TrumpNet: Talking Presidents

[redacted]

[redacted]

Abstract

We wish to explore current implementations of photo-realistic lip-sync videos and potential improvements. We hope to leverage modern developments in both generative and discriminative modeling techniques to extend ObamaNet [4], a lip-sync generator that uses a U-Net backbone. TrumpNet extends on ObamaNet by incorporating additional jaw keypoints, applying pix2pixHD (instead of pix2pix), and training our model on Trump videos.

Introduction

In this project, we will explore different architectures to take text input and generate corresponding synchronized lip-sync using existing text to speech synthesis. This project poses as an extension to the paper proposing ObamaNet that uses a U-Net architecture [4]. The module is based on three parts to cover text-to-speech, speech to mouth points, and mouth points to video frames. Their specific implementation was based on Char2Wav for text-to-speech, a time-delayed LSTM to generate mouth points, and a Pix2Pix to generate the video frames based on the mouth points. In our project, we will focus solely on reimplementing the video generation element (we will not change the text-to-speech or the mouth points generation part).

Our implementation will consist of extending a current implementation [6] and exploring different datasets and methods of data processing to ideally improve upon their implementation.

Related Work

ObamaNet was first introduced in a 2017 paper by Rithesh Kumar, Jose Sotelo, Kundan Kumar, Alexandre de Brébisson, and Yoshua Bengio [4]. The paper discusses mainly the methods and techniques they used to create the network, while also mentioning the papers and other research findings they used when deciding what type of architecture they wanted to implement. Because our project will be a recreation and rebuilding of ObamaNet, we will be taking a look at many of the design choices and decisions made in the paper to guide us during the development of our project.

Alongside the ObamaNet paper, we will also be taking a look at the papers they cited. In particular, the Pix2Pix model was inspired by a paper written by Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros [3]. The Pix2Pix paper describes a method for solving image to image translation problems by exploring the effectiveness of conditional adversarial networks. This consisted mostly of testing out different conditional GANs and comparing their pixel accuracy and IoU scores. This paper outlines how Pix2Pix (used in ObamaNet for video generation from keypoints) was designed. If we consider using an alternative approach, this paper can be useful for figuring out where certain methods may fall short, and where others will succeed.

Dataset

Our data was obtained by extracting youtube videos for footage focused on the speaker's face. We began with the same dataset used by the original authors of ObamaNet. This data comes from 300 weekly presidential addresses given by ex-president Barack Obama (17 hours of video footage total). As the authors note, this data has the advantage that the speaker stays within the center of the videos [4].

The videos were downloaded using the youtube-dl library. From these videos, we were able to turn each frame into an image using ffmpeg and then feed the images into the network, which will be able to generate keypoints around the mouth and lips to create an outline that we can then use to train the network with as input. The dataset also consists of generated audio files that we will use as guiding input for the mouth and lip movement. Afterward, some preprocessing is done on the videos in order to get them into the formats and resolutions that the different models require.

For our experimentation, we will train various models on Trump's weekly addresses[1]. There are currently 50 Trump weekly addresses, ranging from February 10, 2017, to June 23, 2018.

Architecture

ObamaNet builds on the model used by Suwajanakorn et al. (2017) [5]. The main differences are ObamaNet uses a neural network instead of a computer vision model and prepends a text-to-speech synthesizer instead of directly feeding audio input.

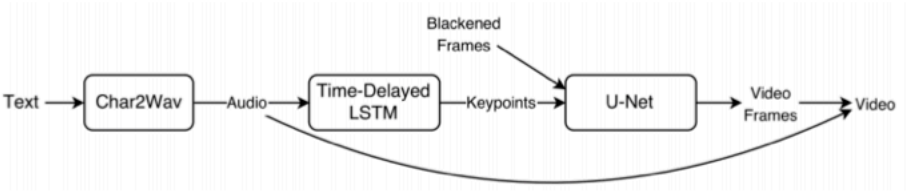


Figure 1: Flow diagram of our generation system (figure credits: [4]). ObamaNet uses Char2Wav to generate audio from text and uses U-Net architecture instead of computer vision model to translate facial keypoints into photo-realistic video frames.

