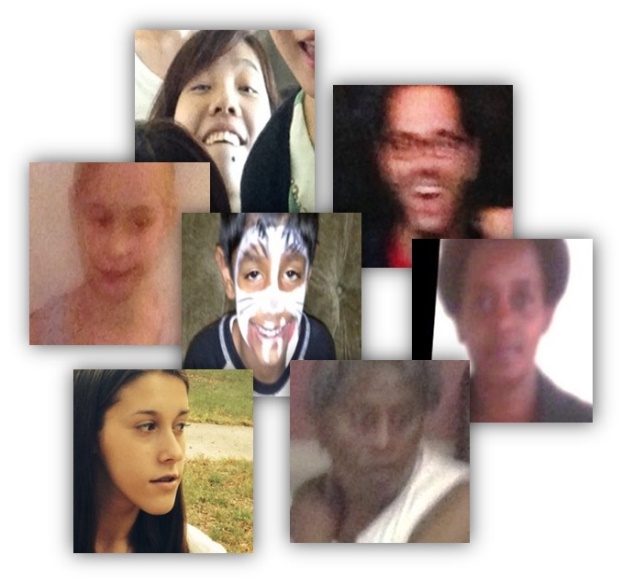
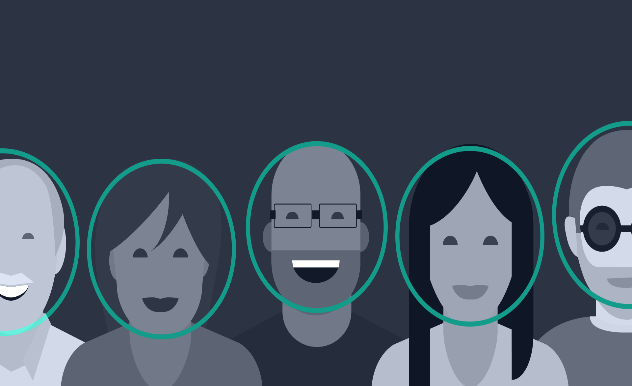
**Machine Learning Project – CS 4347.001**

**Gender Classification Using Convolutional Network**

Xuan T. Le

Github link: https://github.com/XuanTLe/PROJECT\_CS4347

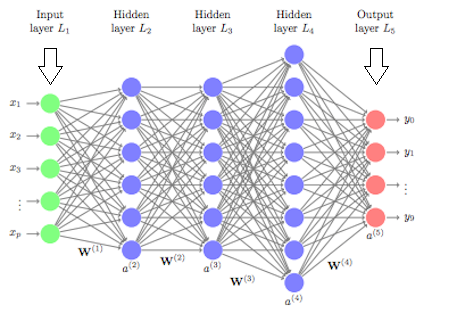
**1. Introduction**

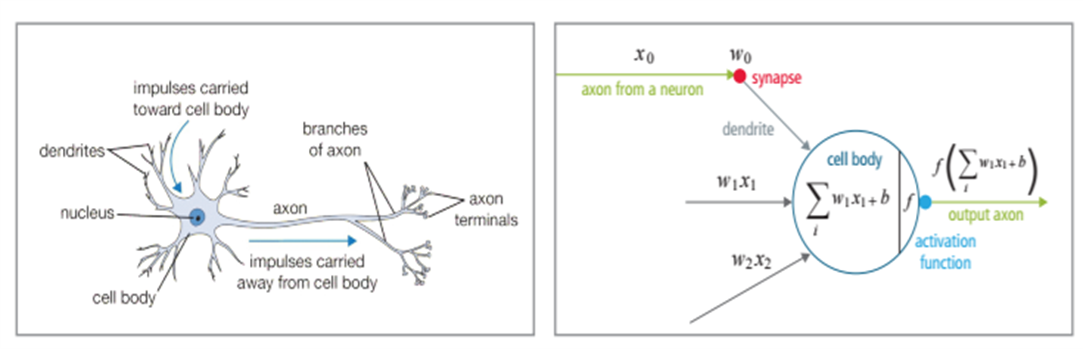
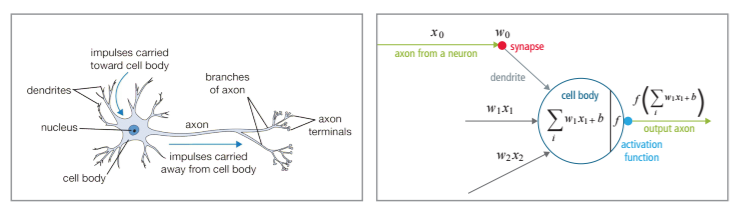
Gender is one of the fundamental information that is vital in today’s world. It does not only identify a person, but also is an element of how the world view one person and how they view themselves. Together with other important information about a person, such as name, age, and ethnicity, gender increasingly becomes vital in society as it is a core factor to identify a person for different goals in law enforcement, medical, marketing, social surveys, etc. For this reason, in the technology world, recognizing gender from images and videos plays a key role in intelligent applications such as visual surveillance, control access, and human-computer interaction.

