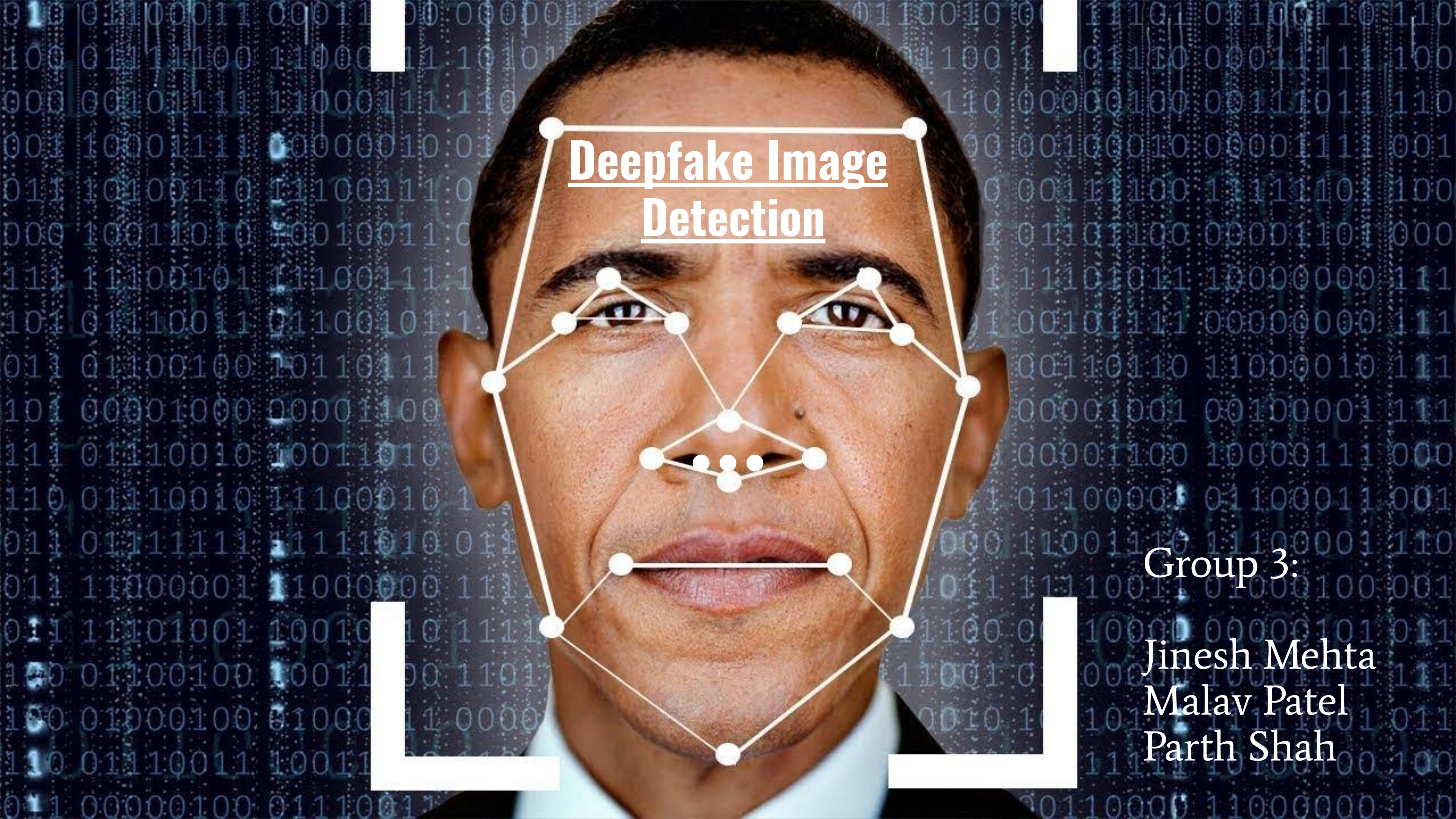

Deepfake Image Detection

Group 3:

Jinesh Mehta
Malav Patel
Parth Shah

Motivation

- Deepfake techniques, which present realistic AI-generated videos of people doing and saying fictitious things, have the potential to significantly influence how people assess the legitimacy of information presented online.
- Because deepfakes can be used maliciously as a source of misinformation, manipulation, harassment, and persuasion, these content generation, and modification technologies may have an impact on the quality of public discourse and the protection of human rights.
- The goal of this project is to create novel new technologies that can aid in the detection of deepfakes and manipulated media.

Description

- Our team will compare models for predicting whether a video/image is real or fake to find the one with the best performance.
- This project will provide a foundation for what model would work best for real/fake image/video classification, which can then be built upon by news organizations and social media platforms in the future to prevent the virality or misuse of deep fake content.

