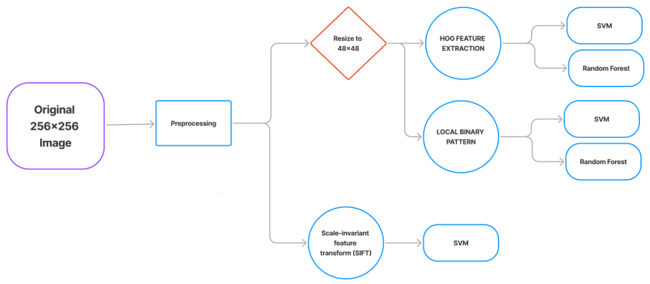


Figure 1. Workflow Flowchart

4.1. Why Resize

We resized the images to enhance computational efficiency and model performance. High-resolution images contain millio of pixels, leading to significant computational demands during training and inference. By reducing the image size, we lower the number of features, resulting in faster processing times and reduced memory usage. Additionally, traditional machine learning techniques, like SVMs and Random Forests, struggle with high-dimensional data due to the ”curse of dimensionality,” making it difficult to detect meaningful patterns.

4.2. Pre-processing and Data Augmentation

To ensure optimal model performance, we applied both pre-processing and data augmentation techniques to the dataset. **Pre-processing** aims to standardize and clean the data, preparing it for model training, while **Data Augmentation** artificially expands the dataset, introducing variability that helps the model generalize better to unseen data. For augmentation, we utilized the *Albumentations* library, which offers efficient augmentation strategies. The techniques applied are as follows:

4.2.1 Data Augmentation Techniques

- **Spatial Transformations:** We employed spatial transformations such as **Random 90-Degree Rotations**, **Horizontal Flip**, and **Vertical Flip**, each with a 50% probability (`p=0.5`). These transformations introduce variation in the orientation of the images without altering the essential features, allowing the model to become invariant to the position of key features.

- **Image Quality Transformations:** To account for real-world imperfections, we applied **Gaussian Noise** and **Blurring** to a subset of images (`p=0.2`). This step ensures the model remains robust even when presented with low-quality or distorted images, such as those affected by noise or motion blur.

- **Color Transformations:** We introduced variation in lighting and color properties using **Random Brightness/Contrast Adjustments** and **Hue/Saturation/Value (HSV) Shifts**, each with a 30% probability (`p=0.3`). These transformations help the model handle different lighting conditions and color variations common in real-world environments.

- **Mild Distortions:** To simulate minor geometric distortions, we included **Optical Distortion** and **Grid Distortion** with a 20% probability (`p=0.2`). These introduce slight warping and distortions, making the model more resilient to variations in perspective or imperfections during image capture.

4.2.2 Augmentation Pipeline

The augmentation process was implemented through a custom pipeline using `Albumentations`. The pipeline includes the following steps:

1. **Image Selection:** From each dataset split (`Train`, `Test`, `Validation`), we randomly selected a fraction of the images (approximately 10%) for augmentation, ensuring diversity while preserving the integrity of the original dataset.

2. **Image Augmentation:** Each selected image underwent the aforementioned transformations, and the augmented outputs were saved in their respective directories.

3. **Visualization:** To validate the transformations, we visualized pairs of original and augmented images to ensure the correctness and consistency of the augmentation pipeline, and to verify that key image features were preserved.

4. **Data Recording:** After augmentation, we created CSV files for each dataset split, containing the paths of the augmented images, the original images, their corresponding labels, and split information, facilitating smooth integration into the training pipeline.

This combined pre-processing and augmentation strategy was carefully designed to enhance model generalization and mitigate overfitting, particularly when working with limited datasets. By introducing diverse transformations, the model can better handle variability in real-world data while maintaining strong performance across all dataset splits.

4.3. Feature Extraction

Histogram of Oriented Gradients (HOG) is a feature descriptor used in image processing and computer vision. It captures the distribution of gradients and edge directions within an image, effectively representing the shape and structure of objects. HOG is especially useful for object detection due to its invariance to changes in illumination and its ability to handle variations in object scale. A csv file containing the resized images is read and the HOG feature extraction is applied to each image and the resulting embedding is saved in a separate csv.

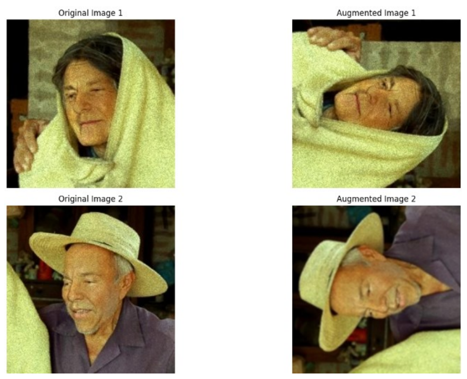


Figure 2. Before and After Pre-processing

Local Binary Patterns (LBP) is also used as a feature extraction technique due to its effectiveness in capturing texture information. LBP encodes local spatial patterns and contrasts in images, making it suitable for differentiating between real and synthetic content. The code reads a CSV file containing flattened representations of 48x48 pixel images and their labels (”Real” or ”Fake”). It applies LBP feature extraction to each image, generating a normalized histogram that represents the distribution of texture patterns. These extracted features are stored in a new DataFrame with the mapped labels and saved to a CSV file .

Scale-Invariant Feature Transform (SIFT) is a feature extraction method that identifies and describes local features in images. It detects keypoints across different scales and orientations, ensuring robustness to variations in image scale, rotation, and illumination. SIFT generates descriptors for these keypoints, which can be used for tasks such as image matching, object recognition, and classification.

