Deepfakes – CS 599:
Final Report

Project Members:

Adarsh Gopalakrishnan (7744347704)

Niveditha Kumaran (2196665393)

Abstract

Over the past decade, there have been multiple advances in deep learning for the application and synthesis of deep fakes. In this report, we investigate the two main approaches available under generative adversarial networks(GANs) and autoencoders and implement a novel approach of a self-attention GAN, which utilizes an autoencoder as the generative part of a GAN. This approach is tried on image and binary mask data sets of multiple personalities generated by a multi-task cascading convolutional neural network(MTCNN) using input videos and the results are observed for realness and accuracy along with other factors such as angle of face, lighting and number of possible positions captured. The training of the model and generation of videos is done on Google Colab GPU instance with Google Drive.

Introduction

Deepfakes, which originated from the combination of deep learning to generate fake images or videos, refers to the methodologies for generating and modifying images and videos using superimposition of existing data with machine learning and deep learning techniques.

E.g. The actor, Bill Hader's face being replaced with faces of multiple other personalities, Arnold Schwarzenegger, Tom Cruise and Al Pacino, respectively.

As deep learning networks grow more and more sophisticated, ability to train and utilize deep learning networks in a non-academic setting is growing easier by the year. Although deepfakes can be used or generated for illegitimate or nefarious purposes, the goal of this project is to explore multiple architectures available in the research community and to implement them for comedic purposes.

Objectives

The goal of this project is to explore different possible implementations of deep learning networks for deepfakes, specifically two types of architectures. Finally, given two short videos of two difference personalities speaking as input, the network should be able to detect the corresponding facial features frame by frame and swap the underlying structure to output the modified target video with the face swapped.

To achieve this goal, there are multiple steps involved , utilizing machine and deep learning, to allow us to reach the goal. They are as follows:

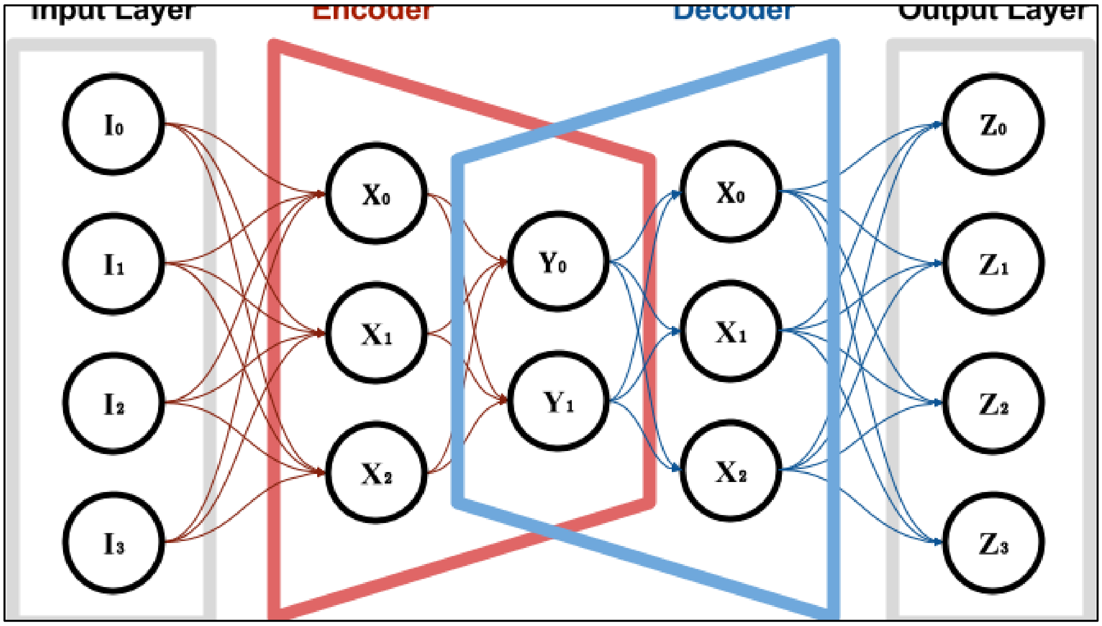
- Extract faces from videos
- Pre-process images for facial features
- Build architecture required for training
- Train model to generate swapped faces
- Convert images to output video