“Automatic age and gender classification has become relevant to an increasing amount of applications, particularly since the rise of social platforms and social media. Nevertheless, performance of existing methods on real-world images is still significantly lacking, especially when compared to the tremendous leaps in performance recently reported for the related task of face recognition”. (Levi, Hassner)

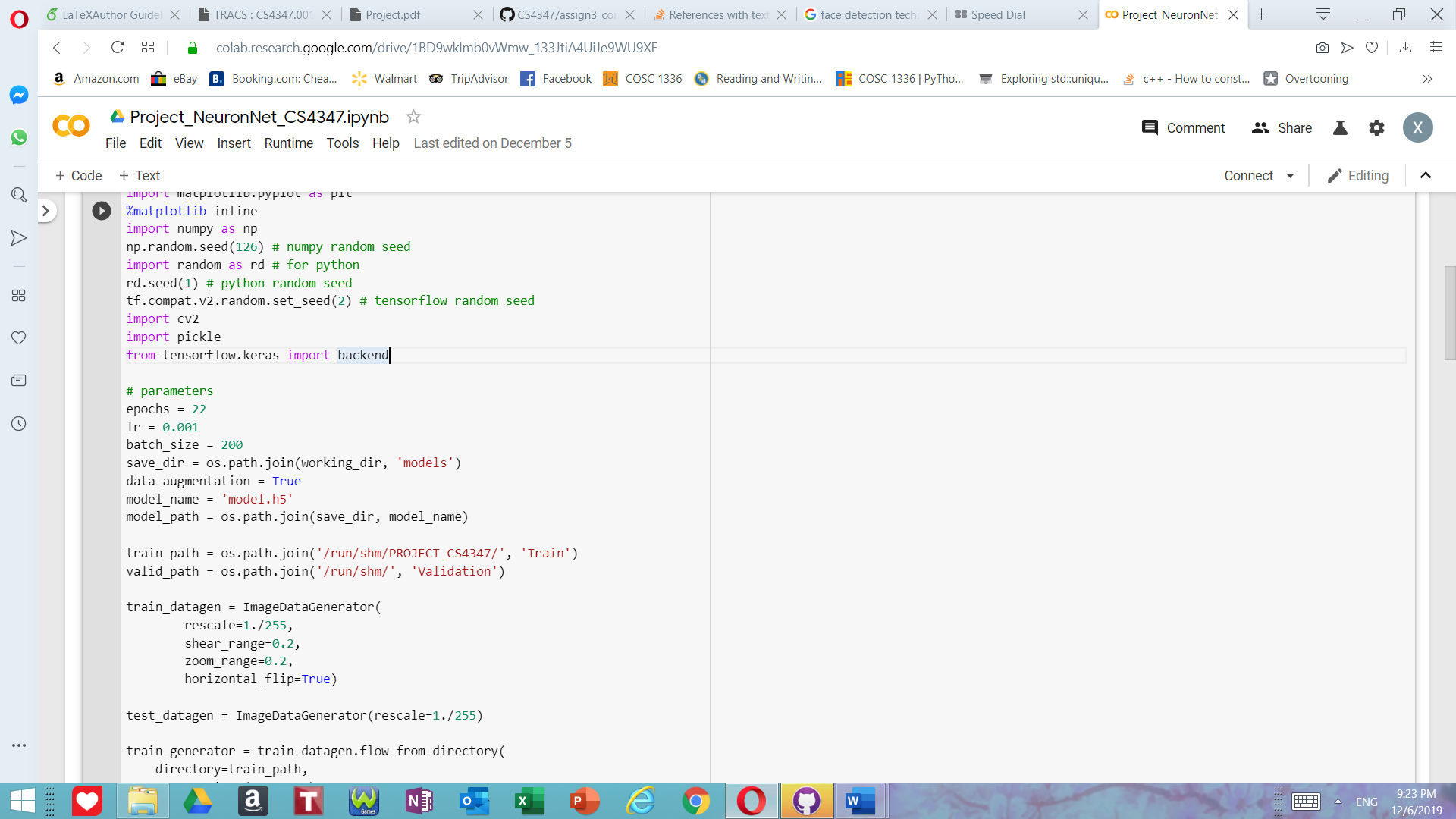
For this reason, Gil Levi and Tal Hassner have developed Age and Gender Classification using Convolutional Neural Networks and published them in IEE Workshop on Analysis and Modeling of Faces and Gestures (AMFG), at the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) in Boston in 2015. The result “demonstrates that the networks dramatically outperform other dramatically outperform current state-of-the-art methods.” (Levi, Hassner)

