ECE324

lockdin: Identifying Engagement in Zoom Audiences

Final Report

Word Count: 1969

Due on December 6, 2020

Professor J. Rose

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

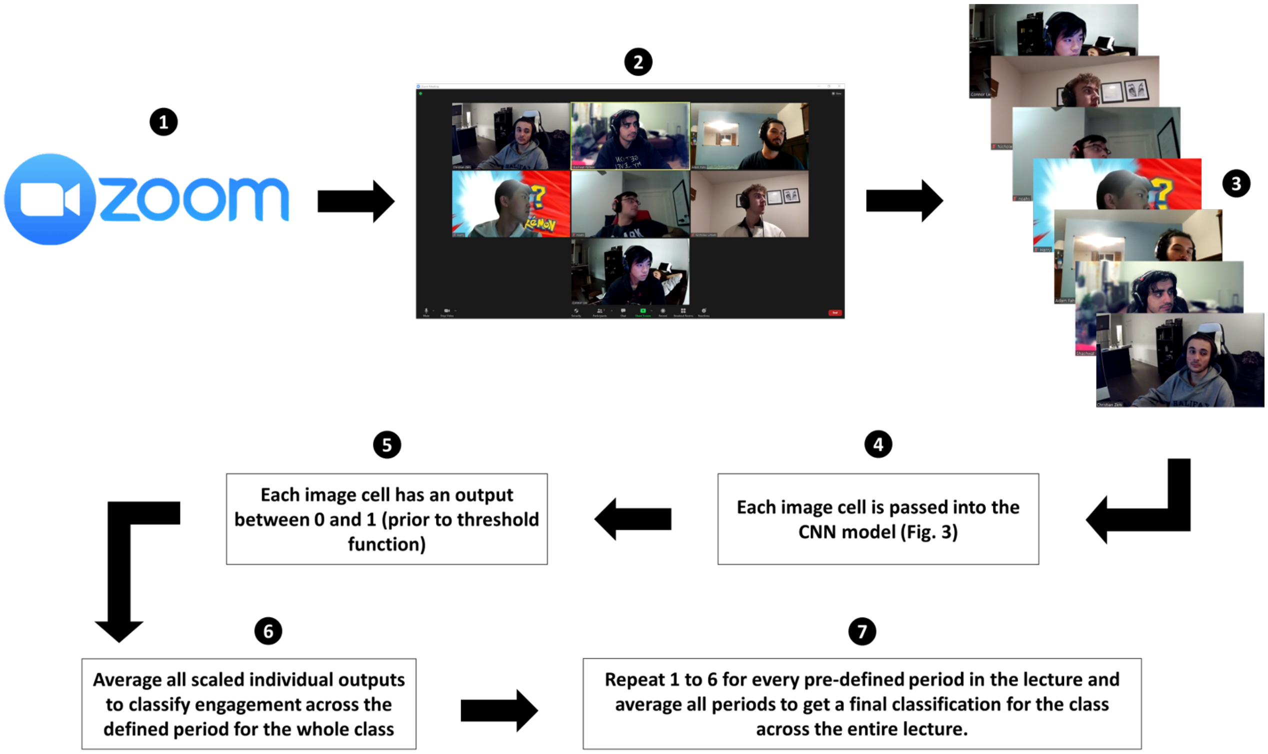
Geoffrey Jiang	**Shashwat Panwar**	**Christian Zeni**
1005014921	1004168839	1004880346

Introduction

Due to the pandemic, online learning has taken over the world and has become a mandatory part of almost every student and professor's lives. As it currently stands both professors and students are still trying to figure out the best way to teach/learn. Lockdin, will create a tool that will offer professors a metric to assist them in creating an engaging learning environment. Given the complexity of human emotion coupled with the fact that no two people express themselves in the same way, measuring engagement is not an easy task. In addition, engagement is a combination of many different features making it hard to categorize perfectly. Having said that, we choose to approach this problem with a neural network as fundamentally it can create its own categories. Thus, we want to leverage the fact that neural nets can identify patterns on their own, something that other approaches cannot provide.