Related Work

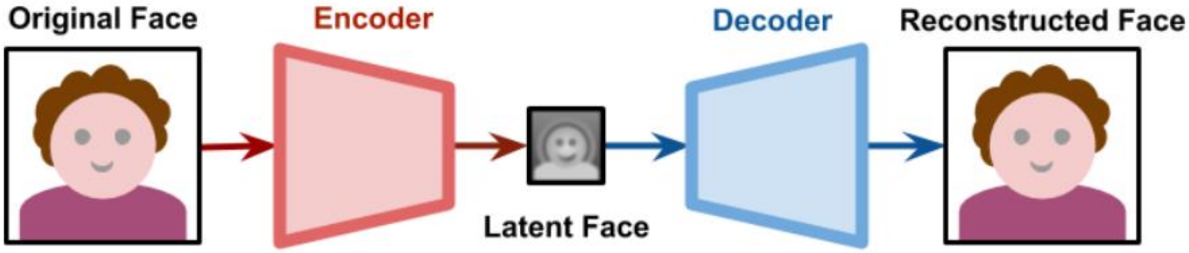
The main objective of the architecture is to provide image-image translation from two different faces. This involves generating and superimposing faces which can be done by multiple deep learning networks. In particular, we will be focusing on autoencoders and generative adversarial networks.

a. Autoencoders

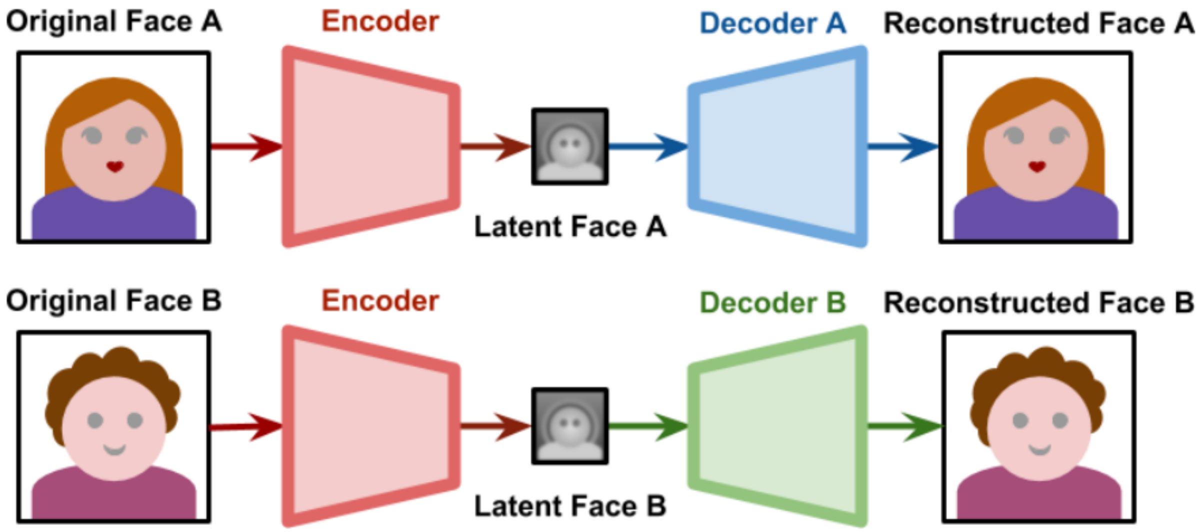
Autoencoders are artificial neural networks(ANNs) which consists of an encoder and a decoder. The input goes to the encoder which results in a lower dimensional representation of the image as seen in the image. The encoder is constructed such that the number of nodes in progressing layers decreases and therefore results in capturing the latent features. This middle layer is latent space representation of the input image.



The decoder then takes the output of the encoder and reconstructs the input image using the latent features. This is a lossy process, but allows the network to learn the latent features, specifically the facial features which we will need for replacement.