**2. Problem description**

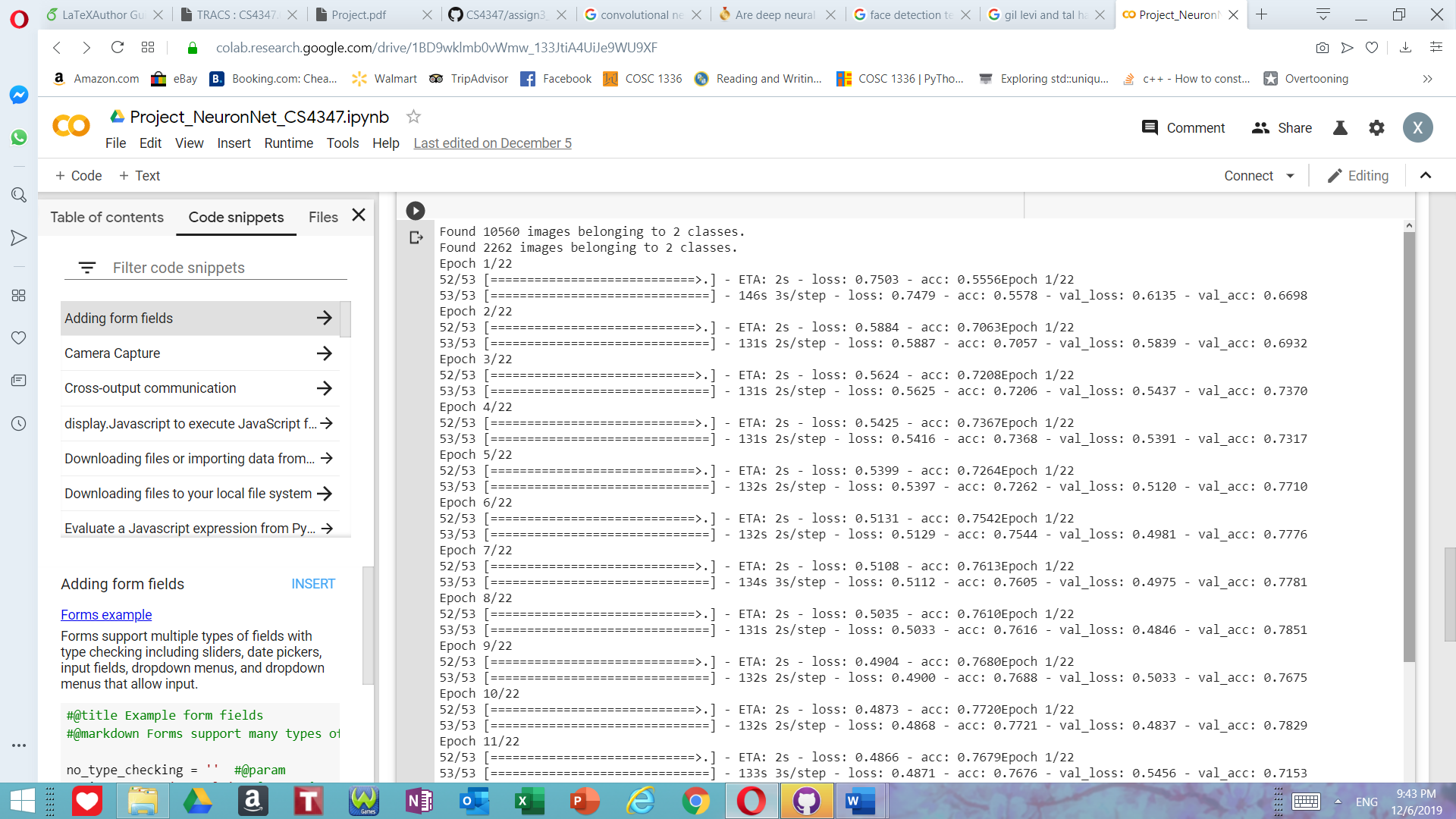
For the focus and deep understanding, I pick gender as the topic for two categories to classify rather multiclassification. Through researching, I learn that there are several applications in Computer Vision that support useful means for classification based on images such as OpenCV. At first, I look through all the current and published products on Internet for the study and reference. Then, I decide which path and method I choose for my project. I find out that Keras library is easier to use compared to PyTorch as it focuses on the interface, the templates, the outside rather than digging deeper behind the scene as in the case of PyTorch. I learn that Convolutional Neural Networks (CNNs) are the best technique currently in classifying and predicting images. I look through other techniques such as KNN, Naïve Bayes, SVM for the goal of my project, but they are not designed to be well suited at recognizing and classify images. Whereas CNN models after human’s neural network. Thus, it learns and recognizes similarly to how we learn and recognize objects.

