Capstone Project 3: Using deep learning to distinguish real and fake faces

# Introduction

As we move into the data age, facial recognition has reached a new peak. Some examples of facial recognition include surveillance technology for law enforcement/casinos or facial filters used in Snapchat or images. While it is undisputed that this technology has improved our lives tremendously, it has some significant downside. Especially in the realm of law enforcement, a wrong facial recognition matching could put the wrong suspects into jail and have a tremendous negative impact on their life. In social media, facial recognition determines the validity of profile photos. It helps determine whether the account is real or fake, and helps prevent internet trolls. Therefore, **we will build a deep learning convolution neural network to distinguish faces from photoshopped or not and see how well they can determine the two groups.**

# Data

The computational intelligence and photography lab provided this dataset and posted it on the Kaggle website. It contains 2041 photos of faces (1000s being real and ~1000s being fake). The fake faces were faces modified via photoshop, explicitly targeting the eyes, noses, and mouth.

# Methodology

We will be using a fastai version of the deep learning neural network (built on Pytorch) to create a classifier to distinguish the faces for this project. Instead of making the neural network from scratch, we will be using transfer learning techniques and use existing architects with a customized latter layer to perform the classifier function (In fastai, we call the custom layer head). Also, we will use the error rate as our metrics and cross-vacreatelidation loss as our loss function.

# Transformation + Augmentation

Since we are dealing with images, we will transform the pictures using the built-in batch transformation module in the fastai. The batch transformation in the fastai includes flipping, random light transformation, rotation, and zoom. It should encompass the majority of changes required for handling variations of an image during analysis. While all these augmented images are a variation of a particular image, they will improve our model's accuracy.

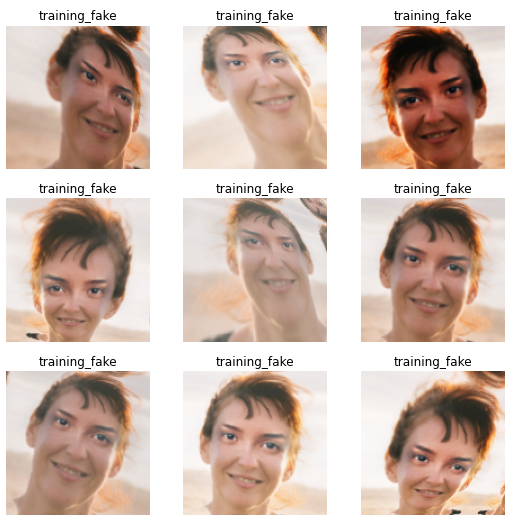


Figure : Augmentation of a face using batch augmentation module of the fastai. It includes flip and other transformation

# Model Selection/Evaluation

Two deep learning models/architects were used to build the classifiers to distinguish the fake and the real faces. The two architects are ResNet 34 and MobileNetV2. ResNet is a short name for a residual network and MobileNet is a convolutional neural network specifically designed to handle mobile images. I provided the details of both architects below for people who are interested in it.

ResNet: https://neurohive.io/en/popular-networks/resnet/

MobileNet: https://arxiv.org/abs/1704.04861

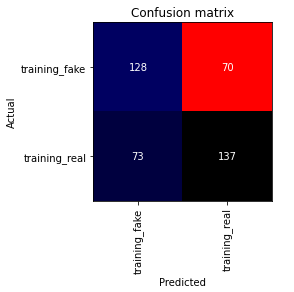


Figure Confusion Matrix for the Reset34 Model

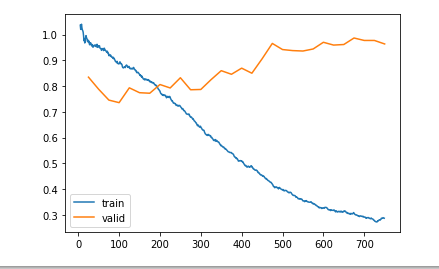


Figure How the loss function changes during the training of ResNet34

Based on the confusion matrix and our loss function, we can see that a ResNet34 experienced overfitting when the epochs become deeper. Also, the final model’s error rate on our model was ~33 %. Therefore, the classifier was good but not good enough. We will now look at MobileNetV2.

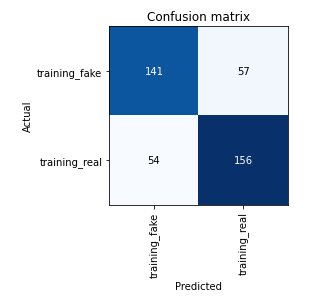


Figure Confusion Matrix for the MobileNetV2 Model

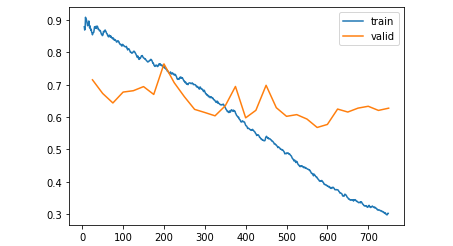


Figure How the loss function changes during the training of MobileNetV2

Based on the confusion matrix and our loss function, the MobileNetV2 outperforms the ResNet 34. The difference in loss function at both models' lowest point is roughly 0.1, and the error rate of MobileNetV2 was ~25%, which is 8% lower than ResNet. Also, the overfitting characteristics of ResNet did not carry out to MobileNetV2. Instead, the loss function appears to oscillate around 0.65 instead of getting worse with increase epochs. Therefore, MobileNet is probably a better model for this particular purpose.

# Evaluating Activation Layer

One of the things we can do to evaluate deep learning convolution neural network is to see our architects' activation layers. Activation layers are useful because, ideally, our activation layer should distinguish the real and fake faces. When we look at the last layer of our architects, the activation layer, it should tell us what features the model is looking at to evaluate whether this is a face. We can then see if those features are the ones being used to distinguish real and fake faces.

# 

Figure ResNet Activation Layer. Yellow means high activation area, blue means low activation area



Figure MobileNetV2 Activation Layer. Yellow means high activation area, blue means low activation area

When looking at the activation layers, it is still challenging to tell what the model is looking for to determine a face and not a face. This is expected. Computer vision research is a Ph.D. worthy subject that is still developing as we speak. It is unlikely for a computer vision eustatic like me to interpret these images with significant meaning. One thing in particular that I noticed is that the models are looking at the foreheads quite often. Since we know that eyes, nose, and mouth are the parts of faces that were photoshopped, it is not surprising that our models were not able to get lower than 10 % in terms of error rate. It might just be because our model is looking at the wrong thing!

# Conclusion and Final Take-Aways

If the question is, can we build a deep learning neural network that can distinguish a real and fake face? The answer is yes, we definitely can. However, this model is essentially a black box that’s impossible to understand and, can be inaccurate depend on your error tolerance. Therefore, it is very important to evaluate model bias based on the image that was given.