# CMP-SCI 5390 Final Project Phase 1: Data Preparation

Jackson Hoenig

March 4, 2021

## Contents

## 1 Description

Every decade there have been advances in technology that have changed life drastically for everyone. In the 1980's it was the personal computer, in the 2010's it was the prevalence of mobile devices. I believe that for the 2020's we will see Artificial Intelligence used more than ever before. The subset of AI, Deep Learning, especially seems to be on the rise with computer vision in automobiles, and other video-related operations. Of all of the potential uses of Deep Learning, we may need to fear Deepfakes the most. a Deepfake is a video that has been altered in some way to make it look real. Usually, a driving video of a person is edited to make the face look like it is someone else. This for the past few years has mostly been just for entertainment purposes and easily detectable by an informed viewer. However, with graphics cards becoming stronger and the creation of Deepfake libraries, videos are being created that look almost indistinguishable from a real video. This is a major problem for the security of evidence, public officials, and identity theft. That is why we must have some way of detecting fake videos from real videos. Without such a device people caught on film committing a crime could say it was a Deepfake and other innocent people could be accused with such a video. Society could crumble into chaos with video evidence meaning nothing to anyone.

Others have tried to create Deepfake detection devices with Deep Learning [2], I am attempting to do the same and have created my dataset of 503 real images and 361 frames of Deepfaked videos. This is a small dataset but it was created entirely by myself. I plan on also attempting to work with a dataset from Facebook.ai[1] however since I am trying to run this project exclusively on Google Colab I can only use part of the dataset at a given time.

## 2 Organization

### 2.1 Created Image Dataset

As stated previously, I created a dataset of 503 images. These images were 4k images taken with my Moto g(8) smartphone. They were then uploaded to Google Drive then edited with my Google Colab notebook to crop them to a square and resize them to 256x256 to match the deepfake maker notebook I used. With the Deepfake maker, I used I gave it various driving videos and sample images that were taken from the 503 real images in the same dataset. I did this to try and make the deepfake detector have the hardest possible time. I then created 8 videos and extracted the 361 images from those videos.

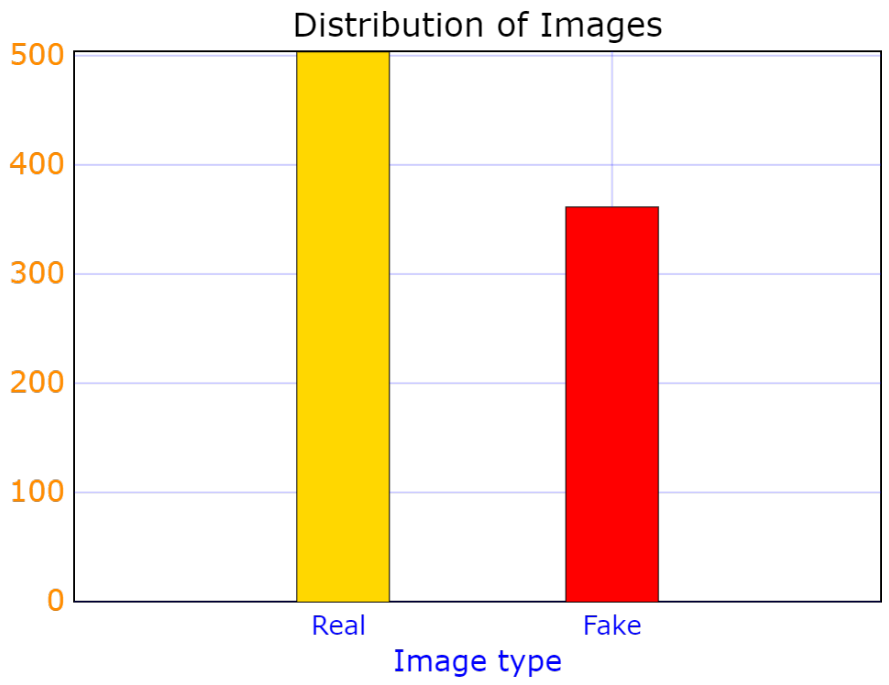
I have a few concerns with this dataset. Firstly, it is very small with a total of only 864 images, even with data augmentation it seems that I will need more to get accurate results. This is difficult though because it is very time-intensive to take hundreds of pictures, upload them, preprocess them, then make deepfakes of them. Secondly, these images are being compared when in reality it would be ideal to have the full videos compared to each other. Finally, it is unrealistic to think this model will perform well on any face other than my own since that is all it was trained on. It would be better if I had a diverse dataset of both real and fake faces to train my model on.

## 2.2 Downloaded Video Dataset

I have been looking for a dataset online, and I found a large one at dfdc.ai, however, this dataset was terabytes big and I could not handle resources of that size. I decided to download 10Gb of the dataset, but I realized that the videos are not labeled as real or fake. This leads me to believe that the dataset is intentionally incomplete for the competition. Therefore I will have to keep searching for another dataset that includes labels. Or create my own.

