DeepDetect : Detect Fake and AI-generated Images
End-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 objective is to enhance the ability to discern genuine content from fabricated images, thereby reducing the potential for misinformation. We have explored various machine learning techniques, including Support Vector Machines (SVM), Random Forests, and Convolutional Neural Networks (CNNs), to classify images as real or fake and compared the effectiveness of traditional ML models and CNNs. Github link : `https://github.com/himangg/Machine-Learning-Project`

1. Introduction

1.1. Problem Statement and its Significance

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. Our project aims to develop a robust system capable of detecting AI-generated images, thereby helping to combat the spread of misinformation and fake content.

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

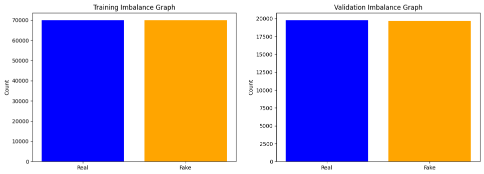
The effectiveness of deep learning models in differentiating real images from AI-generated ones is investigated in the paper ”Deep Learning for Image Authentication: A Comparative Study on Real and AI-Generated Image Classification” by Gaye Ediboglu Bartos and Serel Akyol. The research is centered around 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

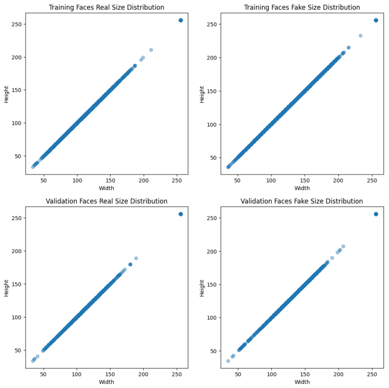
We have used this dataset.This dataset contains manipulated images and real images. The manipulated images are the faces which are created by various means. Each image is a 256 X 256 jpg image of human face either real or fake By using bar graphs, scatter plots, and violin plots, we analyze the similarities 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.

3.1. Visualization of Dataset

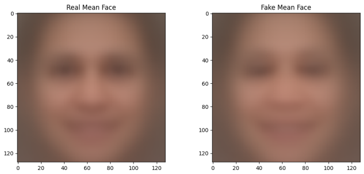
- **Class Imbalance Graphs:** In order to avoid overfitting, we analyzed the amount of images per category in both the training and validation sets. If there were a large imbalance in the number of images in a class, we would have conducted undersampling to achieve a more equal distribution in the dataset.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



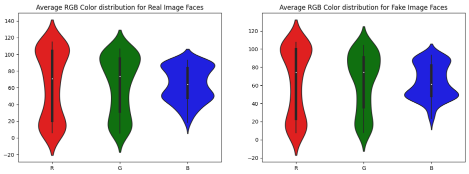
- **Scatter plot of face dimensions:** Faces were extracted from images for deepfake detection. A width vs height scatterplot was created but did not show a difference between real and fake image dimensions, suggesting it may not help distinguish them.



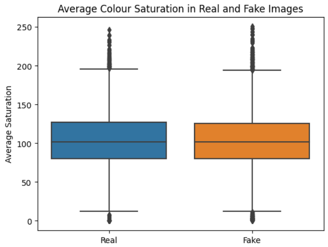
- **Mean Real and Fake Face images:** We created a visual representation of the average face extracted from both the real and fake image datasets in order to compare the mean face images of each type. We noticed distinct discrepancies since they did not resemble each other.



- **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

4.1. Classical Machine Learning

4.1.1 Pre-processing and Data Augmentation

Pre-processing and data augmentation improve model performance by standardizing data and expanding dataset.

4.1.2 Data Augmentation Techniques

- Spatial Transformations
- Image Quality Transformations through noise
- Color Transformations
- Mild Distortions

4.1.3 Methodology

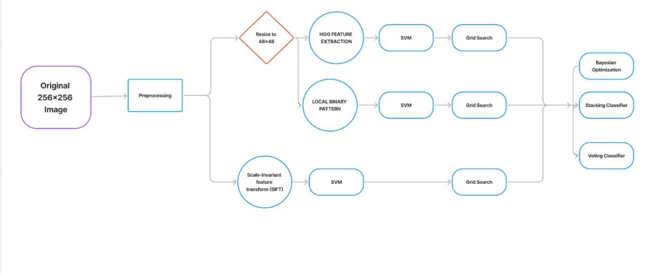


Figure 1. Work Flowchart

Our mid-semester project explored various feature extraction techniques, including LBP, SIFT, and HOG, to classify images. In this phase, we aim to combine these features by performing hyperparameter tuning(on a smaller sample than the dataset) using grid search and applying ensemble learning methods to enhance classification performance. This approach leverages the strengths of multiple models for improved accuracy.