Illustration

An overview of the actual process behind the idea is laid out in **Fig. 2**. It starts off with a periodic (predefined by user i.e. every X minutes) snapshot of a Zoom lecture from the perspective of the professor in gallery view which is fed into the PIL based cropping script described in **Fig. 1**, giving us individual images to be fed into our now trained CNN model giving us a single binary classification output for each person in the screenshot. This is where the process stops with regards to what we have coded for the scope of this course. The next addition to the application involves averaging the output values for each individual in order to give an overall average classification for the class' engagement level. This process is completed once every predefined period (i.e. once every minute) and then averaged over all periods to get a final engagement scale for the whole zoom lecture.



4 Each image cell is passed into the CNN model (Fig. 3)

5 Each image cell has an output between 0 and 1 (prior to threshold function)

6 Average all scaled individual outputs to classify engagement across the defined period for the whole class

7 Repeat 1 to 6 for every pre-defined period in the lecture and average all periods to get a final classification for the class across the entire lecture.

Figure 1: 7 step outline of the entire process encompassed by the overarching idea behind lockedin

Background and Related Work

As outlined by "Engagement detection in online learning: a review" there have been many approaches to determining the engagement of online students. One of these studies, "DAiSEE: Towards User Engagement Recognition in the Wild" used three different CNN models, for a multi-classification of boredom, engagement confusion and frustration at four levels, very low, low, high, and very high. A different approach is seen in, "The faces of engagement: Automatic recognition of student engagement from facial expressions", which applied three different machine learning techniques (Boost, Support Vector Machine and MLR) to detect engagement based on facial expressions [1].

Data Collection and Processing

Starting off, we looked at different pre-existing datasets for facial and body expression. Notable datasets that we found were the Yale Face dataset as well as the Labeled Faces in Wild dataset. However, as discussed in previous sections, because engagement is not as 1 dimensional as the labels that these datasets contain, we opted to fully collect our own data. To do this, we created a list of features that could potentially classify someone as engaged or unengaged. This list is shown below with some example photos in **Table 1**.

In total, each participant posed in around 12 positions each, all of which were combinations of the above features. In doing so we were able to gather 281 photos in total, which we then labeled in accordance to [1].

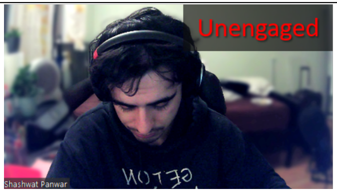
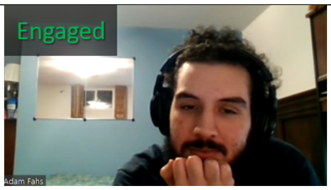
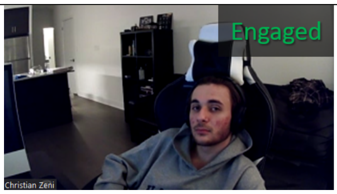
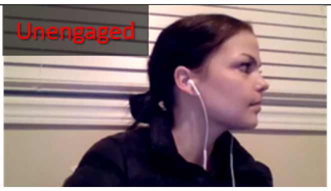
Feature/Expression	**Example photo**
Gaze direction	
Hands on face	
Lean	
Body Orientation	

Table 1: Some key features pertaining to an engaged vs unengaged classification

From here, we began processing and augmenting the photos to build up our complete dataset of which we would use in our model's training/testing. (Add stuff about resizing, how we decided on what size to land on etc). After resizing all of our photos to our final size, we began the process of expanding our data by augmenting the data. To accomplish this, we used blur, horizontal flipping, as well as brightness adjustments for a total of 5 augmented photos for each original photo. We experimented with more combinations however we found that the model because memorizing too heavily so we kept our augmentation to these 3 main methods.

