EECS 605 Final Project Proposal: Hand Gesture Recognition Deployed on the DeepLens Edge Device

For my proposed project, I am planning on training and deploying a hand gesture recognition model to the DeepLens device. This model will be able to accurately predict how many fingers a person is holding up on one hand. Details on the relevant prior work, the dataset used, the deep-learning model structure, and deployment on a DeepLens device along with a Heroku app are provided in the following sections.

Prior Work:

With the idea of this project in mind, I looked to see if there were similar projects online and if they used any simple datasets of hand gestures. Somewhat to my surprise, I couldn't find anything that seemed like it would provide a good simple dataset of simple hand gestures. The closest I found was a dataset with 20,000 images of near infrared images called the "Hand Gesture Recognition Database" [1]. The infrared nature of these images made the hands stand out very clearly against any background, but any model trained on that dataset would not work to deploy on the DeepLens device because it uses a standard RGB camera. I was able to find a couple of articles that walked through training a CNN model to recognize hand gestures [2,3]. These models were trained on the previously mentioned infrared dataset and another similar dataset with RGB-D videos with distance information included. Such articles and likely more like them show there have been many projects done on this topic but perhaps not done with deploying on a DeepLens edge device.

Dataset:

As mentioned previously, I was not able to find a readily available dataset to train a deep-learning model for hand gesture recognition to be deployed on the DeepLens device. So, using the information learned in the EECS 605 class, I created a DeepLens project to capture images every 0.5 seconds, preprocess the images by cropping, converting to grayscale, resizing, and then upload them to an S3 bucket. Using this project, I created my own dataset of roughly 3,000 images to be used to train my deep-learning model. More details on creating this dataset and its challenges and potential shortcomings can be provided in the final report but a few example processed images are shown below:

Deep-Learning Model Structure:

While a project like this might have been a daunting computer vision project just a few years ago, with the advent of deep-learning, especially CNN based approaches, a project like this can be fairly straightforward. The data was already appropriately preprocessed so all that was left was to split the data

into train, validation, and test sets and then train with a specified deep-learning model. After testing several different models, I found that a CNN based approach with 4 convolutional layers and one dense layer followed by the final output layer gave the best performance on my dataset. The convolution layers use filter widths of 16, 16, 32, and 32 and have (3,3) convolution kernels. Each layer has ReLU activation except for the final output layer which has a softmax activation. The model was trained using TensorFlow's Keras and then exported to an ONNX model. More details on setting up and training the model and the challenges faced can be provided in the final report but the best validation accuracy was 95% with a test accuracy of 92%. The accuracy for smartphone images uploaded to the Heroku should be similar although it may vary depending on complicating factors such as the person's skin tone, drastically different lighting conditions, and possibly different image formatting issues.

DeepLens Deployment and Heroku App:

I still need to go through the proper steps to deploy my ONNX on the DeepLens device, but this should be fairly straightforward as it is nearly identical to the project of handwritten digits recognition but with a different model and only 5 potential numbers. Creating the Heroku app will probably be a bit harder just because everything with creating a special Docker environment and uploading that to a Heroku app was quite a pain for the related lab. This was because Docker kept crashing and there was very poor documentation for what we were trying to do with the Heroku app. Hopefully, doing this again won't prove as difficult and should provide a cool way to test out my trained hand gesture recognition model.

References:

[1] Leap Motion, "Hand Gesture Recognition Database", *Kaggle.com*, 2021. [Online]. Available: https://www.kaggle.com/gti-upm/leapgestrecog/version/1. [Accessed: 27- Mar- 2021]

[2] F. Borba, "Tutorial: Using Deep Learning and CNNs to make a Hand Gesture recognition model", *Medium*, 2021. [Online]. Available: https://towardsdatascience.com/tutorial-using-deep-learning-and-cnns-to-make-a-hand-gesture-recognition-model-371770b63a51. [Accessed: 27- Mar- 2021]

[3] Y. BENAFFANE, "[Deep Learning] Hand gesture recognition", *Medium*, 2020. [Online]. Available: https://becominghuman.ai/deep-learning-hand-gesture-recognition-b265f4e6cf02. [Accessed: 27- Mar- 2021]