DeepFake Detection using CNN

David Johnson

Advisor: Dr. Kaushik Roy

Department of Computer Science

North Carolina A&T State University

Agenda

- Introduction
- Related Work
- Dataset Used
- Methodology
- Results
- Future Work

Introduction

- Growing concern of faked videos used for propaganda
- Sometimes easy to detect, subjectively (some people may not notice a deepfake while others may)
- By using large sets of data to compare real to fake, potential to detect fake videos

Introduction

- With a large enough dataset, and adequate preprocessing, a binary classification CNN could be built to detect features in images that can distinguish between original and manipulated images
- There are popular sources of deepfakes, or people to deepfake, such as Nicolas Cage or Vladimir Putin. While many of them are for comedic purposes, they still demonstrate the effectiveness of deep neural networks to create realistic, fake videos
- The rising popularity of deepfakes presents itself with a somewhat easily accessible source of original content, and manipulated content, to better train a CNN

Related Work

Huaxiao Mo, Bolin Chen, and Weiqi Luo. 2018. Fake Faces Identification via Convolutional Neural Network. In IH&MMSec ’18: 6th ACM Workshop on Information Hiding and Multimedia Security, June 20–22, 2018, Innsbruck, Austria. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3206004. 3206009

- Purpose:
    - To construct a simple CNN to detect fake faces. By using 30k images from the CELEBA-HQ dataset, train a model to distinguish real and fake celebrity faces.
- Result:
    - A model that could achieve over 99% accuracy on the dataset
- This would be useful since the current goal is to have a baseline CNN that can effectively distinguish original and manipulated frames

Related Work

S. Burroughs,K. Roy ,B. Gokaraju, and Khoa Luu, "Detection Analysis of DeepFake Technology by Reverse Engineering Approach (DREA) of Feature Matching, " International Conference on Machine Intelligence and Smart Systems 2020 (MISS-2K20, Springer in Algorithm for Intelligent System (AIS) book series.

- Purpose:
  - Using SIFT to provide facial key points as inputs for CNN training. The dataset used was FaceForensics++\DeepFake
- Result:
  - Using SIFT over a period of fifty epochs allowed accuracies around 93%, compared to just using CNN with slightly lower accuracy
- This would be useful since both areas of research will be using the FaceForensics++ collections of datasets, yet instead of DeepFake, DeepFakeDetection will be used. Unfortunately, the way the CNN was trained was not included, although that was not the goal of that paper

Related Work

K. Zhang, Z. Zhang, Z. Li and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," in IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499-1503, Oct. 2016, doi: 10.1109/LSP.2016.2603342.

- Purpose:
  - The construction of a CNN model to detect and align faces in an image, and to also locate facial landmarks, such as eyes, nose, and mouth.
- Result:
  - MTCNN that can detect and locate faces and facial landmarks with an accuracy of about 90%
- MTCNN would work better as a filter for the input dataset to detect faces and important facial features. Compared to OpenCV, MTCNN rejects much more false negative faces than OpenCV while also providing facial features to pass to a CNN

Related Work

D. Wodago, S. Atnafu, "Deepfake Video Detection Using Convolutional Vision Transformer," arXiv:2102.11126.

- Purpose:
  - Create a model that outputs learnable features to a Vision Transformer. Feature Learning does not have a fully connected layer, like CNN, and is meant for feature detection instead of classification
- Result:
  - A model that classifies videos as real or fake with an accuracy of 91.5% by focusing on learned facial features and also reviewing extracted faces that contain non-faces.
- This paper's dataset comes from the same FaceForensics++ collection. While using a set with different manipulation techniques and the same face detection methods, this paper would be a good guide to building a base CNN to detect facial features.

Related Work

A. Karandikar, V. Deshpande, S. Singh, S. Nagbhidkar, S. Agrawal, "Deepfake Video Detection Using Convolutional Neural Network, " International Journal of Advanced Trends in Computer Science and Engineering 2020. doi: 10.30534/ijatcse/2020/62922020.

