Fake Face Detection Using Standard Machine Learning Models

Team MomentsYAR

Rohit Kumar[22264][1], Anoushka Pawan[22045][2], and Yash Choudhary[22379][3]

[1]Department of Physics, IISERB
[2]Department of Economic Sciences, IISERB
[3]Department of Physics, IISERB

Abstract

Advances in artificially altered media, especially by deepfakes, has raised substantial challenges for authenticity in digital images. This report addresses the issue of distinguishing between real and fake face images by applying standard machine learning models. Using a basic kaggle dataset containing both original and manipulated face images, we apply several algorithms, like decision trees, random forests, and support vector machines to classify these images correctly. We measure the effective-ness of each model by using metrics such as accuracy, precision, recall, and F1-score. Such classification reports provide valuable contributions to understanding the applicability of more traditional machine learning techniques toward enhancing fake face detection.

Code — https://colab.research.google.com/drive/1B70OUj2UzVmP7fpsv09r3TS-ISMv_iBl?usp=sharing

Datasets — https://www.kaggle.com/datasets/ciplab/real-and-fake-face-detection

Problem Statement

The dataset contains both real and fake face images. Use standard machine learning models to distinguish these two types of faces. Report and compare the performance of these methods.

Explanation The objective is to differentiate between genuine and synthetic face images utilizing machine learning models. The process consists of categorizing face images based on their authenticity, of which many algorithms analyze the patterns and other features that distinguish between two classes of face images. Model performance can be assessed using an accuracy metric or F1 score to determine which techniques work the best to distinguish between face images.

Data Specifications

Description
We are only provided a training Kaggle dataset. There are a total number of 2041 image files, of which 960 are fake/manipulated images and 1081 are real images.
Format : .JPG

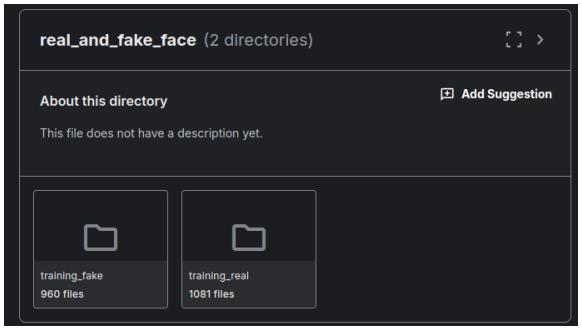

Figure 1: Kaggle Dataset

Target

Binary Classification

What is it?

It is a type of classification task that involves assigning one of two possible classes (or labels) to each input. In other words, the model's goal is to decide between two categories based on input data.

In the context of distinguishing between real and fake images, binary classification refers to the process of training a model to identify whether a given image is either:

- Real: Genuine images that have not been altered or manipulated.
- Fake: Images that have been generated, altered, or synthesized, often using techniques like deepfakes or other forms of image manipulation.

Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a crucial step in the data analysis process that focuses on summarizing and visualizing the main characteristics of a dataset. It helps us understand the data, detect anomalies, and assess data quality, guiding further analysis and modeling.

The key points include:-

- Data Summarization: Providing descriptive statistics to summarize the dataset's features and distribution.
- Visualization: Utilizing graphs like histograms, box plots, and scatter plots to identify patterns and relationships.
- Data Quality Assessment: Checking for missing values and inconsistencies to ensure reliability before modeling.

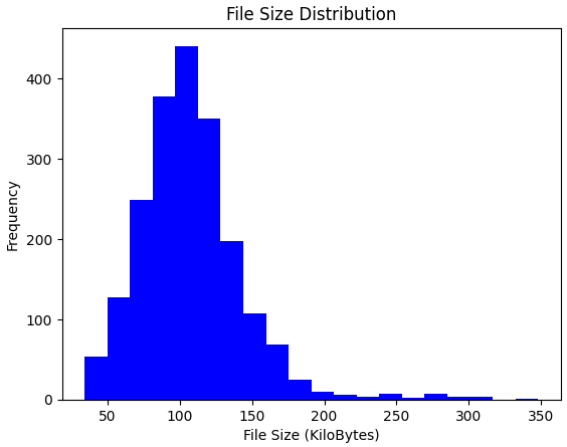

Figure 2: Image File Sizes

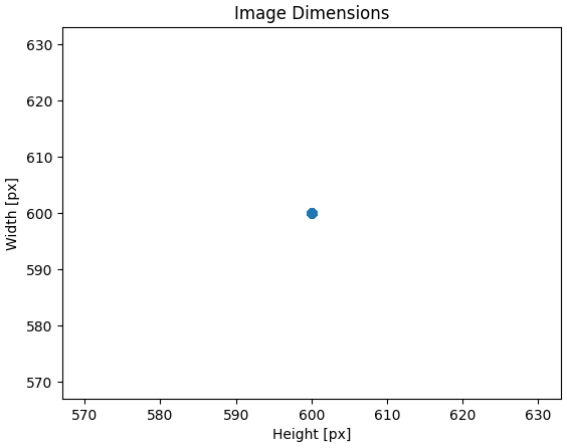

Figure 3: The Original Dimensions of Images

The above images have been extracted from the Python Code File, which was used to perform the aforementioned Exploratory Data Analysis.