In order to obtain the individual images of every person in a single screenshot, we developed a cropping script where the input is a window sized screenshot and splits it into cell images as per **Fig. 2**. This was done using the PIL library [2], specifically the cropping box tool which allowed us essentially parse the screenshot and extract a set of pixels that matched up with a cell. This process was enumerated for the different screenshot sets we collected. All images were resized to 260 x 145 pixels prior to labelling, because that was the smallest cell size that was output from the cropping script. No distortion occurred in the resizing process because we discovered the ratio of Zoom call cells remains consistent regardless of how many people show up in gallery view.

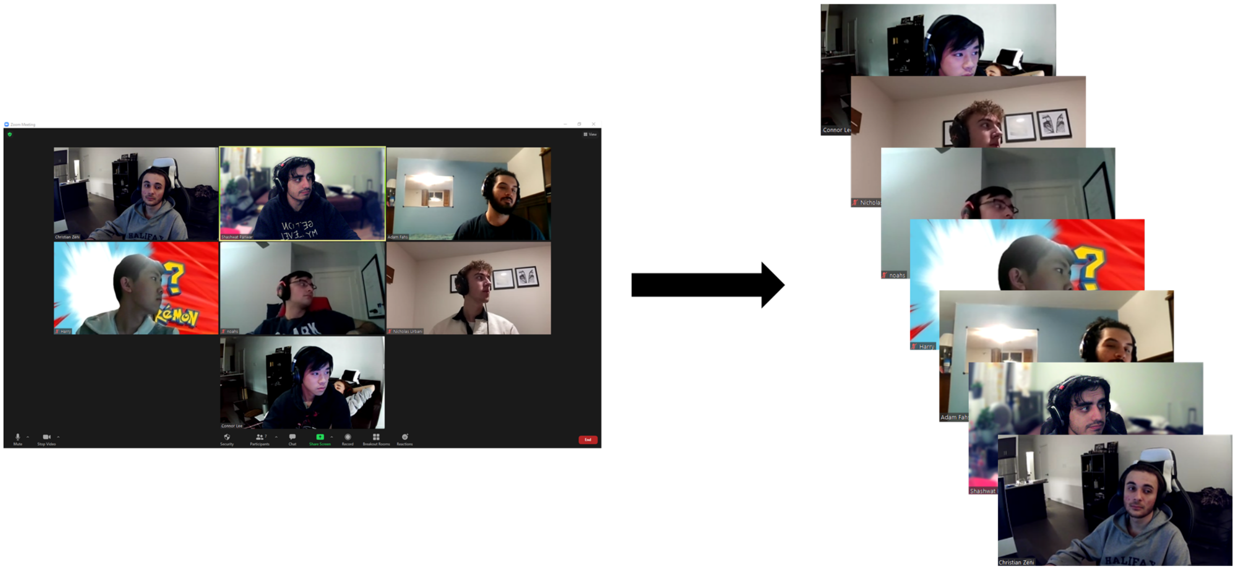


Figure 2: Outline of the cropping script process built using the PIL library

Architecture

The final architecture was a two layer CNN followed by one hidden layer. Each convolutional layer was followed by a ReLu activation function and a (2,2) max pool. The hidden layer was also followed by a ReLu activation and the final output was passed through a Sigmoid activation. More details regarding the final architecture are depicted in Figure 3. The model was trained with the following hyper-parameters, batch size = 30, learning rate = 0.01, epochs = 44, loss function = BCE, optimizer = SGD.