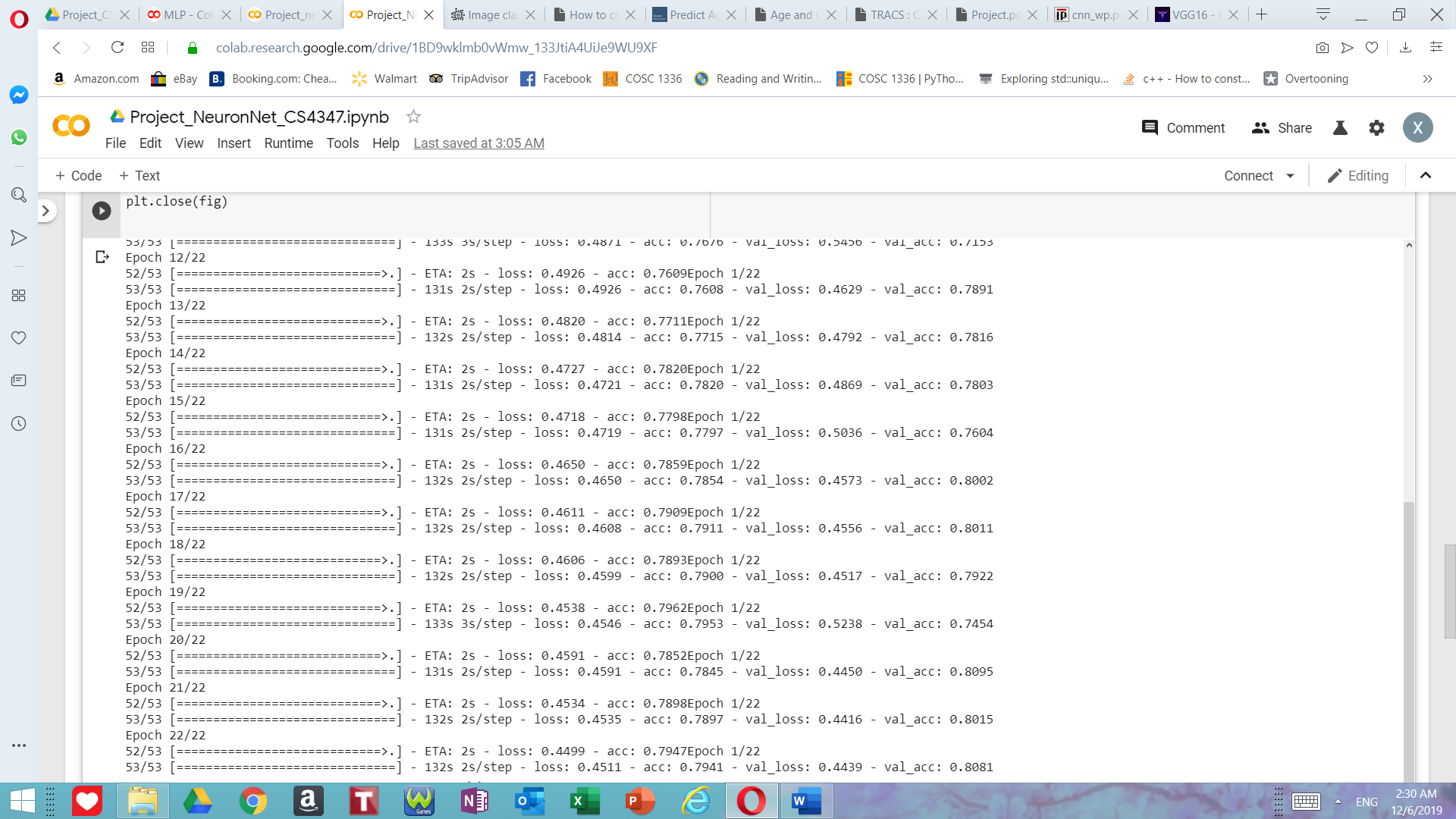
CNN is a network of connected neurons that transfer messages to each other. The network has an input layer that receives the low-level patterns from the image, hidden layers and finally an output layer. The neurons reside in each layer has the connection to each other, such that each time we tune the weights on these connections, the network learns better. The result is that, when the trained network receives a new image with similar patterns it has learnt, it gives a more accurate prediction