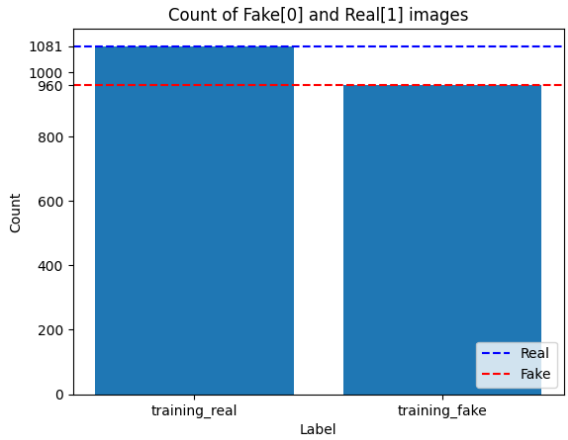

Figure 4: Count of Images

Encoding

Encoding refers to transforming data into a format that can be understood and processed by algorithms. In binary classification, encoding is often used to convert categorical or textual features into numerical values, since most machine learning algorithms require numerical input.

Types of Encoding

- Label Encoding : Label Encoding is a technique used to convert categorical variables into numerical values so that machine learning algorithms can process them. It provides a simple way to convert these categorical values into integers.
- One Hot Encoding : One-Hot Encoding is a technique used to convert categorical variables into a binary matrix where each unique category is represented by a vector of 0s and a single 1. It is useful when you have nominal categorical data.

Encoding Preference

Label Encoding

Why?

It is used in binary classification because it simplifies the process by converting labels into numerical values (1 and 0). This makes the data compatible with ML algorithms, which require numerical inputs while being efficient and straightforward.

Table 1: Dataset Classes

Class	Real	Fake
Label	1	0
Images Count	1081	960

Data Pre-processing

De-noising

Denoising removes noise from an image to improve quality. Common methods include Gaussian blur, which smooths the image by averaging pixel values. Denoising helps enhance feature extraction by removing irrelevant details.

Feature Extraction

- Local binary pattern(LBP) : It is a feature extraction technique that compares each pixel to its neighbours, generating a binary pattern. It is commonly used to capture texture features in images, making it effective for tasks like face and object recognition.

- Histogram of Oriented Gradients(HOG) : It extracts features by calculating gradient directions in cells, creating histograms of these directions, and normalizing them across blocks. It's widely used for object detection,especially for tasks like pedestrian recognition.

Literature Review

We went through a few papers and were able to find 1 paper(s) that helped us grasp the whole concept and made our minds clear about what we should do, further in our projects and what expectations to keep. The papers and their short review is mentioned below:

Paper 1

Classification of Real and Fake Human Faces Using Deep Learning

Author(s): Fatima Maher Salman and Samy S. Abu-Naser

Summary : The paper presents a deep learning approach to detect real and fake human faces using a dataset of 9,000 images, including real, mid-level, hard, and easy fake images. The primary model, ResNet50, achieved the highest performance with a training accuracy of 100% , validation accuracy of 99.18%, and testing accuracy of 99%. Several other models, including VGG16, MobileNet, and InceptionV3, were also evaluated, showing strong results, with InceptionV3 reaching a testing accuracy of 99%.

Key methods involved data preprocessing, image augmentation, and the use of convolutional neural networks (CNN). The study highlights the challenge posed by the growing use of GANs (Generative Adversarial Networks) to create fake images that can deceive even humans. The results demonstrate that deep learning models can effectively detect fake images, though further improvements can be made through larger datasets and more advanced machine learning techniques.

The research suggests that future studies should focus on dataset quality and accuracy as critical factors for the advancement of fake image detection technology.

Standard Machine Learning Models

Our aim was to train and test the various standard machine learning models that we wish to use to predict and apply for fake face detection.
Few of such ML models that we came across are mentioned below.