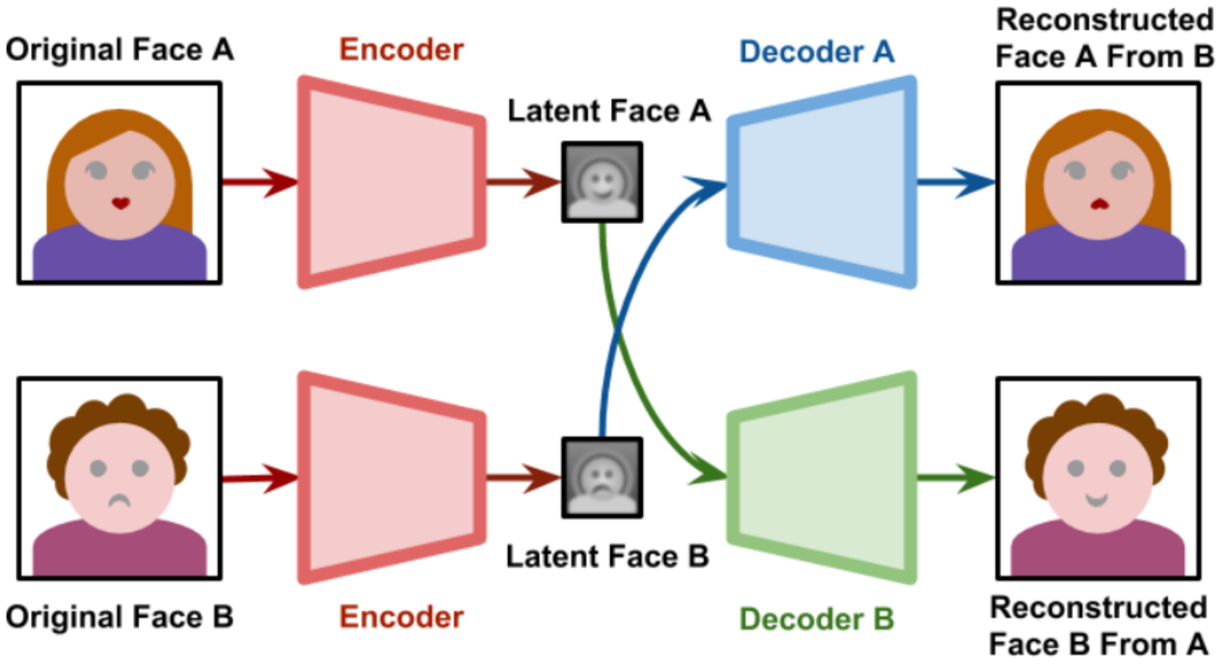


Autoencoders are utilized in deepfakes by training the same encoder on all possible input faces and learn the specific latent features. The autoencoders are trained so that all faces share the same encoder but difference decoders. This ensures that during the training phase, the encoder is able to learn the facial features and the specific decoders are able to construct the respective faces.



Training Phase for Autoencoders

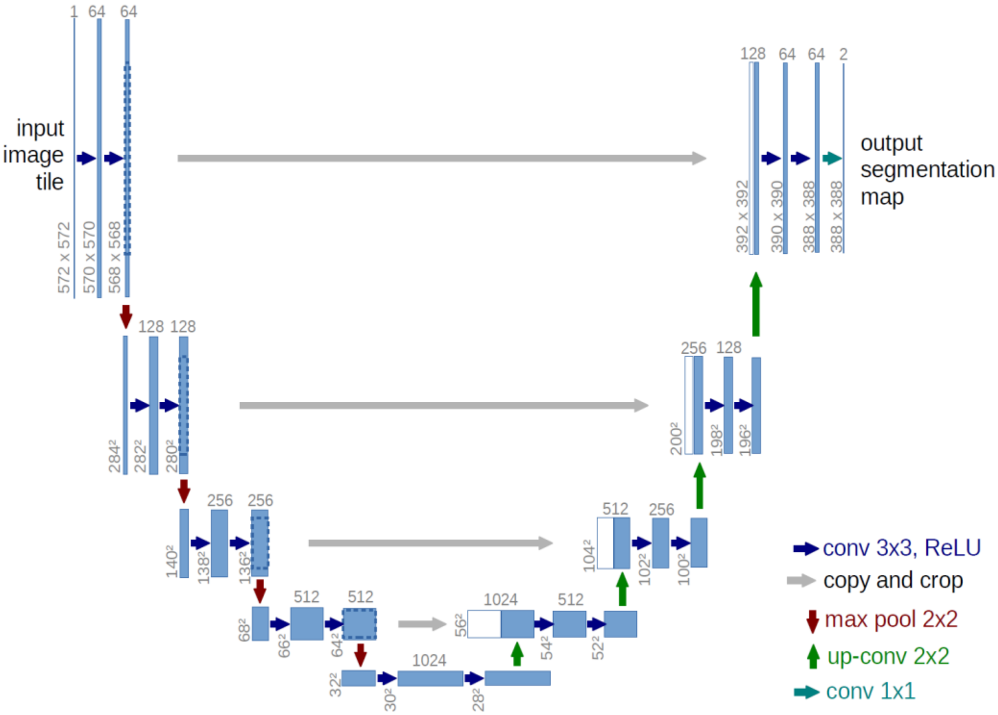
And during the testing phase, the autoencoders can be used to generate deepfakes by passing the face A , resulting in latent face A and the reconstructed by Decoder B and vice versa.