- Purpose:
  - To create a CNN to classify original and manipulated frames based on facial feature patterns, or the inconsistent lack thereof, and video compression patterns. Specifically, the model was trained to focus on discrepancies around the face.
- Result:
  - A CNN model was created using transfer learning to use extracted facial features as input. Model accuracy achieved roughly 70%.
- This would be useful since both areas of research will be using extracted facial features as input to then train a CNN. This paper also is the closest to a simplified, base CNN.

Goal

- Detecting DeepFake videos
    1. Extract frames from videos
        - Detect face in frames
        - Extract facial features
    2. Train a model to compare facial features from real/fake frames

    1. Accurately classify real/fake frames

    1. Classify video without needing to manually test each frame per run

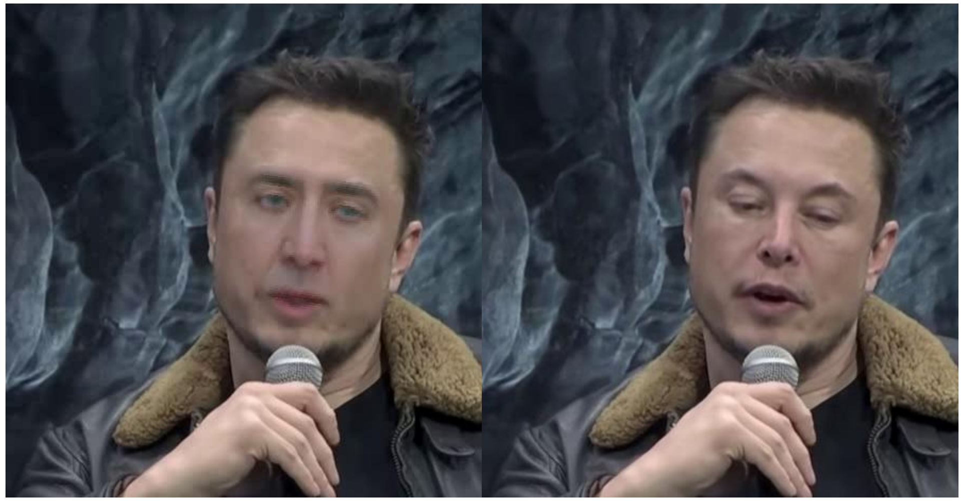
Dataset

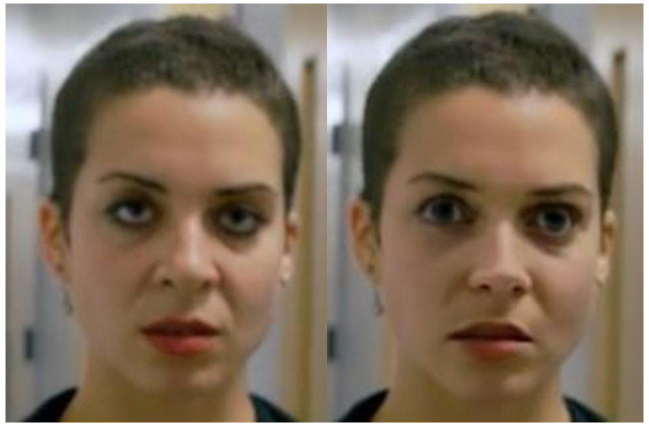
- UADFV dataset
  - https://yuezunli.github.io/
  - A collection of videos containing 49 real and 49 fake videos of world leaders and celebrities
    - Fake videos are generated from the real videos
- FaceForensics/FaceForensics++
  - https://github.com/ondyari/FaceForensics
  - Collection of fake videos using different methods from a set of 1000 original videos
  - DeepFakeDetection, Deepfakes, Face2Face, FaceShifter, FaceSwap, NeuralTextures