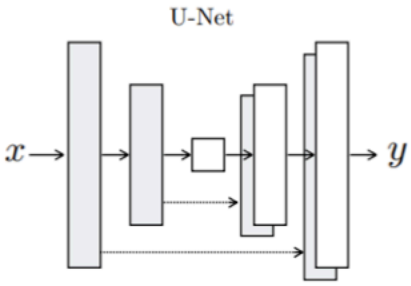


Figure 2: U-Net architecture

In our implementation we execute different architectures from the U-Net above, namely Pix2Pix. Furthermore, our implementation uses OpenFace[1] to extract the keypoints of the lips and the jaw while preprocessing the images. The preprocessed images with the lips and jaws drawn on top are fed into the Pix2Pix architecture. Pix2Pix is a Conditional GAN [3] with a U-Net generator

[1]Trump weekly addresses YouTube playlist: https://www.youtube.com/playlist?list=PLRJNAhZxtqH-MfozN_cNchyScPOBuhJrM

and separate discriminator network. We further train a Pix2PixHD [7] architecture to compare the results. As our impelmentation adds jaw keypoints to the preprocessing, we also train an implementation with only the lip keypoints to verify that adding the jaw keypoints improves performance. We preprocess the images of Trump from the compiled images collected from his weekly addresses.

Baseline Reimplementation

We collected a dataset from youtube videos published by The Obama White House youtube channel. We specifically focus on the many "Weekly Address" videos released that consist of multiple short clips of Obama speaking straight into the camera. We take those clips and separate them into images frame by frame. For each of these images, the keypoints of the face were determined and a black box was generated around the mouth. The lips based on audio are drawn onto the box; we show our reimplementation in figure 2. Following the box augmentation, we then generate lips using pix2pix.

The ObamaNet network, available via github, uses sample data and a pretrained model that we did not alter. The implementation also uses a Pix2Pix architecture but does not incorporate jaw keypoints. The network is capable of taking in audio files to output a .mp4 file of the generated video with the mouth regenerated to match audio file input. An example of the network's performance is visible in Figure 3.

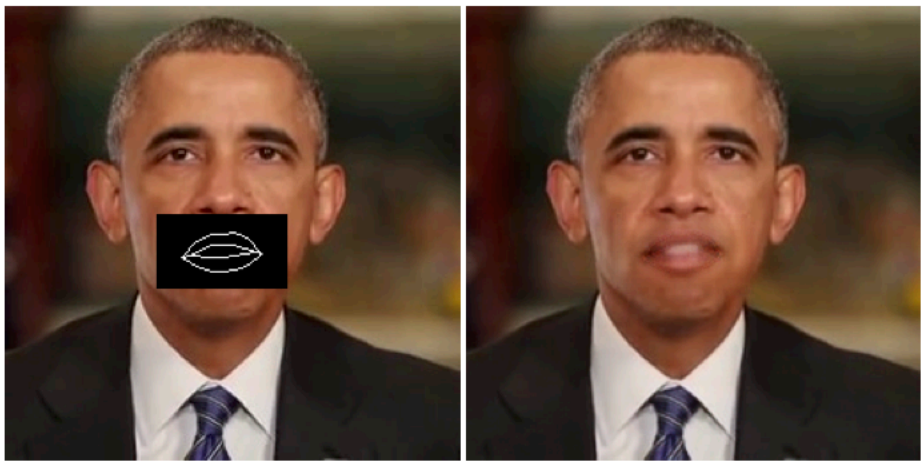


Figure 3: Example of generated mouth keypoints and generated mouth layered over of original obama footage

With this implementation, the lips on the generated video are not very clear and show obvious signs of video manipulation. The area around the mouth covered by the black box and regenerated by Pix2Pix is blurry and lower resolution than the area around it. As we do not know the dataset size or the training amount, we cannot speak to how much this network was trained and how that may be an attribute to it's lower performance.

Incorporating Jawline

In our baseline implementation, the lips were the only facial feature being reconstructed to mouth the words of our audio input. However, the human face uses more than just lips when speaking. To create more natural-looking videos, we incorporated the jawline in addition to the lips. In the data preprocessing step, we adjust how the black box was drawn onto the images and add jaw keypoints as input to the Pix2Pix model.

We use the OpenFace model[1] to incorporate lip and jawline input. Specifically, we use the facial landmark detector from the dlib library to generate 68 facial keypoints Figure 4.

Figure 4: 68 facial keypoints extracted from dlib facial landmark detector. Keypoints 49-68 correspond to the mouth and lips and keypoints 1-17 correspond to the jawline.