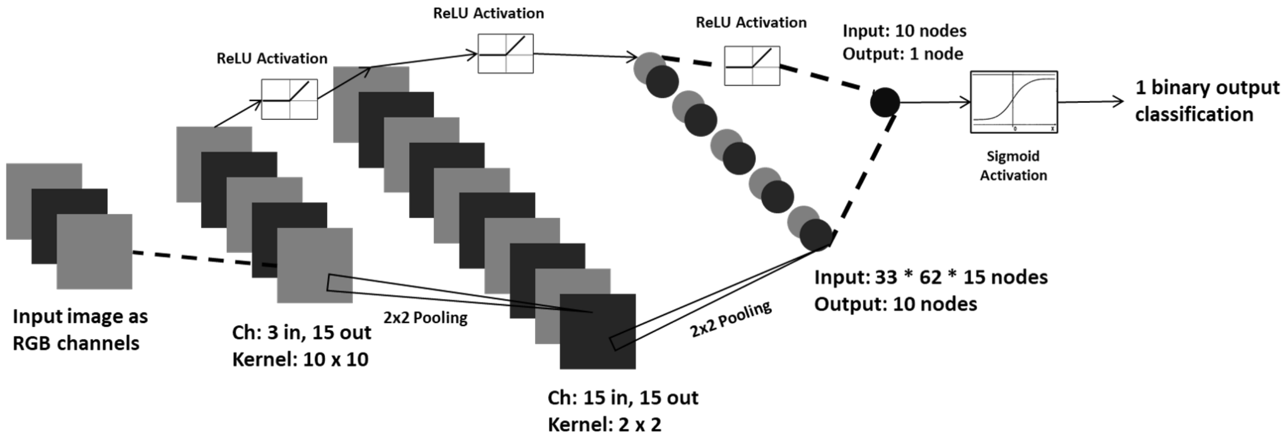


Figure 3: Layered architecture of the final CNN model that was settled on after numerous experiments

Baseline Model

For our baseline model, we chose to use a single hidden layer MLP, we made this decision based on past work that has shown success with this architecture [3]. In addition, it is a very simple architecture that is capable of handling a classification problem and easy to test. The exact architecture is shown below:

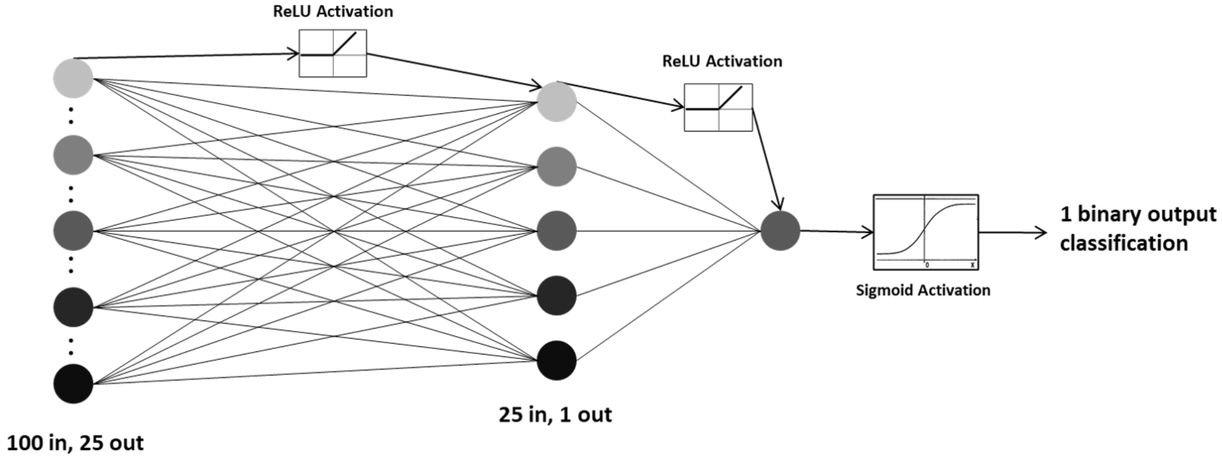


Figure 4: Layered architecture of the final Baseline model that was settled on after numerous experiments