Tools

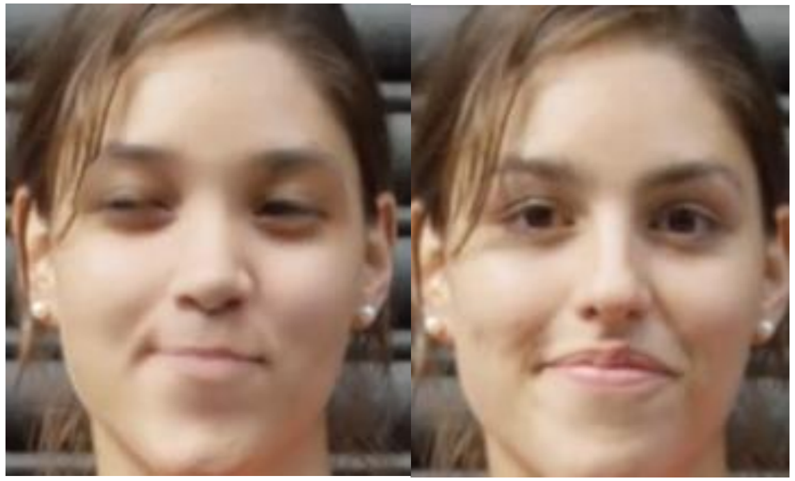
- Tensorflow.keras to create a CNN for classification
  - https://www.tensorflow.org/
- Datasets
  - UADFV
    - https://yuezunli.github.io/
  - FaceForensics++
    - https://github.com/ondyari/FaceForensics
- OpenCV to extract frames
  - https://opencv.org/
- MTCNN for face detection, alignment, and facial landmarks
  - https://kpzhang93.github.io/MTCNN_face_detection_alignment/paper/spl.pdf

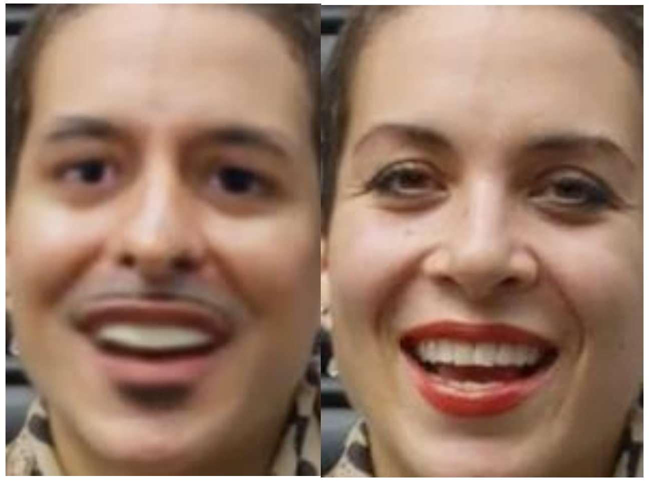
Example of Fake vs Real Frames (from UADFV/DeepFakeDetection datasets)





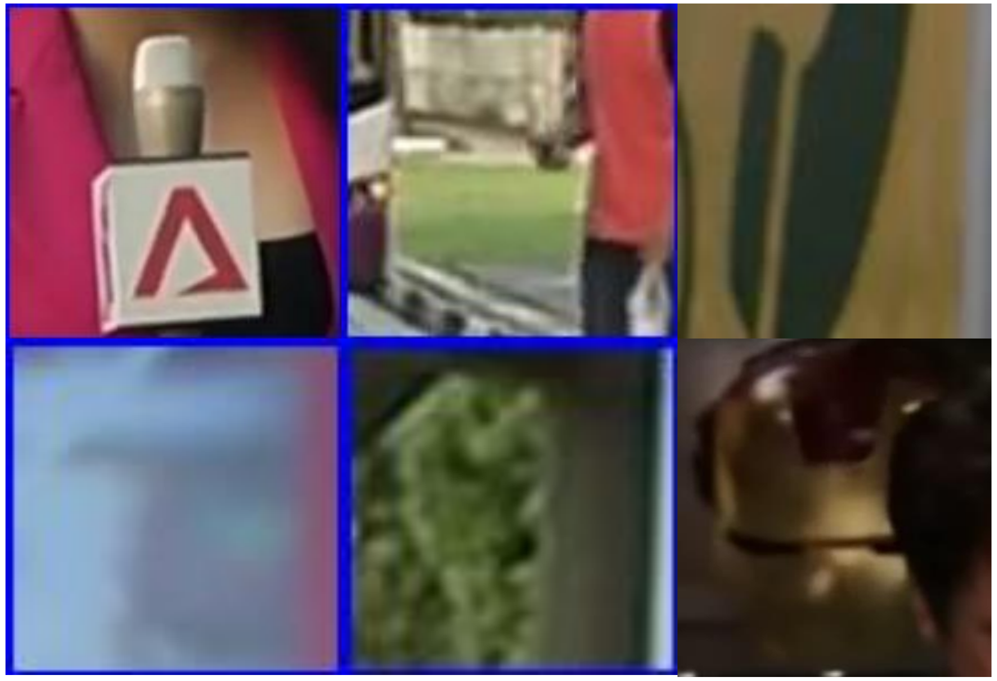
Left face is manipulated. Right face is original.