Testing Phase for Autoencoders

b. Generative Adversarial Networks (GANs)

Generative Adversarial Networks or GANs for short, are a specific type of deep learning network consisting of two networks, namely the generator network and the discriminator network. The role of the generator network is to generate new data instances through a deep learning network , for example, a U-network which utilizes convolution followed by de-convolution with the goal of generating passable target faces without being caught.



The discriminator network then evaluates the authenticity of output images from the generator network and decides whether each instance of data it reviews, belongs to the actual training dataset or not.

In terms of deepfakes, the generator takes in face of person A and returns a B-like image as part of a semi-supervised learning process. This generated B like image is fed into the discriminator alongside a stream of real B images taken from the actual dataset. The discriminator takes in both the real and fake images and returns the probabilities, with 1 representing a true image and 0 representing a fake image.

The generator network and discriminator network compete against each other to minimize the loss and moving towards the goal of generating images indistinguishable from real images.

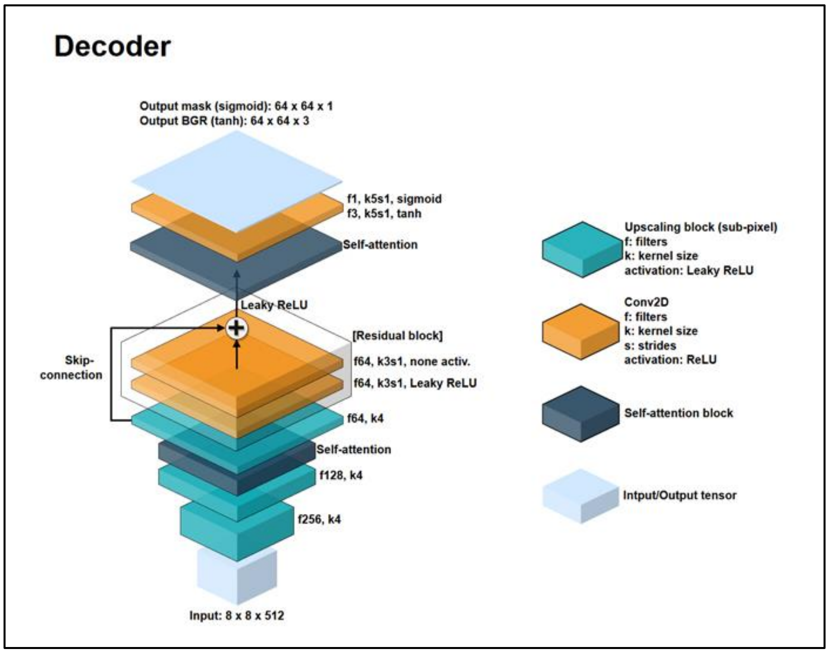
Methodology - Combination of GANs and Autoencoders

