MAIS 202 - Deliverable 3

1. Final Training Results

Final Training Results We've made several enhancements to our model since our last update. Previously, we faced challenges with computational capabilities and accuracy, especially when increasing the number of categories. We discovered that our model was trained with an excessive number of samples (around 150,000). As a result, we reduced this number to 5,000, as our program only needs to achieve a certain level of accuracy while the user is drawing to identify the category of the input.

Our current model is equipped with a 2D convolutional layer, a max pooling layer, a flatten layer, and a dense layer, which is a significant upgrade from our previous model. This has notably improved our accuracy. However, due to computational limitations (with one epoch estimated to take 16 hours), we opted to run our project on an NVIDIA 10 GPU on Google Colab. We achieved impressive results after just one epoch, with about 95% accuracy across the 345 categories. To conserve resources, we halted further training.

Compared to our previous submission, this is a substantial improvement. Previously, we couldn't even train the model with more than 50 categories without our computer crashing. Moreover, as we increased the number of categories, our accuracy dropped significantly (around 50% accuracy with 50 categories). Therefore, we are extremely pleased with the progress we've made. Finally, we've implemented checkpoints to save our model's weights periodically during the training process. This functionality provides peace of mind by mitigating concerns about potential interruptions, such as Google Colab sessions ending unexpectedly due to inactivity or training halting from resource limitations.

2. Final demonstration proposal

We've spent considerable time discussing how to gamify our model. Our plan is to integrate it into a website. Regrettably, due to our busy schedules, we haven't had the opportunity to set it up yet. However, our model training is nearing completion.

Here's how it will work: Similar to Google's Quick, Draw, the user will be given a prompt to draw. As the user draws, our model analyses the input and compares it to the 345 categories. Based on its training, it will identify the category the drawing corresponds to. To differentiate our model from Google's Quick, Draw and add a creative twist, we've decided to incorporate a competitive element. Our idea is to compare two drawings made by users to see which one most closely matches the selected category, thereby determining the winner. Alternatively, we could see which drawing can be associated with a category the fastest.