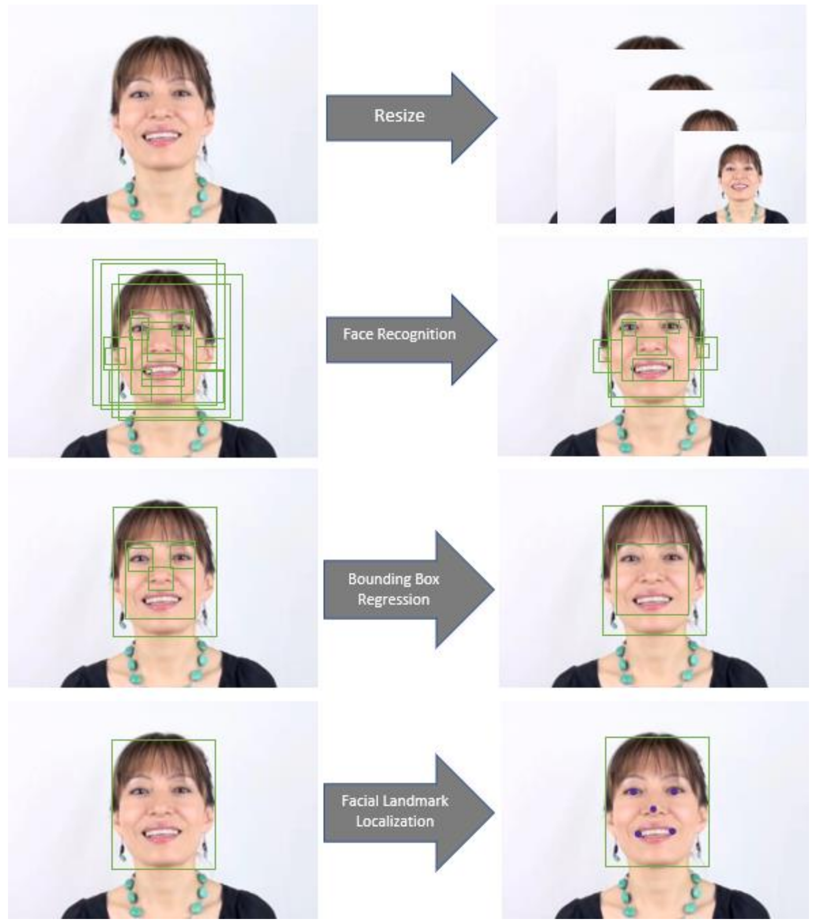
PreProcessing (OpenCV from UADFV/DeepFakeDetection datasets)

- False extracted “face” from OpenCV



- True extracted face from OpenCV

PreProcessing (MTCNN face detection and alignment)