The following hyperparameters were identified as the best according to the grid search algorithm:

- **SIFT**
  - Best Accuracy: 61.00%
  - Best SIFT Parameters: `n_features=256, contrast_threshold=0.06, edge_threshold=5`
  - Best SVM Parameters: `{'C': 0.1, 'kernel': 'linear'}`
- **HOG**

  - Best HOG Parameters: {'pixels_per_cell': (8, 8), 'cells_per_block': (3, 3), 'orientations': 9}
  - Validation Accuracy: 0.6892857142857143
- **LBP**
  - Best LBP Parameters: {'radius': 2, 'n_points': 8, 'method': 'uniform'}
  - Validation Accuracy: 0.6000

We applied ensemble learning to combine models trained on HOG, LBP, and SIFT features. The Voting Classifier uses majority voting, ensuring stability and leveraging the strengths of individual models. The Stacking Classifier uses a meta-classifier to combine predictions optimally, capturing inter-model relationships. These methods enhance classification accuracy and robustness by integrating diverse feature representations.

4.2. Convolutional Neural Networks (CNNs)

4.2.1 Preprocessing

We normalized the image input by dividing the pixel by 255; this converts the image pixel to a range of [0, 1]. This step is crucial as it could accelerate the convergence process. Additionally, we also resized the image to 128x128 to ensure uniformity.

4.2.2 Grid Search

We used a custom grid search to optimize our hyperparameters; some of the optimizers we used are Adam and RMSprop. We compared two learning rates (0.001 and 0.0001) and batch sizes (32 and 64). Each combination was evaluated to identify the optimal parameters that gave the best accuracy and minimum loss on the validation dataset. We applied the grid search on a smaller dataset as it would have consumed much time. This approach helped us to fine-tune our model and get optimal performance. Out of them, the best model was the {'batch_size': 32, 'learning_rate': 0.001, 'optimizer': 'rmsprop'}.

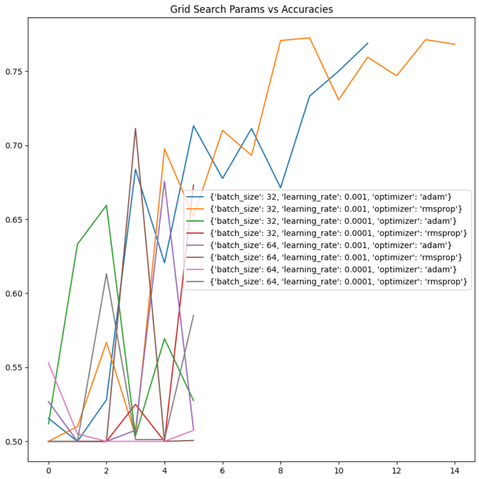


Figure 2. Work Flowchart

4.2.3 Methodology

The literature review conducted by us showed that one of the most promising ways to detect deepfake images has been through the use of Convolutional Neural Networks (CNNs). The inbuilt ability to learn spatial hierarchies and features makes it ideal for this task.

The CNN architecture we used consists of four convolutional layers (with increasing filter sizes of 32, 64, 64, and 128), each followed by batch normalization, max pooling, and dropout with 0.2 probability. The model ends with a dense layer of 256 neurons, and we have used a SoftMax output layer for binary classification. The use of dropout and batch normalization helps speed up the training, regularize, and prevent overfitting. The increasing number of filters in the convolutional layers helps capture progressively more complex features as the layers go deeper. A large dense layer of 256 neurons was used, allowing the model to learn more complex patterns.

We trained the model for 20 epochs with early stopping to prevent overfitting and capture the best performance on the validation set. The early stopping mechanism monitored the validation loss, with a patience of 6 epochs, triggering when the loss did not improve by more than 0.001.