Logistic Regression

In reference to machine learning models, logistic regression is used for classification by modeling the probability that a given input belongs to a certain class. It works by applying a logistic function to a linear combination of the input features. The model outputs a value between 0 and 1, which is interpreted as the probability of the target class. Logistic regression is particularly effective for problems where the relationship between features and the target is approximately linear. However, its performance can be limited when the data exhibits more complex, nonlinear relationships.

- **How it works ?**: Logistic regression calculates a weighted sum of input features, passes it through a sigmoid function to predict a probability, and classifies the output based on a threshold (usually 0.5). During training, it adjusts the weights to minimize prediction error using optimization methods like gradient descent

- **Result**: Upon implementing logistic regression on our dataset performing a maximum of 1000 iterations , it performed quite well and gave us accuracy score of around 60.1%.

Random Forest

Random forests are an ensemble learning technique that combines multiple decision trees to improve model performance. Each tree is trained on random subsets of data and features. The predictions are aggregated, thus reducing variance and preventing overfitting. Hence, they improve generalization.

- **How it works ?**: Random Forest builds multiple decision trees, each trained on a unique subset of data and features (bagging). During prediction, each tree independently provides its result. For classification, the algorithm selects the class with the majority vote, and for regression, it averages the outputs. This approach reduces the impact of over-fitting and ensures robust and accurate predictions by leveraging the collective decision of diverse trees.

- **Result** : We took a total of 150 estimators while implementing the Random Forest(RF) classifier. we got around 62.3% accuracy.

K-Nearest Neighbors (KNN)

It is a simple, instance-based machine learning algorithm used for classification and regression tasks. KNN is widely used due to its simplicity and effectiveness, especially in scenarios where the decision boundary is not well-defined.

- **How it works ?** KNN classifies images based on the majority label of the nearest neighbors in feature space. For fake face detection, KNN can work if the extracted features cluster well into real and fake categories.It works by calculating the distance (usually Euclidean) between features of the test image and the training images.
- **Result**: Taking nearest neighbors to be 15 for KNN. we got around 62.3% accuracy.

Support Vector Machine(SVM)

SVMs are supervised machine learning model that classify data by finding the optimal hyperplane that separates different classes. For non-linearly separable data, SVMs use kernel functions to transform data into higher dimensions. They are effective in high-dimensional spaces.

- **How it works ?**: SVM works well for binary classification problems like fake vs. real face detection. It finds an optimal hyperplane that separates the two classes based on the features extracted from the images.
- **Result** : Among all standard MLMs, we noticed SVM outperforms by a small margin with accuracy score of 62.8%.

Table 2: Classification Report Table

Model	Accuracy	Precision	Recall	F1-Score
LR	0.602	0.586	0.618	0.601
RF	0.623	0.584	0.789	0.601
KNN	0.623	0.620	0.583	0.601
SVM	0.628	0.594	0.749	0.662

CNN Trial

To achieve greater accuracy and better classification results, we tried experimenting with CNN; A deep learning architecture primarily used for image recognition and classification tasks. It consists of convolutional layers that automatically learn features such as edges, textures, and shapes from raw images, pooling layers that reduce dimensionality while retaining essential information, and fully connected layers that make final predictions.

- **Result**: We applied CNN to our data set and got an accuracy score of almost 64%, which is better than the other thedatasets.

Experimental Results

The classifier comparison reveals strong overall performance across SVM, Random Forest, KNN, and Logistic Regression, with similar accuracy among all models.
KNN achieves the highest precision, effectively minimizing false positives, while Random Forest excels in recall, demonstrating strong ability to identify true positives. SVM also performs well in recall. In terms of F1 score, which balances precision and recall, Random Forest and SVM lead, reflecting their overall strength. KNN and Logistic Regression show competitive but slightly lower F1 scores. Overall, Random Forest and SVM stand out as the most consistent performers, making them effective choices for this classification task.
A better visualization is given below in figure 5,6,7 & 8

```
Performance of Logistic Regression on Test Set:
Accuracy: 0.6014669926650367
Precision: 0.5857142857142857
Recall: 0.6180904522613065
F1 Score: 0.6014669926650367
[[123  87]
 [ 76 123]]

--------------------------------------------------

Performance of Random Forest on Test Set:
Accuracy: 0.6234718826405868
Precision: 0.5836431226765799
Recall: 0.7889447236180904
F1 Score: 0.6709401709401709
[[ 98 112]
 [ 42 157]]

--------------------------------------------------
```

Figure 5: LR and RF

```
Performance of KNN on Test Set:
Accuracy: 0.6234718826405868
Precision: 0.6203208556149733
Recall: 0.5829145728643216
F1 Score: 0.6010362694300518
[[139  71]
 [ 83 116]]

--------------------------------------------------

Performance of SVM on Test Set:
Accuracy: 0.628361858190709
Precision: 0.5936254980079682
Recall: 0.7487437185929648
F1 Score: 0.6622222222222223
[[108 102]
 [ 50 149]]

--------------------------------------------------
```