P-Net used for face recognition to feed…

R-Net used for bounding box regression to feed…

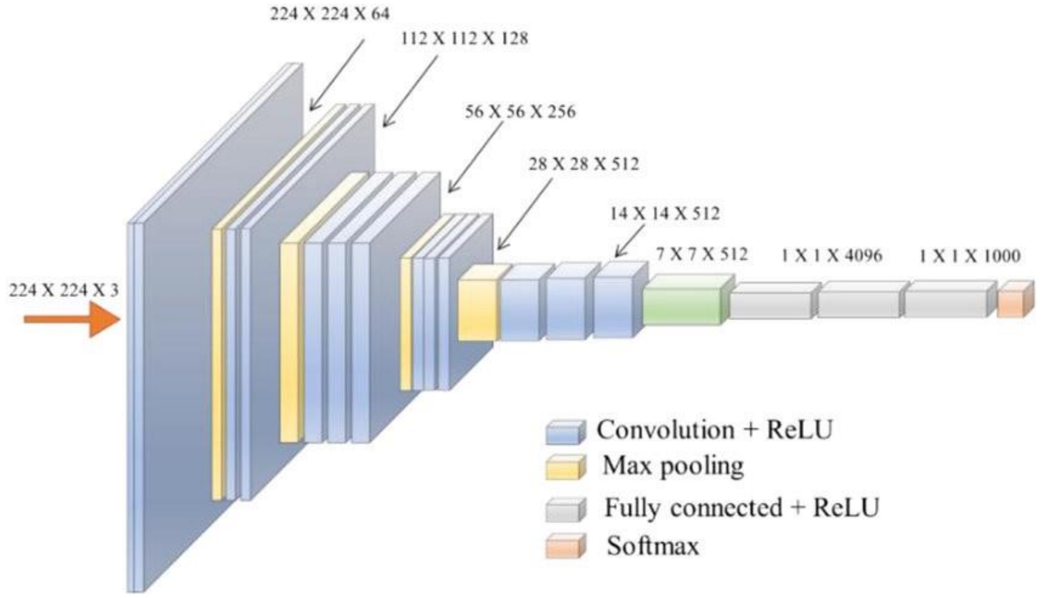
O-Net used for facial landmark localization

CNN Model

- VGGNet is a 2D convolutional layer, with an *n* x *n* filter, where *n* is odd, and finally a max pooling layer

- Blocks can be added sequentially, typically by doubling the depth with each block
  - e.g., 32, 64, 128, 256, 512, …

- Number of layers should adjust with complexity of distinguishing features



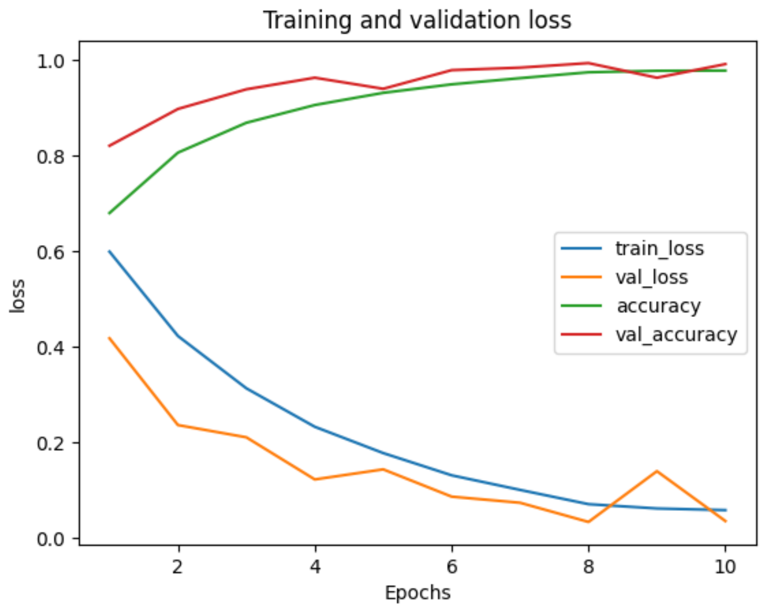
- Pandiyan, Vigneashwara et al. "In-process virtual verification of weld seam removal in robotic abrasive belt grinding process using deep learning." Robotics and Computer-Integrated Manufacturing (2019): n. pag.

Experimental Setup

- Using a subset of the DeepFakeDetection dataset, 10,000 images are used as input to fit the model.