Quantitative Results

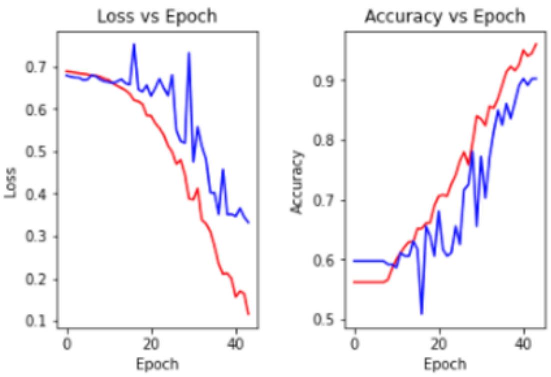


Figure 5: Training (red) and Validation (blue) results from our best CNN model architecture

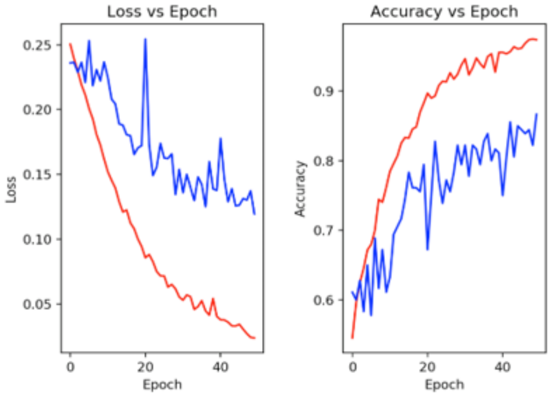


Figure 6: Training (red) and Validation (blue) results from our best baseline (MLP) model architecture

Metrics	**CNN Model**	**MLP Model**
Training Accuracy	96.1%	97.4%
Training Loss	0.12	0.024
Validation Accuracy	90.3%	86.7%
Validation Loss	0.33	0.119
Test Accuracy	86.3%	86.3%
Test Loss	0.526	0.121

Table 2: Best CNN vs MLP Results Quantified

The numerical metrics we used to compare the models are listed in **Table 2**. The relatively simple MLP model used as our baseline produced competitive results with respect to our CNN but the MLP consistently showed signs of overfitting very early on in the majority of our experimental runs. Due to this, the MLP was most likely “memorizing” rather than “learning” about the images themselves. Using only the test and validation accuracies would lead a user to believe that the MLP is not much worse at this task than the CNN but combining more metrics into the comparison (such as loss) allow for more fair analysis as noted by the signs of overfitting done by the MLP. Due to the classification being binary, confusion matrices as metrics are essentially redundant because of what is already tabulated above.

Qualitative Results

Given that the final model is a CNN we wanted to look at the feature maps. As demonstrated in Figure 7 (Pictures on the feature maps slide), the feature maps seem to identify a person's location in the frame quite well. However, it does not recognize eyes and eye position very well if at all. This is an issue because where a student is looking has a large role in whether they are engaged or not. In addition, the feature maps show that background items such as frames on the wall or chairs are also being "picked up", which is unnecessary noise and can lead to issues with proper classification.

