WE WOULD LIKE TO THANK OUR ADVISOR PROFESSOR VISHAL PATEL FOR GUIDING US AND PROVIDING THE RESOURCES AND ADVICE NEEDED FOR OUR PROJECT.

For our algorithms to work accurately and in a timely manner, input images must be scaled down. The scaled down images introduced a difficult challenge because the poor resolution resulted in worse accuracy by the TensorFlow model. The height at which images were taken also created a similar problem, as the height at which the camera is held increases, the clarity of the image decreases, again, resulting in greater inaccuracy.

The part of the algorithm used to detect the location of coins, as shown Figure 2, is made up of existing computer vision algorithms provided by OpenCV, given in Figure 1.

Each coin is cropped out of the resized image and passed to a re-trained TensorFlow model called Inception V3.

Over 29,000 images were used to retrain the model to identify coins. The majority of the images were frames from video recordings of different coins at varying angles and lighting conditions. The variation of these videos is how the algorithm accounts for different environmental conditions.

The results outputted from the TensorFlow model are tracked to determine the sum of the coins.

The accuracy and time efficiency of the app was tested against various human participants. As seen in Figure 3, as the # of coins increases, the time to count those coins also increases for both the app and the human participant. There is an increasing linear relationship between the amount of time it takes for a human participant to count how many coins versus the # of coins. For the app, the relationship starts off linear and then there is a logarithmic relationship near 35 coins indicating that as the # of coins increases, the amount of time it takes for the app to count the amount of money begins to level off. Based on these results, we can see that around 35 coins, it takes less time for the app to calculate how much money there is versus the human participant.

Percent error for the app is random and there is no relationship to the # of coins because the inputs vary based on lighting, angles, and heights. However, for a human participant, there is a lower percent error for a lower # of coins. As the # of coins increases, the amount of percent error increases for the human participant. These results are given in Figure 4. Overall, the app percent error can be minimized with additional training of the neural network. Regardless, we have a hypothesis that as the # of coins increase, there will eventually be a point where the percent error for the app is less than that of a human participant, especially with more training.