5. Results and Analysis

5.1. Classical Machine Learning

Voting Classifier Test Accuracy: 0.72

Stacking Classifier Accuracy: 0.78

The results demonstrate that the Stacking Classifier achieved higher accuracy (78%) than the Voting Classifier (72%). This highlights the effectiveness of stacking in learning inter-model relationships and optimally combining predictions, as opposed to the more straightforward majority voting approach, thereby better utilizing the diverse feature sets.

5.2. Convolutional Neural Networks (CNNs)

Results

The CNN model trained on the best hyperparameters obtained from the grid search resulted in the following validation vs. training accuracy and loss graphs:

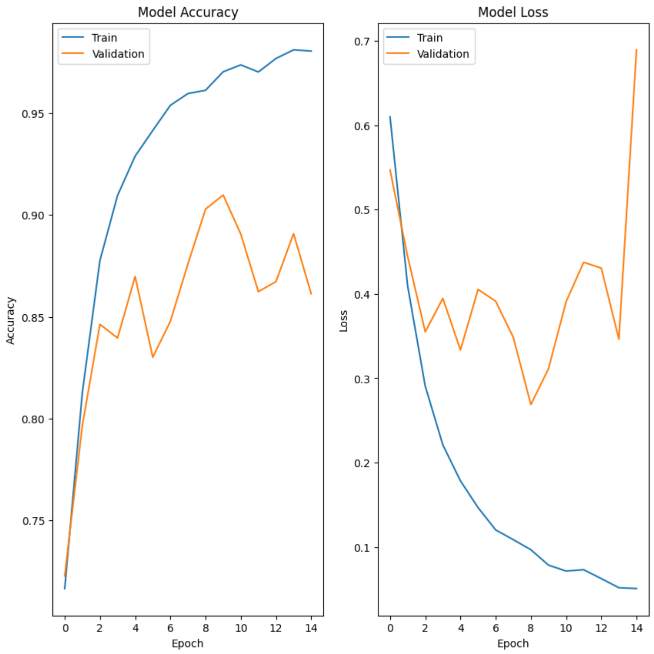


Figure 3. Work Flowchart

The model achieved a validation accuracy of approximately 90.28%, demonstrating its effectiveness in distinguishing between real and deepfake images. During training, the validation loss initially decreased, then fluctuated before stabilizing, prompting early stopping to prevent overfitting.

The final model was evaluated on the test set, where it also achieved an accuracy of 84.764%, confirming its strong generalization ability.

6. Conclusion

This work discussed the application of classical machine learning methods and CNN for image classification, in particular, detecting deepfakes. From this study, it was apparent that the different feature extraction techniques, HOG, LBP, SIFT combined with ensemble learning, do pretty well. Experimentally, it has been proved that the test accuracy by the Stacking Classifier at 78% is significantly better than the Voting Classifier at 72%. So, the inter-model relationship pays off to leverage it.

CNNs significantly outperformed classical approaches, achieving a validation accuracy of 90.28% and a test accuracy of 84.76%. The use of grid search for hyperparameter optimization and architectural elements like dropout, batch normalization, and early stopping contributed to improved performance and generalization.

Overall, the results emphasize the superiority of CNNs in deepfake detection task. We explored how well traditional could perform in such a field like image classification where models like CNNs dominate the field. We were able to achieve quite high accuracy results with the traditional models too displaying their capability in this field too.

6.1. contributions

- Himang Chandra Garg: Worked on the CNN model, data preprocessing, and hyperparameter tuning.
- Dasari Sai Harsh: Worked on the classical machine learning models, feature extraction, and ensemble learning.
- Nishil Agarwal: Worked on the dataset analysis, data visualization, and hyperparameter tuning.
- Piyush Narula: Worked on the dataset analysis, data visualization, and hyperparameter tuning.

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] Keras-cnn-tutorial. Available: https://victorzhou.com/blog/keras-cnn-tutorial

[9] Image-classification-using-convolutional-neural-networks. Available: https://www.analyticsvidhya.com/blog/2021/01/image-classification-using-convolutional-neura

[10] Building-and-using-a-convolutional-neural-network. Available: https://rohan09.medium.com/building-and-using-a-convolutional-neural-netw