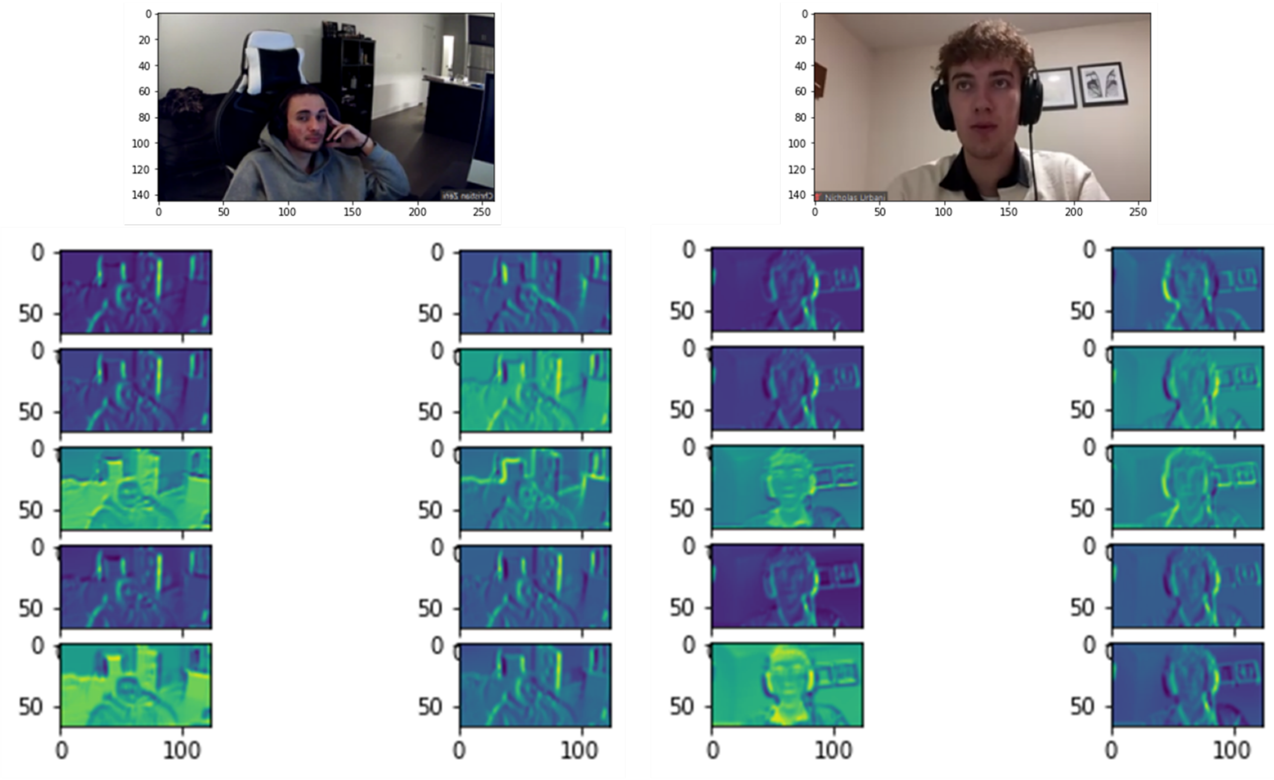
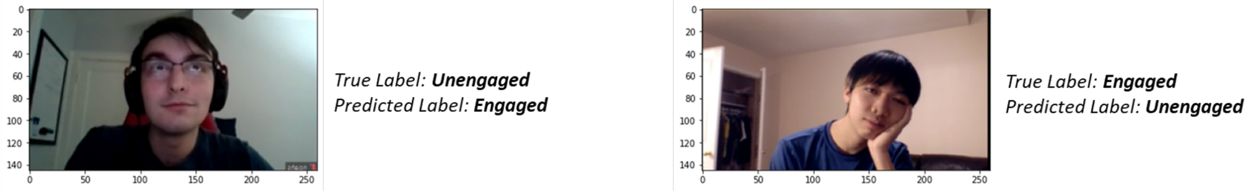


Figure 7: Feature maps and their respective images for the final CNN

We also thought it informative to examine photos that the CNN classified incorrectly. In Figure 8 (the ones below) we can see that pictures that are ambiguous, meaning an argument could be made for engaged or disengaged, are difficult for the CNN to classify correctly.



True Label: ***Unengaged***
Predicted Label: ***Engaged***
True Label: ***Engaged***
Predicted Label: ***Unengaged***
0
20
40
60
80
100
120
140
0
50
100
150
200
250

Figure 8: Inputs that were classified incorrectly by the final model

Discussion and Learning

With the results of the project, we can conclude that for the majority of cases, our model will be able to accurately predict engagement. We were also able to account for and adapt to different settings, expressions, and variations of expressions of engagement between a number of participants in order to properly simulate a real zoom meeting. However, one thing we must keep in mind is that with still frame captures of individuals, we may catch them looking away for a split second or in some momentary unengaged position. Therefore, to provide an accurate prediction, the most effective way that our model could be used is to take images over several minutes at regular intervals and determine the average engagement over that period.