Figure 5: Left: generated mouth and jaw keypoints. Right: Original Obama footage

Alternative Training Network - Pix2pixHD

We explored potentional improvements to the performance by implementing the pix2pixHD model. One way Pix2PixHD improves upon Pix2Pix is by allowing the GAN discriminator to differentiate between high-resolution images. It does this by utilizing three discriminators. These three discriminators all share the same network structure where two of the discriminators are trained on downsampled images. These three discriminators allow the generator to learn to produce both coarse and fine details. In addition, Pix2PixHD utilizes two networks for the generator. One network is trained on lower resolution images, and then the second network is appended to the first, and finally both are trained on high resolution images. These two networks allow the generator to use both global information as well as local information to generate the output image [7]. A full breakdown of the Pix2PixHD architecture can be viewed in the Appendix in Figure 14.

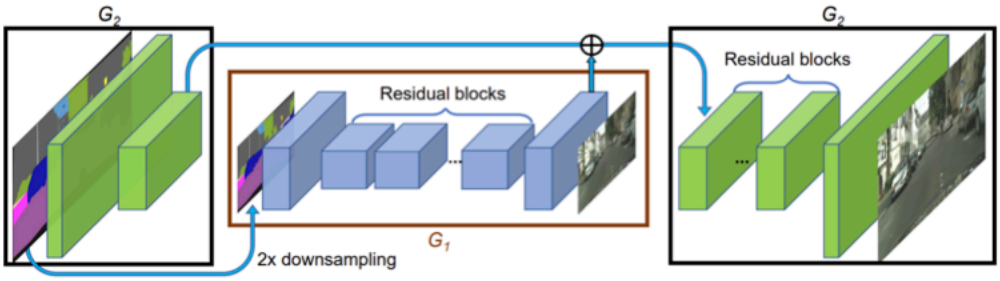


Figure 6: Network architecture of our generator. We first train a residual network G1 on lower resolution images. Then, another residual network G2 is appended to G1 and the two networks are trained jointly on high resolution images. Specifically, the input to the residual blocks in G2 is the element-wise sum of the feature map from G2 and the last feature map from G1.[7]

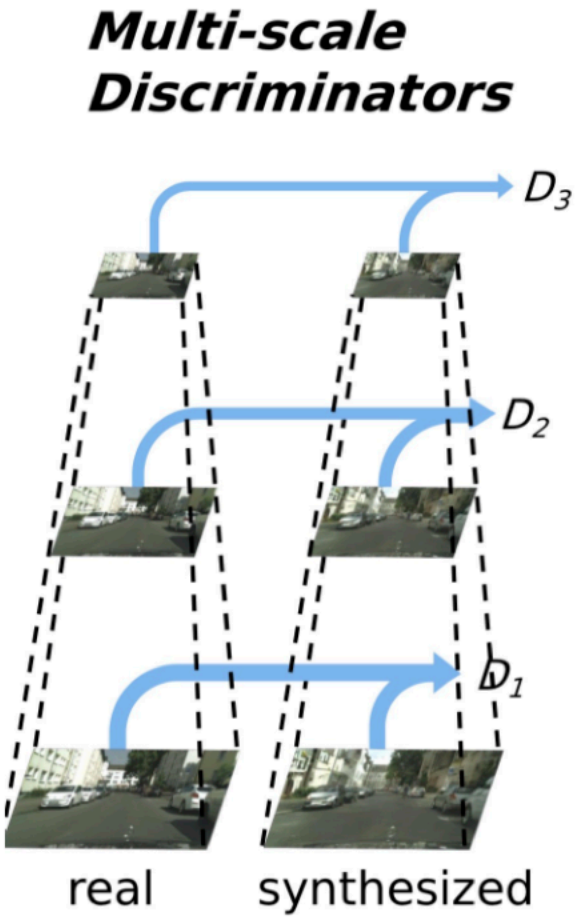


Figure 7: The three different multi-scale discriminators that are trained on three different scalings of the same image. [2]

Evaluation Metrics

We evaluate our outputs with two different metrics. Our first metric is mean squared difference of reconstructed keypoints (see Figure 8). For this metric we extract the keypoints from generated images and their corresponding target images (ground truth) to compute the L2 loss between the two.

Figure 8: Extract keypoints (visualized here as green dots) to calculate L2 losses

Results

Our models run for 350K+ training iterations.