For the dataset, I downloaded UTKface from Kaggle which consists of 23708 images with annotation from age, gender, and ethnicity. The image’s name has the annotations embedded like this: 12\_1\_0\_20170109204507030. The first number represents age (1 to 110 years-old), the second is gender (1 for female, 0 for male), the third one is ethnicity (0: white, 1: asian, 3: African, 4:Latin and India). Since the dataset does not have CVS files for metadata, I wrote a script to access annotations in the image’s name to collect labels which is the second number in the name. From my script, the images are classified and moved to the train and test folders respectively with same distribution, which each set maintains all ages and ethnicities. The images are already processed from original dataset, which are cropping and aligning.

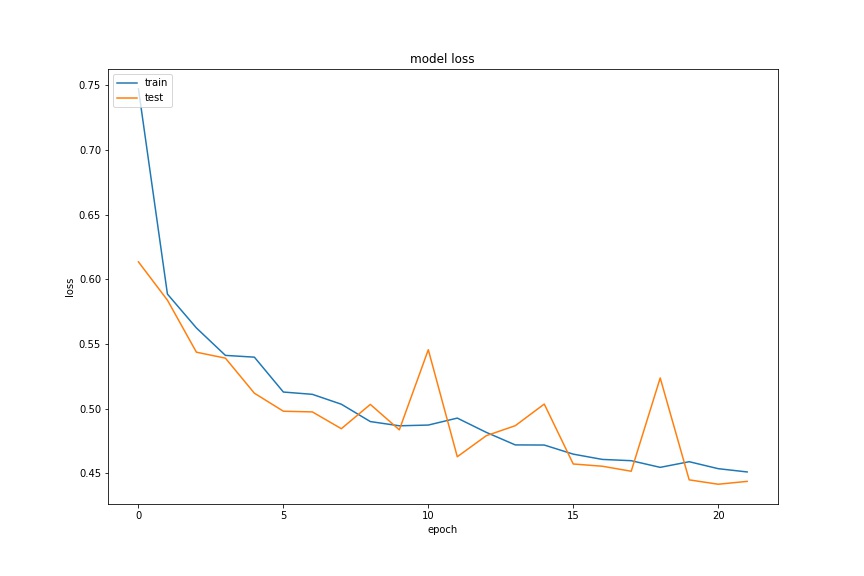
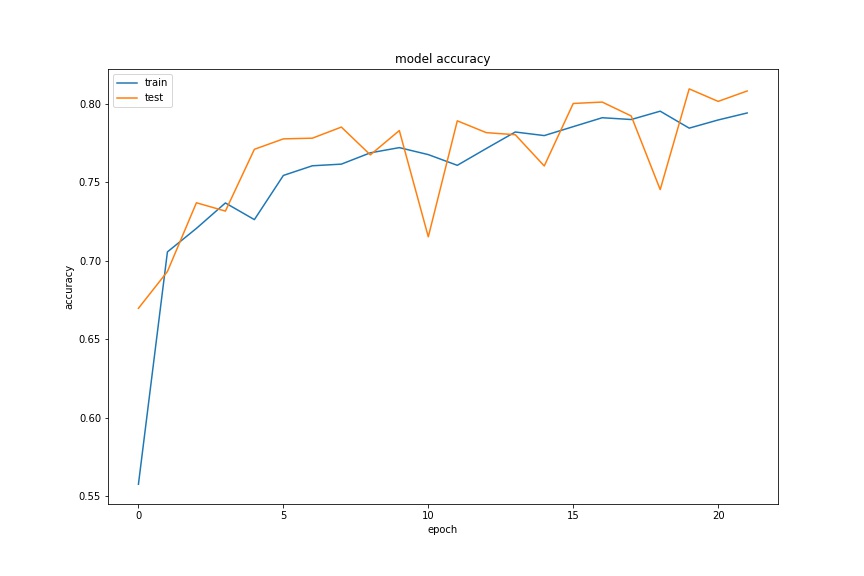
I perform data augmentation to increase the diversity of the dataset and to avoid overfitting. I rescaled the images by dividing 255. I used transfer learning for this classification. The pretrained model is VGG16 from Keras library. VGG16 is proposed by K. Simonyan and A. Zisserman from the University of Oxford. It was one of the top models that has high performance on the dataset ImageNet of over 14 million images from 1000 categories. I remove the last layer, freeze other layers to remain original weights, and add my last layer to classify 2 classes, male and female. I used Adam optimizer and categorical\_crossentropy for loss function. I then plot graphs of loss and accuracy for training and validation data using matplotlib.