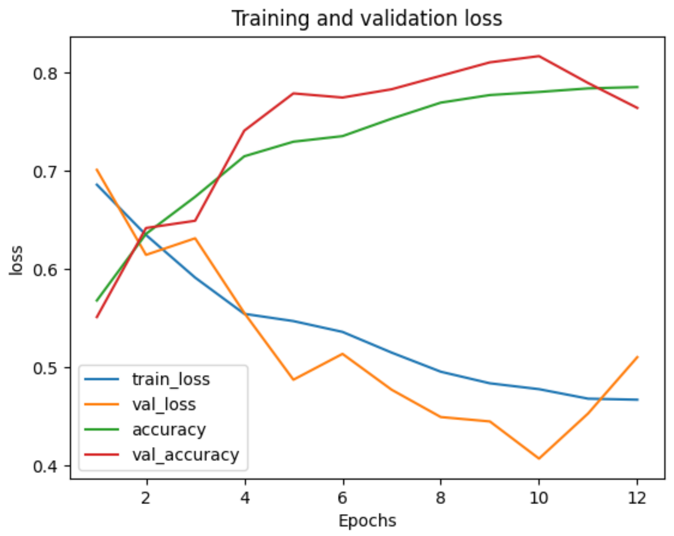
80/20 Training/Testing

- 8000 images for training, 2000 for testing.



90/10 Training/Testing

- 9000 images for training, 1000 for testing.

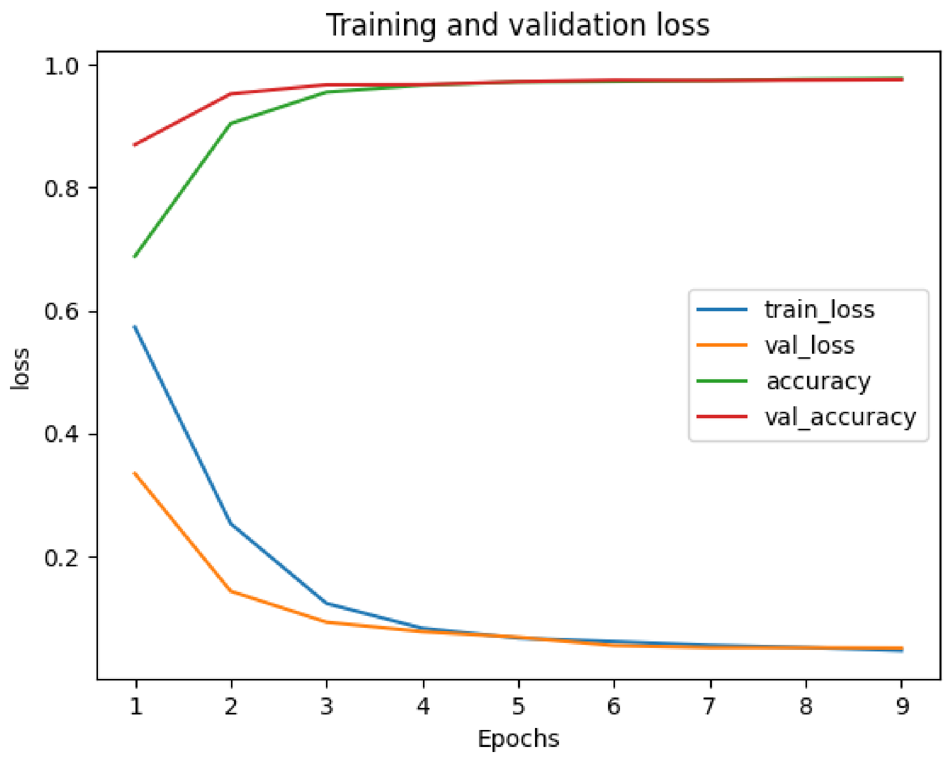


Experimental Setup

- Images are resized to 224 x 224 for training and prediction.
- Number of epochs varies between 10 and 15, by using EarlyStopping, stop training when validation loss is not longer decreasing between epochs
- Learning rate is initially 0.0001, by using ReduceLROnPlateau lower learning rate when validation loss is no longer decreasing between epochs
- Batch size is 32