STAGE1

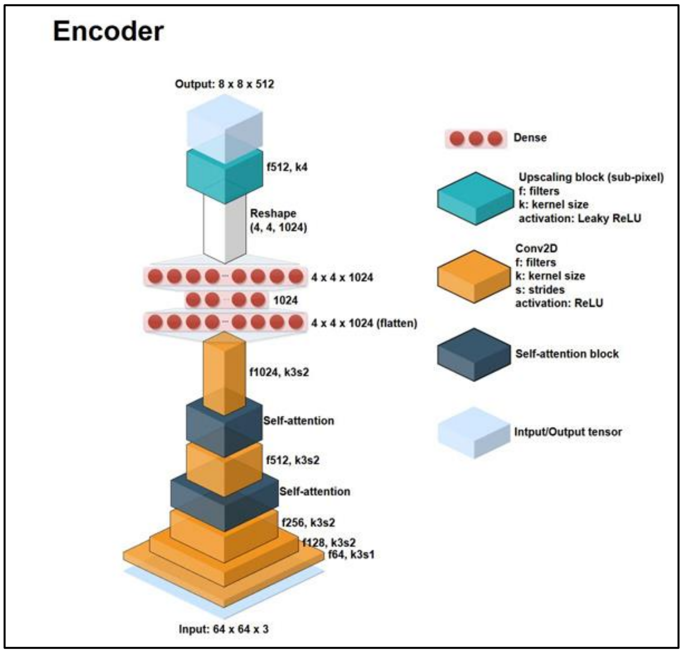
The source and target video, or video A and video B, are defined and given as input in .mp4 format. The input image and corresponding binary mask dataset is generated using a multi task cascading convolutional neural network (MTCNN). This is used on the source and the target video to generate the image and binary mask dataset for face A as well as for face B. These are used as input datasets for training the model which will swap the faces. Transfer learning is utilized for the MTCNN where the weights are imported from a trained model of a Github repository.

STAGE 2

A combination of a generative adversarial network and autoencoders is tried to take the best of both networks. In this combined architecture called a Self-Attention Generative Adversarial Network (SAGAN), we utilize the autoencoder as the generative part of the GAN, thereby utilizing the latent space representation for more efficient generation of images. This would be the final architecture for training on the final image dataset generated from the input videos.



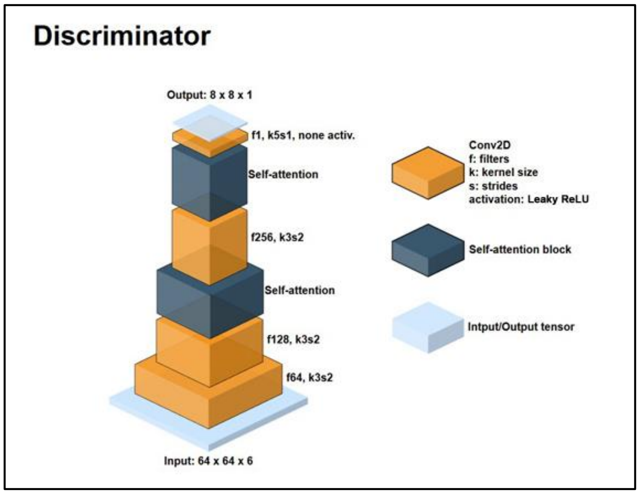
The discriminator network serves as the counterpart to the autoencoder to generate the output images.



Refining Objectives for methodology

Expanding on the objectives of the project, we utilize various optimized techniques and deep learning networks to improve the output at each stage of the process.