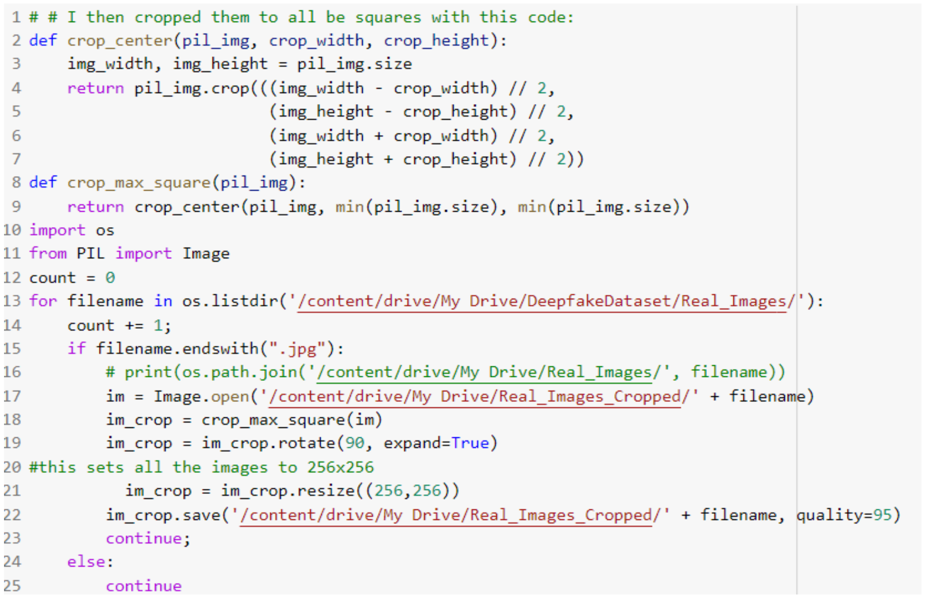
## 3 Distribution of Images

As you can see from this bar graph. The distribution of images is 361/503 or 361/864 is the percentage of fake images. So our baseline for detecting a fake would be 37%.



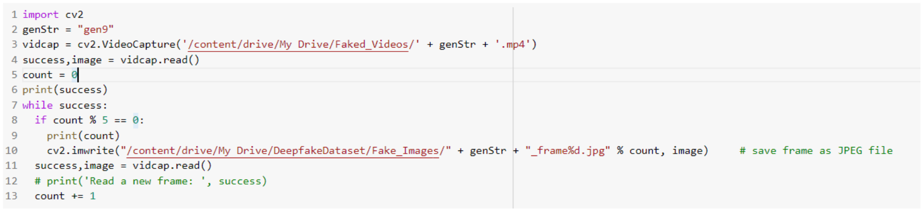
## 4 Normalization

As stated previously, all of my images that were taken with my camera were originally at a high definition but reduced to 256 by 256 pixels after first being cropped to an aspect ratio of a square. I did this with the following code: Then



```python
# # I then cropped them to all be squares with this code:
def crop_center(pil_img, crop_width, crop_height):
    img_width, img_height = pil_img.size
    return pil_img.crop(((img_width - crop_width) // 2,
                         (img_height - crop_height) // 2,
                         (img_width + crop_width) // 2,
                         (img_height + crop_height) // 2))
def crop_max_square(pil_img):
    return crop_center(pil_img, min(pil_img.size), min(pil_img.size))
import os
from PIL import Image
count = 0
for filename in os.listdir('/content/drive/My Drive/DeepfakeDataset/Real_Images/'):
    count += 1;
    if filename.endswith(".jpg"):
        # print(os.path.join('/content/drive/My Drive/Real_Images/', filename))
        im = Image.open('/content/drive/My Drive/Real_Images_Cropped/' + filename)
        im_crop = crop_max_square(im)
        im_crop = im_crop.rotate(90, expand=True)
#this sets all the images to 256x256
          im_crop = im_crop.resize((256,256))
        im_crop.save('/content/drive/My Drive/Real_Images_Cropped/' + filename, quality=95)
        continue;
    else:
        continue
```

the images for the fake videos were extracted from the deepfake videos that were already 256x256 pixels. Each 5th frame was extracted to be as an image to prevent to close of images be included.



```python
import cv2
genStr = "gen9"
vidcap = cv2.VideoCapture('/content/drive/My Drive/Faked_Videos/' + genStr + '.mp4')
success,image = vidcap.read()
count = 0
print(success)
while success:
  if count % 5 == 0:
    print(count)
    cv2.imwrite("/content/drive/My Drive/DeepfakeDataset/Fake_Images/" + genStr + "_frame%d.jpg" % count, image)     # save frame as JPEG file
  success,image = vidcap.read()
  # print('Read a new frame: ', success)
  count += 1
```

## References

[1] URL: `https://dfdc.ai/`.

[2] P. Korshunov and S. Marcel. “Vulnerability assessment and detection of Deepfake videos”. In: *2019 International Conference on Biometrics (ICB)*. 2019, pp. 1–6. DOI: `10.1109/ICB45273.2019.8987375`.