Preliminary Results

UADFV/DeepFakeDetection combined datasets

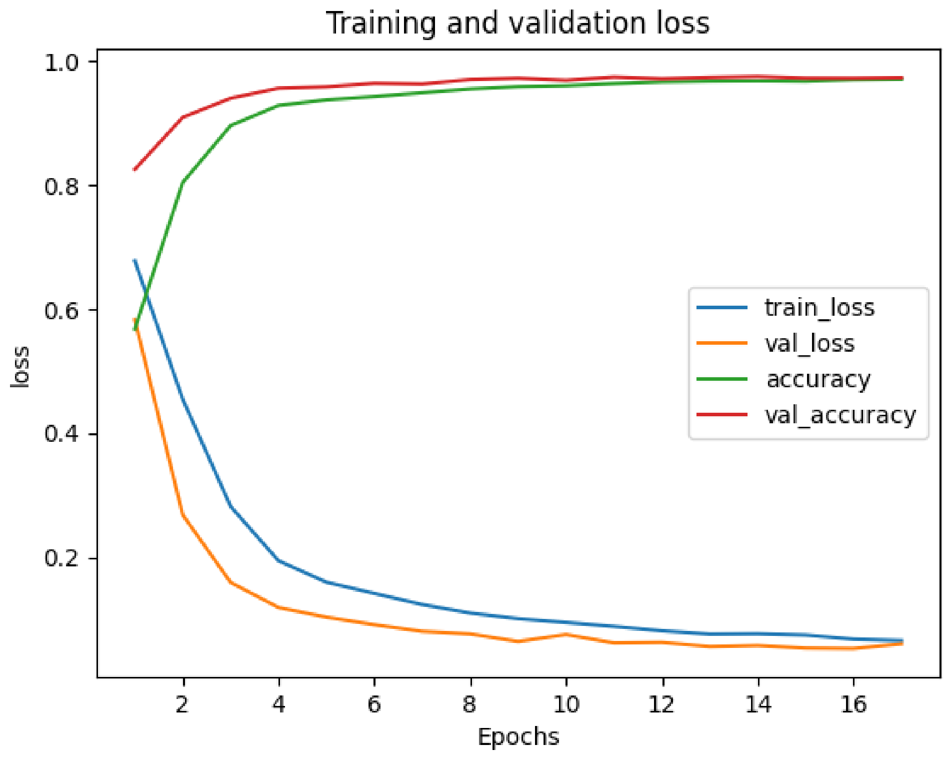


- OpenCV for face detection.
  - After manually going through the extracted frames, many of the extracted "faces" were not faces, neither did they look like faces. OpenCV's face detection should be further adjusted
- Highest perceived accuracy on the test data of 91.2%
- MTCNN for face detection
  - To learn distinct features for binary classification

Image Data Augmentation

- The idea is to enlarge the training dataset by created “new,” modified images of the dataset. Images can be shifted horizontally or vertically, rotated, or flipped
- When successful, overfitting should be reduced, along with val_loss. Accuracy would also increase
- Image Data Augmentation will most likely be needed because of the size and (lack of) diversity of the dataset
- val_loss decreased, accuracy increased
- Classification accuracy is becoming very slightly increased, still not correct
- UADFV/DeepFakeDetection combined datasets

Preliminary Results: Image Data Augmentation



- Roughly same results as the previous
- Highest perceived accuracy on the test data of 91.2%
- Augmentation barely helped but *did* allow some change in prediction
- Trained using faces extracted with OpenCV instead of MTCNN.

Current Problems

- Recreating set of extracted faces

- MTCNN is slow but accurate
  - Concept of FastMTCNN - consider a video with ten frames (f0, f1, …, f9):
    - Rectangle around detected face in f_n
    - Reuse rectangle in f_{n+1} and f_{n+2}
    - Rectangle around detected face in f_{n+3}
    - Repeat
- OpenCV is fast but inaccurate. Adjusting haar cascade classifier allows more accurate results at the expense of speed. Figured using a trained CNN would be better

Future Work

- Use Image Data Augmentation to expand the dataset (many of the images are very similar)
- Prediction/classification
  - Currently, test models have accuracy between 50-60%
- Use transfer learning with MTCNN as a filter to use output facial features as input for training new CNN model
  - Notably the eyes, region around the eyes, and mouth
- Look into Patch-Based CNN
  - Using more specialized models to compare images that hardly show differences

Acknowledgements

- National Security Agency
- Center for Cyber Defense, NCAT

Thank you

- Q&A