Here is our results on mean squared error in reconstructed keypoints:

- Pix2PixHD lip + jaw: 0.931
- Pix2Pix lip + jaw: 1.955
- Pix2Pix lip: 1.784

The keypoints extracted from the Pix2Pix model using lips and jaws performs worse than the model using lips only (1.784 vs 1.955). This may be because the Pix2Pix model trained on lips and jaws has more room for error as it has to reconstruct the jaw keypoints whereas the Pix2Pix model trained on lips only is able to use the original image's jaw keypoints without having to reconstruct them. However, we see that Pix2PixHD with mouth and jaw keypoints has the lowest mean squared error of 0.930, which is much less than either Pix2Pix model. This suggests Pix2PixHD is a clear improvement on Pix2Pix.

Our second metric is grayscale L2 pixel difference of ground truth images vs reconstructed images. We take a look at grayscale L2 pixel difference (for a single test image):

- Pix2pixHD jaw: 9.134
- Pix2pix jaw: 10.104
- Pix2pix lip: 6.137

Using this metric, we see Pix2pixHD jaw performs better at reconstructing target image than Pix2Pix jaw (9.134 vs 10.104). However, both perform worse on the L2 metric than Pix2Pix lip (6.137) because both models have more of the input image occluded by the black box. We can see the difference visually in Figure 9.

Figure 9: Grayscale absolute pixel difference with target image

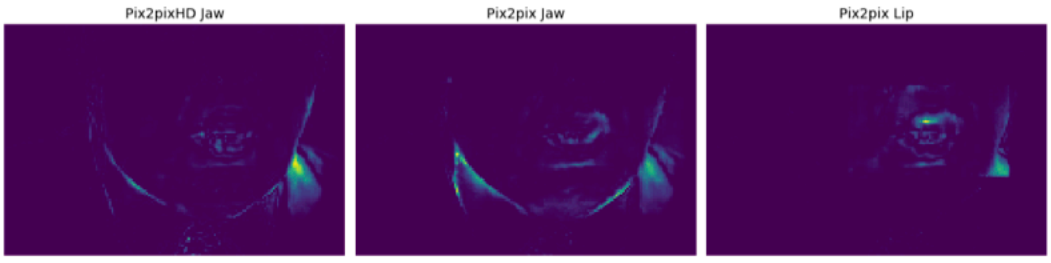


Figure 10: Grayscale L2 pixel difference with target image (zoomed in)

Qualitatively, Pix2PixHD seems much better than Pix2Pix. As can be seen in Figure 11, Pix2Pix loses fine detail in the lip area. On the other hand, Pix2PixHD, at instances, can be hard to distinguish the synthesized image from the original image as can be seen in Figure 12. It is also interesting to note that Pix2PixHD seemed to be able to generate fine details rather quickly in it's learning process as can be seen in Figure 13.

Figure 11: Left: Original footage. Middle: Generated mouth and jaw keypoints. Right: Generated mouth layered over using pix2pix on Trump footage

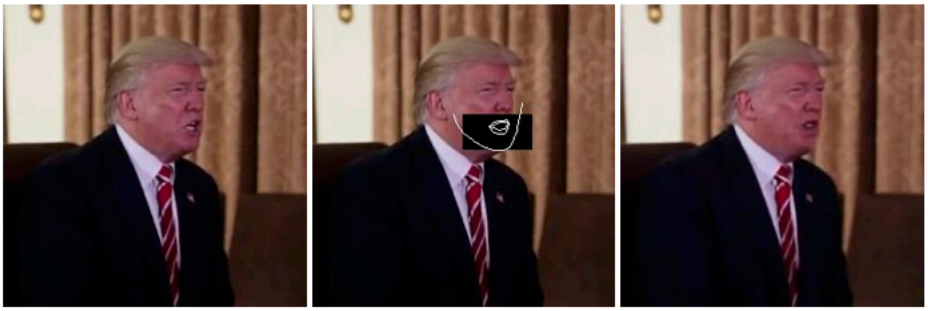


Figure 12: Example of generated mouth keypoints and generated mouth layered over using pix2pixHD on Trump footage. Left: real image, Middle: input image, Right: synthesized image

Figure 13: Progression of generated mouths on Trump photos across different epochs. Epochs: 1, 2, 3, 4

Discussion and Future Work

Analysis

For our experimentation, we wanted to see whether adding a jawline to our training data would output better results. Furthermore, we wanted to test whether Pix2PixHD would produce smoother output than our original Pix2Pix network.