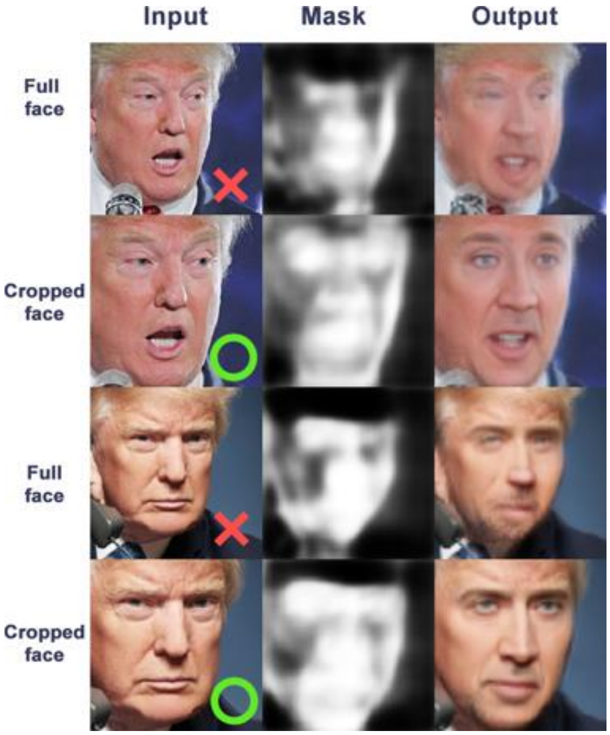
a. Extract faces from videos



When extracting the faces from the input video for training the architecture, it is important to recognize the importance of face alignment and cropping. Using a sophisticated CNN such as Multi-task Cascaded Convolutional Networks (MTCNN) allows us to extract images within a set number of frames and utilize VGGFace for removal of any noise so as to crop the face as much as possible which allows the deep learning network to identify the facial features better and allow for more precision, thereby looking smoother in the final output video.

b. Pre-process images for facial features

Face tracking and alignment with Kalman Filter in combination with MTCNN allows for stable detection of the face and consistent face alignment. The Kalman filter smoothens the boundaries of the frame positions and helps to remove jitter on the resulting face swap image.



c. Build combined architecture required for training

As covered earlier, building a combined architecture such as SAGAN which utilizes both an autoencoder network as well as a GAN structure allows the combined model to generalize better and take the best of both networks when training.

Tools and Dataset Specifications

a. Dataset
   - Video resolution 720p of any format
   - Video duration 30secs- 90secs RGB version
   - Image Resolution 64x64 RGB version
   - Binary Masks for every RGB image

b. Tools
   - Keras
   - Tensorflow
   - Python 3.6
   - Google Colab GPU instance