Figure 6: KNN and SVM

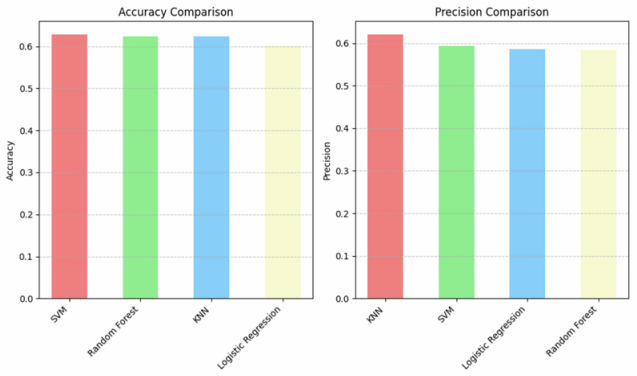

Figure 7: Accuracy and precision comparison

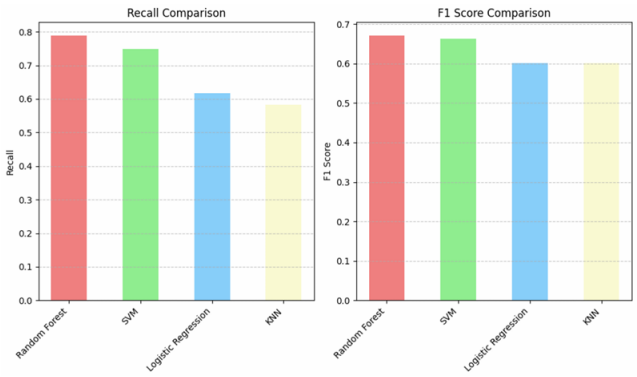

Figure 8: Recall and F1 score comparison

Conclusion

In this project, we aimed to distinguish between real and fake face images using standard machine learning algorithms and then compare the results. We explored various approaches, including

- **Logistic Regression (LR)**
- **Random Forest (RF)**
- **K-Nearest Neighbors (KNN)**
- **Support Vector Machine (SVM)**
- **Convolutional Neural Networks (CNN)**

The dataset was split into training and testing sets, and each model was evaluated based on key performance metrics such as accuracy, precision, recall, and F1 score.

Here's a summary of the key findings from each of the models tested:

Logistic Regression (LR): This model provided a basic linear classifier, which resulted in an accuracy of around 60%. While it was fast to train, its performance was relatively low compared to other models.

Random Forest (RF): This ensemble method performed better than Logistic Regression, achieving an accuracy of approximately 62%. The Random Forest classifier also showed promising results in terms of precision and recall but still lagged behind more complex models like SVM.

k-Nearest Neighbors (KNN): KNN achieved an accuracy of around 62%, with performance being dependent on the choice of distance metric and hyperparameters. While it provided reasonable performance, it suffered from higher computation times during testing, especially with larger datasets.

Support Vector Machine (SVM): Among the traditional machine learning models, SVM outperformed the rest with an accuracy of 63%. It demonstrated superior performance in distinguishing between real and fake faces, likely due to its ability to handle high-dimensional data efficiently and find the optimal decision boundary.

Convolutional Neural Networks (CNN): As expected, CNNs—designed specifically for image data—provided the best results, achieving an accuracy of 64%. This was slightly better than SVM and showed the advantage of deep learning when it comes to image classification tasks.

Final Thoughts:

- The SVM model was the best performing traditional machine learning model, achieving 63% accuracy. However, its performance was still slightly surpassed by the **CNN**, which reached an accuracy of 64%.
- Despite the modest improvement of CNN over SVM, it is clear that deep learning techniques are more suited for image classification tasks, particularly in scenarios involving complex image features like real vs. fake face detection.
- The results underscore the importance of leveraging specialized models like CNNs when dealing with image-based datasets. While traditional machine learning methods (such as SVM) can provide good baseline results, CNNs are likely to provide superior performance for image recognition tasks.

  In conclusion, after testing both traditional and deep learning models, the project shows that Convolutional Neural Networks (CNN) are the most effective approach for distinguishing real and fake face images, with a 64% accuracy.

Acknowledgments

Thanks to each of our friends and family for supporting us. The preparation of the LaTeX file that encompasses the report and research work was supported by **Rakshith V. Dogra**.

Thank you for reading the report. We look forward to receiving your recommendations!