We found that changing the model that the original ObamaNet used, which was Pix2Pix, to an updated model, Pix2PixHD, improved the visual quality of the resulting generated video. The videos created with Pix2Pix have a visible outline within the area where the black box was generated, shown in Figure 11. We saw the that the pix2pixHD model did a much better job dealing with this issue and created videos with a much smoother appearance, seen in Figure 12. This is likely due to the multiple improvements that Pix2pixHD implements, including a coarse-to-fine generator and multi-scale discriminators.

One consistent observation of the resulting models is that the textures around the mouth and lips have some noise. The barrier between the teeth and the lips are blurred, which causes the lips to sometimes look like teeth or vice-versa. A possible reason for this problem is that President Trump's appearance changes between the different addresses in the training set. This is caused by different lighting, as well as whenever the President decides to update his tan, which alters the complexion of his skin. It is also understandable why the network has trouble with teeth, as they are a very small part of an image that relies on very fine details to appear realistic and do not always appear constantly when someone is speaking. The most likely cause of error is simply not training enough. While a single video provides a lot of images to train with, and helps provide decent context for speech patterns, it does not provide much context for different lighting, angles, or positions.

A point to note, our metrics and implementation as a whole does not clearly reflect the usefulness of predicting jaws. This alteration would be an expected improvement when generalizing to new audio and that new audio's lip movement. In our evaluations of models tests were performed on actual videos of Trump, therefore by only predicting the mouth, the jaws would still be correct for the given speech. To actually test whether jaws provide better performance is difficult because by generalizing to new speech, there would be no corresponding ground truth to obtain metrics from.

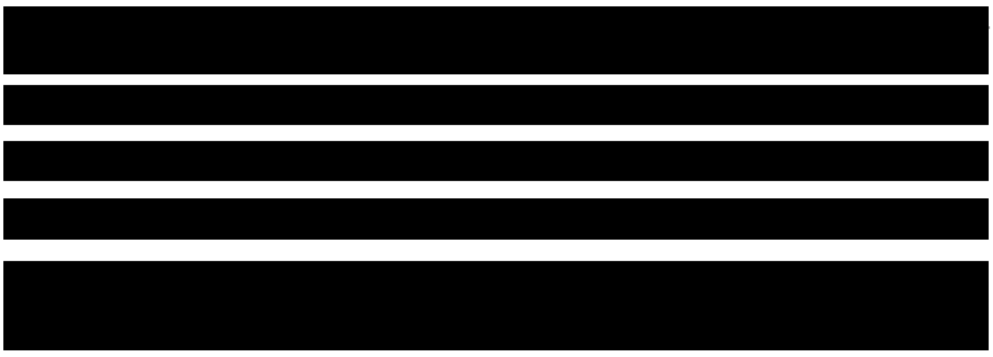
Future Work

Our results are reported using only a subset of available data. Our biggest obstacle has been computational availability and time. While we were able to obtain, extract, and preprocess a lot of the data we limited our training set due to time limitations. The problem is that the sheer amount of data that needs to be processed in order to achieve satisfying results. A single video can easily result in over 3000 images, multiplied by 20, 50, or possibly 100 videos and our preprocessing time and and the training rate would increase signficantly. Because of this, we needed to significantly reduce the size of our training data in order to obtain a trained model within the constrained time strain of this project. This limitation creates much room for improvement. If given the opportunity to work on this project again, we are confident that a model trained longer will produce better results.

Another possible improvement is that currently Pix2Pix and Pix2PixHD generate images from the appropriate keypoints, however in general the shape of your mouth and jaw is very related to what the word spoken and the word about to be spoken. If we had more computational resources, we would also like to investigate perhaps using an LSTM or RNN to model these temporal dependencies. We hypothesize this could lower the dimensionality that our GAN would have to predict and improve our results.

Another thing we noticed is colloquially called the uncanny valley. It's a phenomenon where generated images that look realistic but feel off to the point of where they are creepy. We noticed this in some of our generated images as well. We hypothesize that one of the major reasons is that we don't model the eyes or nostrils. While we speak, despite not being obvious our eyes, nostrils, and brows are an integral part of communication. If given more time we would like to test our hypothesis by having our GAN predict eye movements as well.