Dataset

- Medical image dataset from UCI Machine Learning Repository
  - The dataset is divided into two sets (80 scans and 20 scans). The first 80 scans were used in a blind trial with radiologists and the remaining 20 scans were used in an open trial with radiologists. A table with the ground truth is included with the scans
- Facial image dataset from Kaggle
  - Faces were detected from the first frame of videos using dlib and resized to 224x224 because many pre-trained models exist for this size. This accounts for roughly 80% of the training set. The remaining 20% had either no or multiple faces detected and ignored.
  - This arbitrary selection is also the median, implying that half of the faces are scaled up and half are scaled down. The original size was added as metadata because it affects blurriness and may affect classification.

Note: To improve the model's performance, we may add additional datasets, including video formats.

Main Idea

- Deepfake detection is normally deemed a binary classification problem where classifiers are used to classify between authentic images and tampered ones.
- The main objective of this project is to efficiently distinguish deepfake images from normal images.
- Most of the recent research uses a CNN-based strategy to identify deepfake images, while others used feature-based techniques.
- To detect the deepfake images, few of them used machine learning classifiers. But the novelty of this work is that it can detect deepfake images from normal images with 99% accuracy using the VGGFace model.

Methodologies [Architectures]

- To detect if the images are fake or not, we identified few architectures as listed below. These models are most used to address the problem of deepfakes:
  - VGGFace
    - Five blocks of the layer, with convolutional and max-pooling layers in each block.
    - Two 3×3 convolution layers followed by a pooling layer.
    - Three 3×3 convolution layers are followed by a max-pooling layer.
    - The ReLU activation function is employed in all convolutional layers.
    - Lastly, we fine-tuned the facial characteristics by adding dense layers as well as providing sigmoid activation function.
  - Custom CNN architecture
    - Six convolution layers (Conv2D) each paired with batch normalization, max pooling, and dropout layers
    - Rectified Linear Unit (ReLU) and sigmoid activation functions will be applied for the input and output layers respectively
    - Dropout will be applied to each layer to minimize over-fitting

Methodologies [Architectures]

- DenseNET architecture [Refer. from ‘Keras’s DenseNet-264 arch.]
    - 7 × 7 strides 2 convolutional layer
    - 3 × 3 strides 2 max pooling layer
    - 4 dense blocks paired with batch normalization and ReLU activation function for the input layers and sigmoid activation function for the output layer.
    - Between each dense block, there would be a 2 x 2 average pooling layer along with a 1 x 1 convolutional layer.
    - The last dense block is followed by a classification layer that leverages the feature maps of all layers of the network for the task of classification which will be coupled with a dense block with the sigmoid activation function as the output layer.

Note: We would start with VGGFace and Custom CNN architecture and at later stage of the project, we may implement DenseNet architecture and perform evaluation/analysis on that as well.

Performance and Evaluation Metrics

- **Accuracy**, simply put, indicates how close the model prediction is to the target or actual value (fake vs. real), meaning how many times the model was able to make a correct prediction among all the predictions it has made.
- **Precision**, on the other hand, refers to how consistent results are regardless of how close to the true value they are using the target label.
- The **Recall** is the proportion of actual positives that were identified by the model that were correct. The recall is intuitively the ability of the classifier to find all the positive samples.
- The **F1-score**, by taking into account both precision and recall, balances the precision and recall and indicates model ability to accurately predict both true-positive and true-negative classes.

Ablation Study

- We will be experimenting by changing multiple hyper parameters and while trying to get the best accuracy. Some of the aspects which we will be taking under consideration are as follows:
  - **Exploratory Data Analysis** : It will help us analyze the entire dataset and summarize its main characteristics, like class distribution, size distribution, and so on. Visual methods are often used to display the results of this analysis.
  - **Image Preprocessing** : Massive data can lead to lower estimation variance and hence better predictive performance. More data increases the probability that it contains useful information, which is advantageous.
  - **Color Channel** : We will take the raw image and improve image data (also known as image features) by suppressing unwanted distortions, resizing and/or enhancing important features, making the data more suited to the model and improving performance.
  - **Hyperparameter tuning** : Hyperparameter tuning libraries like Scikit learn Grid Search, Keras Tuner, and others that will try all hyperparameter combinations within the specified range, and it will return the best performing model.

- **Regularization** : This method forces the model to learn a meaningful and generalizable representation of the data by penalizing memorization/overfitting and underfitting, making the model more robust at dealing with data it has never seen before. It includes various techniques which are mentioned below:
    - Adding Dropout
    - Adding or changing the position of Batch Norm
    - Data augmentation
    - Mixup
    - Weight regularization
    - Gradient clipping

Timeline

- Week 9: Pre-processing and cleaning the data
- Week 10: Exploratory Data Analysis of the dataset
- Week 11: Training and Testing using different classifiers
- Week 12: Model Performance measurement and hyper-parameters tuning
- Week 13: Project Report and Presentation

Responsibilities

Overall all three members of the group will contribute roughly proportionally to this project. All the tasks will be discussed and looked over by each member.

The following is the initial task distribution:

- Pre-processing and Cleaning the data - Parth Shah
- Exploratory Data Analysis - Malav Patel
- Modeling and Performance metrics - Jinesh Mehta
- Documentation and Presentation - Everyone

Thank You!