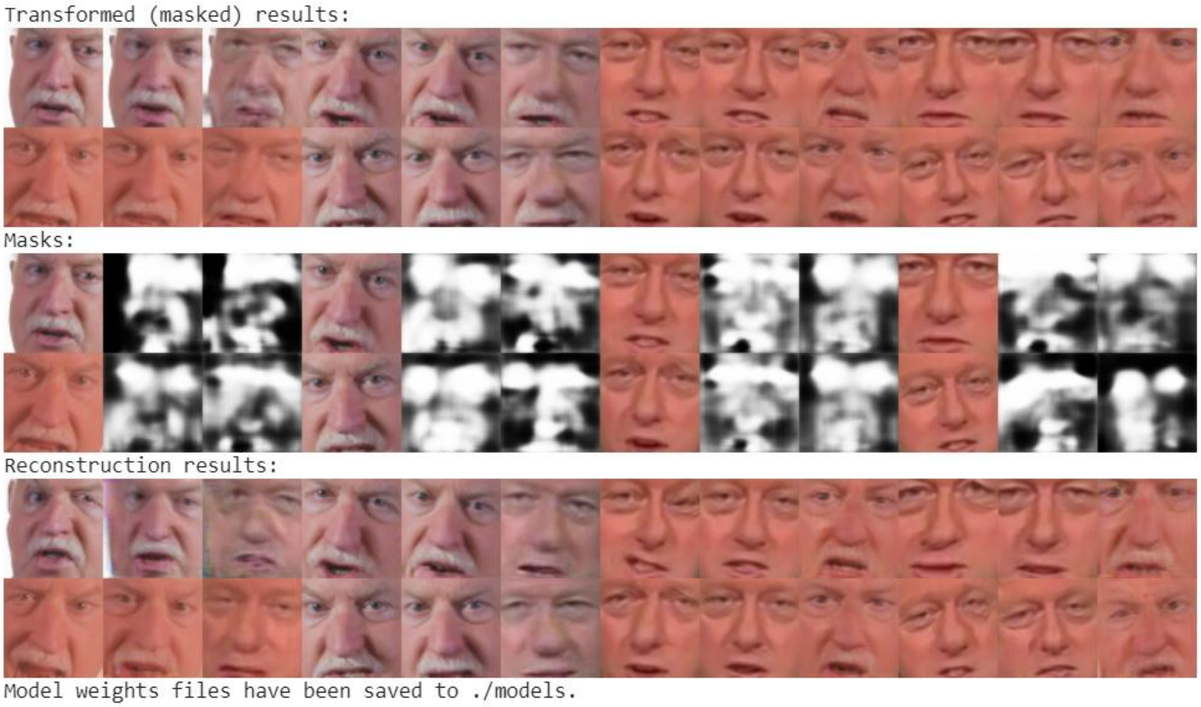
- Google Drive for backend DB

Experiments and Results

Multiple personalities were initially tried to understand the capabilities and hidden factors associated with the input image dataset generation and the final video.

Personalities under consideration:

- Mike Zyda
- Donald Trump
- Matt Damon
- Tom Cruise
- Tom Hanks
- Bill Clinton
- Nicholas Cage



```
Transformed (masked) results:
Masks:
Reconstruction results:
Model weights files have been saved to ./models.
```

Example of Transformed Results from Mike Zyda, to Bill Clinton during training

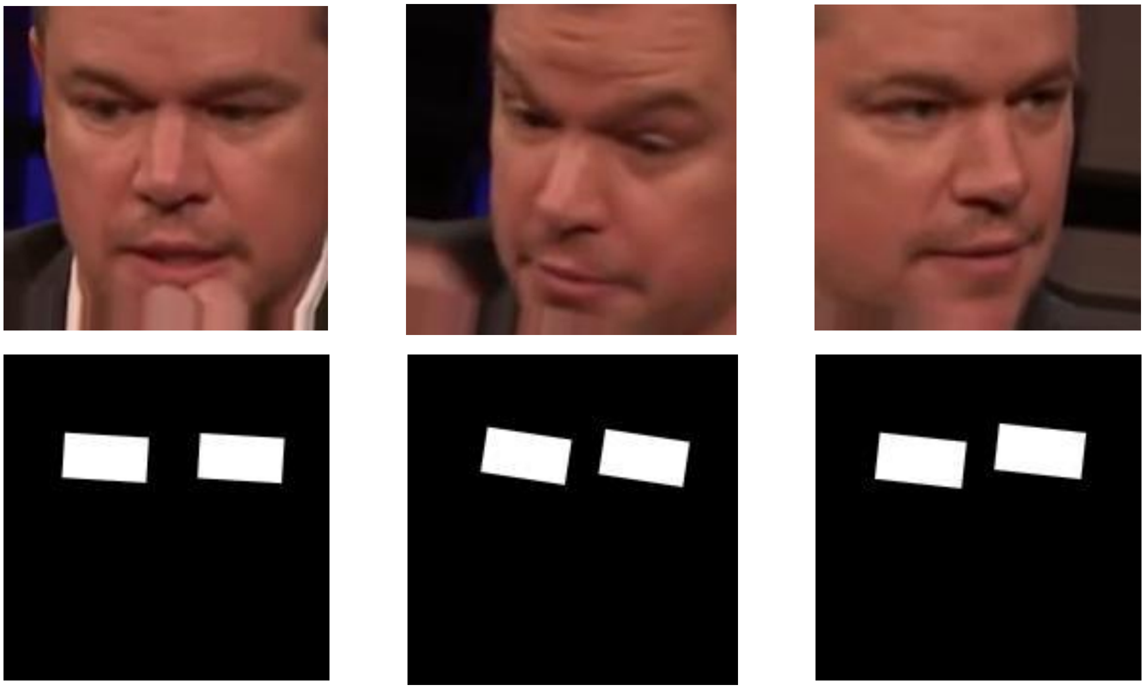


Fig: Sample images from extraction of Matt Damon input video and the corresponding binary marks