Individual Member Contribution



Appendix

```
GlobalGenerator(
  (model): Sequential(
    (0): ReflectionPad2d((3, 3, 3, 3))
    (1): Conv2d(3, 64, kernel_size=(7, 7), stride=(1, 1))
    (2): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False)
    (3): ReLU(inplace=True)
    (4): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2),
                padding=(1, 1))
    (5): InstanceNorm2d(128, eps=1e-05, momentum=0.1, affine=False)
    (6): ReLU(inplace=True)
    (7): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2),
                padding=(1, 1))
    (8): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False)
    (9): ReLU(inplace=True)
    (10): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2),
                padding=(1, 1))
    (11): InstanceNorm2d(512, eps=1e-05, momentum=0.1, affine=False)
    (12): ReLU(inplace=True)
    (13): Conv2d(512, 1024, kernel_size=(3, 3), stride=(2, 2),
                padding=(1, 1))
    (14): InstanceNorm2d(1024, eps=1e-05, momentum=0.1, affine=False)
    (15): ReLU(inplace=True)
    (16): ResnetBlock(...)
    (17): ResnetBlock(...)
    (18): ResnetBlock(...)
    (19): ResnetBlock(...)
    (20): ResnetBlock(...)
    (21): ResnetBlock(...)
    (22): ResnetBlock(...)
    (23): ResnetBlock(...)
    (24): ResnetBlock(...)
    (25): ConvTranspose2d(1024, 512, kernel_size=(3, 3), stride=(2, 2),
                          padding=(1, 1), output_padding=(1, 1))
    (26): InstanceNorm2d(512, eps=1e-05, momentum=0.1, affine=False)
    (27): ReLU(inplace=True)
    (28): ConvTranspose2d(512, 256, kernel_size=(3, 3), stride=(2, 2),
                          padding=(1, 1), output_padding=(1, 1))
    (29): InstanceNorm2d(256, eps=1e-05, momentum=0.1, affine=False)
    (30): ReLU(inplace=True)
    (31): ConvTranspose2d(256, 128, kernel_size=(3, 3), stride=(2, 2),
                          padding=(1, 1), output_padding=(1, 1))
    (32): InstanceNorm2d(128, eps=1e-05, momentum=0.1, affine=False)
    (33): ReLU(inplace=True)
    (34): ConvTranspose2d(128, 64, kernel_size=(3, 3), stride=(2, 2),
                          padding=(1, 1), output_padding=(1, 1))
    (35): InstanceNorm2d(64, eps=1e-05, momentum=0.1, affine=False)
    (36): ReLU(inplace=True)
    (37): ReflectionPad2d((3, 3, 3, 3))
    (38): Conv2d(64, 3, kernel_size=(7, 7), stride=(1, 1))
    (39): Tanh()
  )

  Note: ResnetBlock(...) = ResnetBlock(
      (conv_block): Sequential(
        (0): ReflectionPad2d((1, 1, 1, 1))
        (1): Conv2d(1024, 1024, kernel_size=(3, 3), stride=(1, 1))
        (2): InstanceNorm2d(1024, eps=1e-05, momentum=0.1, affine=False)
        (3): ReLU(inplace=True)
        (4): ReflectionPad2d((1, 1, 1, 1))
        (5): Conv2d(1024, 1024, kernel_size=(3, 3), stride=(1, 1))
        (6): InstanceNorm2d(1024, eps=1e-05, momentum=0.1, affine=False)
      )
    )
```

Figure 14: Pix2pixHD Network Architecture

References

[1] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," 2016.

[2] Aug 2019. [Online]. Available: https://arxivnote.ddlee.cn/2019/08/22/Image-to-image-Translation-pix2pixHD-MUNIT-DRIT-vid2vid-SPADE-INIT-FUNIT.html

[3] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2017.632

[4] R. Kumar, J. Sotelo, K. Kumar, A. de Brebisson, and Y. Bengio, "Obamanet: Photo-realistic lip-sync from text," 2017.

[5] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing obama: Learning lip sync from audio," *ACM Trans. Graph.*, vol. 36, no. 4, July 2017. [Online]. Available: https://doi.org/10.1145/3072959.3073640

[6] A. Victor, "Obama-lip-sync," https://github.com/acvictor/Obama-Lip-Sync, 2019.

[7] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," 2017.

[8] W. Xie, "Vggface2 dataset for face recognition," https://github.com/ox-vgg/vgg_face, 2019.