**3. Results**

First, I run 50 epochs with the batch size of 50 and 0.005 for learning rate on my Laptop’s CPU. I find out that it is extremely slow, so I switch to Google’s Colab for GPU and see a significant improve in performance. However, the result is not what I desired as it is below 80% for accuracy. After tuning and tweaking the parameters by increasing more epochs, changing to 0.001 for learning rate and changing batch size to 200, the performance greatly improves. I learn that for GPU, the large batch size speeds up the training process.



The accuracy increases to 80.81% for validation and 79.41% for training. The loss function gradually decreases. Although accuracy for validation increased, it did fluctuate.



**4.** **Conclusions and future works**

Through this project I have learned many things, improved my skills and gained more experiences. I believe, with more time, I will improve the program for an even higher performance. I think to make the model learn better, I will leave more layers of VGG16 to be trainable, then add my other four layers. Doing this way, the model has more chance to learn its new task and recognize the pictures better. I think the advantage of transfer learning is that it does not have to be trained from the beginning. The early layers can be relied on the pretrained model because the general and low-level patterns appear at this stage. Thus, it will save us a lot of time. Right now, I am implementing another method for gender classification, which is Multilayer perceptron (MLP). It was in used before CNN came to birth. I would like to know the detail and how it will perform differently from CNN. Beyond that, this project evoked my interest in Computer Vision. In the future, I would like to develop an application to detect faces on videos then guess gender and age. I will create my own convolutional neural network and train it from scratch. I hope with the growth of technology and other creative minds, there will be more amazing products and applications that benefit the community and the world.

**References**

[1] Chauhan, Nagesh Singh. “Predict Age and Gender Using Convolutional Neural Network and OpenCV.” *Medium*, Towards Data Science, 1 Aug.2019,towardsdatascience.com/prepared-age-and-gender-using-convolutional-neural-network-and-opencv-fd90390e3ce6.

[2] Chris. "How to Create a Basic MLP Classifier with the Keras Sequential API." *MachineCurve*. N.p., 27 July 2019. Web. 06 Dec. 2019.

[3] Gil Levi and Tal Hassner.*Age and Gender Classification Using Convolutional Neural Networks.* IEEE Workshop on Analysis and Modeling of Faces and Gestures (AMFG), at the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, 2015.

[4] Samer Hijazi, Rishi Kumar, and Chris Rowen. “Using Convolutional Neural Networks for Image Recognition”. IP Group,Cadence. <https://ip.cadence.com/uploads/901/cnn_wp-pdf>

[5] Schaijk, Tom Van. "Image Classification: MLP vs CNN." *Image Classification: MLP vs CNN*. N.p.,12 June 2018. Web. 06 Dec. 2019.

[6] "VGG16-Convolutional Network for Classification and Detection." *VGG16-Convolutional Network for Classification and Detection*. N.p., 21 Nov. 2018. Web. 06 Dec. 2019.