Deepfake Image Detection

Group 3: Jinesh Mehta, Malav Patel, Parth Shah

1. Description:

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. Detecting manipulated media is a technically challenging and rapidly evolving challenge. The goal of this project is to create novel new technologies that can aid in the detection of deepfakes and manipulated media. Our team will compare models for predicting whether a video/image is real or fake to find the best one. 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.

2. Dataset:

We will use datasets from different domains, specifically facial image and medical image datasets from UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Deepfakes%3A+Medical+Image+Tamper+Detection) and Kaggle Dataset (https://www.kaggle.com/datasets/dagnelies/deepfake-faces). For the medical image dataset, there are medical deepfakes: 3D CT scans of human lungs, some of which have been tampered with real cancer removed and fake cancer injected. The goal of this dataset is to distinguish between real and fake cancers and to identify where medical scans have been tampered with. The dataset is divided into two sets (80 scans and 20 scans). The first 80 scans were used in a blind trial with radiologists (who were not told they had been tampered with), and the remaining 20 scans were used in an open trial with radiologists (they were told the truth and asked to identify them). A table with the ground truth is included with the scans. For each scan, the location of cancer (x, y, and z [slice#]) and its classification. A location can be classified as being: True-Benign, (TB): A location that has no cancer, True-Malicious (TM): A location that has real cancer, False-Benign (FB): A location that has real cancer, but it was removed, and False-Malicious (FM): A location that does not have cancer, but fake cancer was injected there. Each scan is in the medical Dicom format, but it can be loaded into Python as a 3D matrix. A scan consists of 512x512 images. The series is typically 100-300 slices long (the z-axis). Cancers can take up several slices along the z-axis. Each pixel's value represents the Hounsfield unit (radiodensity) at that location.

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. The faces were extracted using Python's face recognition package, which uses dlib behind the scenes. 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.

3. Main Idea, Different Models & Evaluation Metrics:

- 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.
- 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.
    - 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
  - DenseNET architecture [Refer. from 'Keras's DenseNet-264 arch.]
    - 7 × 7 strides 2 convolutional layer
    - 3 × 3 strides 2 MaxPooling 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 VGG-19 and Custom CNN arch. At later stage of the project, we may implement DenseNet arch. and perform evaluation/analysis on that as well.

- Evaluation Metrics:
  - Accuracy
  - Precision
  - Recall
  - F1-score

4. Related Work/Literature Review:

1. Kumar, A. (2021, September 1). *How to Create & Detect Deepfakes Using Deep Learning*. Data Analytics. Retrieved October 30, 2022, from https://vitalflux.com/how-to-create-detect-deepfakes-deep-learning/
2. Taeb, M., & Chi, H. (2022). Comparison of Deepfake Detection Techniques through Deep Learning. *Journal of Cybersecurity and Privacy*, *2*(2), 89-106. https://www.mdpi.com/2624-800X/2/1/7/htm

5. Experiments and Ablation Settings:

We will be experimenting by changing multiple hyperparameters while trying to get the best accuracy.

1. **Exploratory Data Analysis**
2. **Add more data**
3. **Image Preprocessing**
4. **Color Channel**
5. **Hyperparameter tuning**
6. **Regularization**

6. Proposed Timeline:

Week 9 - pre-process the data and perform the cleaning of the dataset
Week 10 - perform exploratory data analysis on the dataset
Week 11 - training/testing different classification techniques on the dataset
Week 12 - measuring the performance of the models and tuning the hyperparameters
Week 13 - preparing the project report and final presentation

7. Responsibilities:

Overall, all three group members 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 data - Parth Shah
- Exploratory Data Analysis - Malav Patel
- Modeling and performance metrics - Jinesh Mehta
- Documentation and presentation - Everyone