We believe that our model is definitely performing better than expected based on the variations in data that we fed into it and the large number of total features that it would need to keep track of in order to general a guess. One interesting part of our results which came from our feature maps was the network's tendency to make guesses based on the overall orientation of the head with respect to the shoulders. We found that many feature maps put a lot of emphasis on this whereas we guessed that the model would rely on the positioning of the participant's gazes to make predictions.

If we were to extend this project, I believe that we could attempt to see if we can come up with a categorical classification of engagement rather than a binary one to encompass different levels of engagement. In addition, one thing that we learned during data collection is that we need to set better constraints on what types of data we should be collecting and more rigorously tracking the quality of our collected data and therefore what is ultimately feeding into our network. During the training, we found various photos that were not concretely in one classification over the other as well as some with features that may lead our model astray (such as some conflicting features in the background of some pictures). As a future step, we may implement a feature that removes the background or parts of the background similar to the functionality in Zoom to obtain a better focus on the characteristics of the individual. In addition, during data augmentation, we learned to not augment the data too many times with similar augmentations as it just led to more drastic overfitting and poor results.

Ethical Framework

There are many stakeholders in the implementation of our features, these include the professors/hosts of the zoom meetings, the students/attendees, the company Zoom itself, as well as any other third party companies like universities themselves that are involved. However, for the sake of this project, the main stakeholders that we will focus on will be the students and professors in this setting.

For **students that will be analyzed during Zoom lectures**, the implementation of this into their zoom meetings could have an impact on the autonomy and nonmaleficence of the students. The autonomy of the students is impacted as they are not given a choice as to whether their videos would be used to measure engagement. Furthermore, taking into account the fact that at some institutions like UofT, video is optional, meaning that some students may even be discouraged from turning on their videos due to these new measures. Furthermore, in terms of nonmaleficence, students may benefit from the imposed measures as some professors may take this live feedback into account and try to make their content more engaging. However, the opposite effect may also occur in that professors may accuse the students of not paying attention and penalize them somehow.

For **professors that will implement lockdin**, Increases beneficence as it gives them a tool which notifies them if their current material and delivery is helping students. This gives them the ability to try different teaching approaches and determine if they are effective, as well as giving them insight on when they need to make changes. Overall, it's a tool that can help them improve their craft. On the other hand, it also decreases justice for the professors. This is because a student can be disengaged for reasons other than being disinterested in the material or the way it's presented. However, lockdin does not know this and would tell professors that their class's engagement level is low. This may lead professors into believing that they are not doing their job correctly even though it is out of their control; which is unfair.

Navigating Github Repo

- Models: *models.py*
- Classifier/Training: *main.py*, *lockedin_tools.py*, *setcreation.py*, *accuracy_calculator.py*, and *NNFullModel.py*
- Cropping and Resizing: *cropping_script.py*
- Augmentation: *DataProcessing.py*

Permissions

Member	**Permission Indicator**		
Geoffery	Post Video/Demo - Yes	Post Final Report - Yes	Post Source Code - Yes
Shashwat	Post Video/Demo - Yes	Post Final Report - Yes	Post Source Code - Yes
Christian	Post Video/Demo - Yes	Post Final Report - Yes	Post Source Code - Yes

Table 3: Group member permissions

References

[1] Dewan, M.A.A., Murshed, M. Lin, F. “Engagement detection in online learning: a review”.
Smart Learn. Environ. 6, 1. 2019. Accessed Oct 25th, 2020
https://doi.org/10.1186/s40561-018-0080-z

[2] Clark, Alex. “Pillow - Module”.
Pillow.readthedocs. Accessed Nov 24th, 2020.
https://pillow.readthedocs.io/en/stable/reference/Image.html

[3] Kaggle Competition. “Emotion Detection From Facial Expressions”.
Kaggle. 2016. Accessed Oct 26th, 2020.
https://www.kaggle.com/c/emotion-detection-from-facial-expressions