## Arduino project group 8

Joris Faas 2985888

Tijn Hassing 3004619

Alexandru Lungu 3006301

Narendra Setty 2944200

Hoang Pham - 3002152

Our submission can roughly be split into 2 parts, training the model and running the system:

# Machine learning

our training the lenet5 model part explains the lenet5 model which is in ml.ipynb

our testing few shot details how to use the few shot model which is in ml(1).ipynb

### Training the lenet5 model

In order to run the train model part, you can either use the already trained model, called final\_model.zip inside the machine learning folder, and load it into the test part inside ml.ipynb, keep in mind that in the test part there is a path which has to be changed to your own test folder, which is also sent in the dataset.zip. You can run each file in order while changing the main directory path inside the image loader part of the ml.ipynb document to the downloaded dataset.

### Testing few-shot

For a few shot we are using a special created set of images divided in 5 folders each representing a letter and containing 10 images. Therefore the variable, main\_image\_dir should point to the location of the folder ‘few\_shot\_set’ on your laptop in order to load the correct images.

### Pruning and quantizing model

When you have saved a model after training you can prune and quantize it using the ‘Pruning and quantizing.ipynb’. In this file you first need to load the dataset to use for quantizing, here you need to change the file path to the path where you have the dataset stored. After this you need to shape the data and convert everything to grayscale. After this you can start on pruning where you need to load the model weights that you have saved (In our case ‘final\_model\_weights.h5’). After this you can specify for how many epochs you want to fine-tune your pruning for and what level of sparsity you want in the ‘pruning(0.5, 2, "with fine-tuning”)’ line.

After pruning is done and the pruned model is saved you can quantize it using the directory where the pruned model is saved (In our case final\_pruned\_model) and then specify in which directory you want to save the tflite file.

After you have the tflite file all you have to do is run this unix command ‘xxd -i model\_quantized2.tflite > model\_data.cc’ in a linux based terminal. Now you have a model which can be uploaded to the arduino.

### Running the application

After putting the model\_data.cc in the arduino sketch (which it already is in our submission) you can upload to the arduino, which will make it start broadcasting bluetooth signals. you can then find the device on the bluetooth settings page on your phone under the name “Nano 33 BLE HID Keyboard”. Once you choose to connect to it it will be listed as “Arduino” under linked devices. Now you can open a text-to-speech app or website and make a picture on the arduino which will send the corresponding letter to your phone after a couple of seconds.