In our project on detecting AI-generated images, SIFT is employed to extract distinctive features from the images. Its ability to capture detailed local information makes it effective for distinguishing between real and synthetic content.

4.4. Model Training

In our project, both Support Vector Machines (SVM) and Random Forests are employed for classification tasks using the features extracted from Histogram of Oriented Gradients (HOG) and Local Binary Patterns (LBP). For Random Forests, we follow the common practice of training with a subset of features, specifically using the square root of the total number of features $\sqrt{m}$, which helps enhance model performance and reduces overfitting. Meanwhile, SVM is utilized for training on the features extracted from the Scale-Invariant Feature Transform (SIFT), leveraging its effectiveness in high-dimensional spaces to improve accuracy.

5. Results and Analysis

5.1. HOG with SVM

The classification model employing Histogram of Oriented Gradients (HOG) features with Support Vector Machines (SVM) achieved the following performance metrics:

- **Accuracy:** 0.66, **Precision:** 0.68, **F1 Score:** 0.65, **ROC-AUC Score:** 0.73

The confusion matrix is as follows:

	Predicted Real	Predicted Fake
Actual Real	377	164
Actual Fake	206	343

The results indicate that the model has a balanced performance, with a slight edge in precision for the ”Fake” class. The accuracy of 66% suggests that the model is reasonably effective in distinguishing between real and fake images; however, there is room for improvement, particularly in recall for the ”Fake” class, which could lead to missing genuine instances of AI-generated images. The ROC-AUC score of 0.73 indicates a good capability in distinguishing between the two classes, reinforcing the model's utility for this task.

5.2. Random Forest(with m=4) with LBP

The classification model achieved an accuracy of 0.54, precision of 0.57, recall of 0.38, F1 score of 0.46, and ROC-AUC score of 0.73.

	Predicted Real	**Predicted Fake**
Actual Real	384	157
Actual Fake	341	208

The Random Forest model using Local Binary Patterns (LBP) performed poorly, achieving an accuracy of only 0.54. With a precision of 0.57 for fake images and a recall of 0.38, the model struggles to effectively differentiate between real and fake images, resulting in high false positive and negative rates. The confusion matrix highlights serious misclassification issues.

LBP may be limited by the small image size of 48x48 pixels, which restricts its ability to capture detailed texture information. Thus, the model's current configuration is inadequate for the task.

5.3. SVM with LBP

Accuracy: 0.57, Precision: 0.62, Recall: 0.37, F1 Score: 0.46, ROC-AUC Score: 0.73

The SVM model utilizing Local Binary Patterns (LBP) performed poorly, achieving an accuracy of only 0.57. This reflects significant challenges in differentiating between real and fake images, as indicated by the low recall of 0.37, resulting in many missed detections of fake images. The confusion matrix shows that out of 549 fake images, only 201 were correctly classified.

Since the Random Forests model exhibited suboptimal performance with Local Binary Patterns (LBP), following the same trend observed in the SVM results,t his suggests that LBP may not be well-suited for smaller images (48x48 pixels) and could benefit from larger image sizes to enhance feature extraction and overall classification accuracy.

	Predicted Real	**Predicted Fake**
Actual Real	420	121
Actual Fake	348	201

5.4. SVM with SIFT

The SVM model utilizing Scale-Invariant Feature Transform (SIFT) performed better than previous models, achieving an accuracy of 0.68. The precision for fake images was 0.61, indicating a reasonable level of correctness among the identified fake instances, while the recall score of 0.64 suggests that the model effectively captured a substantial number of fake images. Overall, the F1 score of 0.62 reflects a balanced performance between precision and recall.

	Predicted Real	**Predicted Fake**
Actual Real	392	233
Actual Fake	208	366

6. Conclusion

The traditional models, such as **Support Vector Machines (SVM)** and **Random Forest**, have not delivered satisfactory results. A key challenge has been the **high dimensionality** of the image data, which these models struggle to manage efficiently. To address this, we applied the following preprocessing steps:

- **Image Resizing:** Images were resized to reduce dimensionality, helping the models handle them more effectively.
- **Feature Extraction:** Techniques such as **Histogram of Oriented Gradients (HOG)**, **Local Binary Patterns (LBP)**, and **Scale-Invariant Feature Transform (SIFT)** were used to extract features from the images. These methods aimed to capture essential patterns while reducing the dimensional complexity of the data.

Despite these efforts, the dimensionality reduction resulted in the **loss of crucial information**, which hindered the performance of our models.

To overcome this limitation, we plan to adopt **deep learning techniques**, which are better equipped to handle complex, high-dimensional image data. These methods, combined with advanced augmentation strategies, are expected to significantly enhance the **accuracy** and **robustness** of our AI-generated image detection system.

References

[1] Detecting Fake Images Using Machine Learning. Available: https://ijrpr.com/uploads/V4ISSUE4/IJRPR11629.pdf

[2] Deep Learning for Image Authentication: A Comparative Study on Real and AI-Generated Image Classification. Available: https://www.researchgate.net/publication/375952278_Deep_Learning_for_Image_Authentication_A_Comparative_Study_on_Real_and_AI-Generated_Image_Classification

[3] Exploring and Analyzing Image Data with Python. Available: link

[4] SIFT (Scale-Invariant Feature Transform). Available: https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html

[5] HOG (Histogram of Oriented Gradients). Available: https://machinelearningmastery.com/opencv_hog/

[6] Local Binary Pattern (LBP). Available: link

[7] SVM (Support Vector Machine). Available: https://scikit-learn.org/1.5/modules/svm.html

[8] Random Forest. Available: https://scikit-learn.org/1.5/modules/svm.html