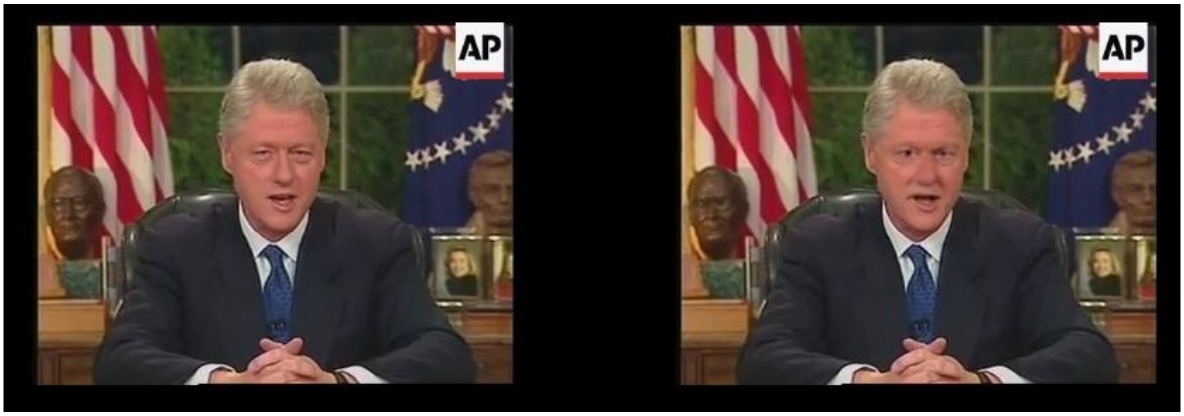


Fig: Still of final result video of superimposition of Mike Zyda on Bill Clinton

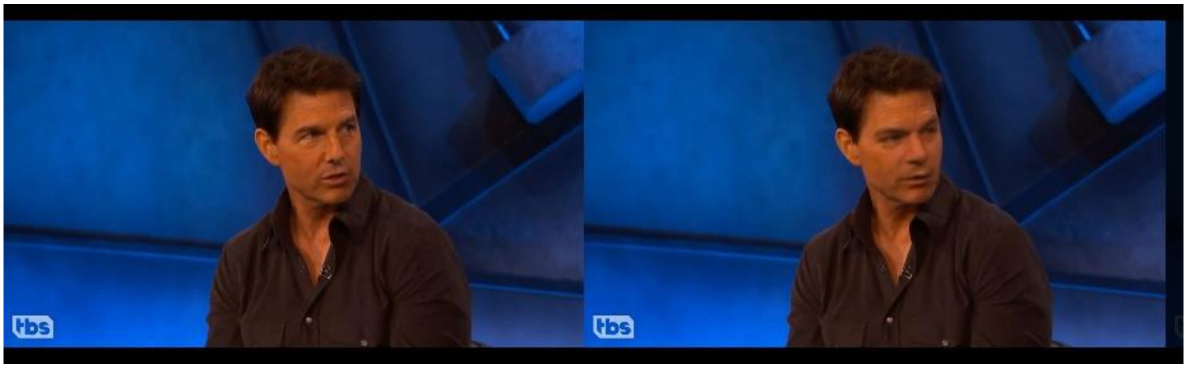


Fig: Still of final result video of superimposition of Matt Damon on Tom Cruise

Future Scope

Future work on deepfakes can extend to account for the following additions:

- Extract face masks for the entire face including the jawline and hairline.
- Handle different angles in mask creation. A model capable of formulating a facial feature mask for a surprise angle encountered during sample conversion.
- The project can also extend to account for variations in audio along with the transformed video

References

[1] Retrieved from https://en.wikipedia.org/wiki/Deepfake

[2] Retrieved from https://www.alanzucconi.com/2018/03/14/understanding-the-technology-behind-deepfakes/

[3] Retrieved from https://skymind.ai/wiki/generative-adversarial-network-gan

[4] Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10):1499–1503

[5] Deep Face Recognition – Omkar M. Parkhi et al. – bmvc, 2015

[6] Retrieved from https://github.com/shaoanlu/faceswap-GAN

[7] Han Zhang, Ian Goodfellow, Dimitris Metaxas, Augustus Odena (2018). Self-Attention Generative Adversarial Networks. arXiv:1805.08318 [stat.ML]

[8] Ian J. Goodfellow_, Jean Pouget-Abadiey, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozairz, Aaron Courville, Yoshua Bengio(2014). Generative Adversarial Nets. NIPS 2014