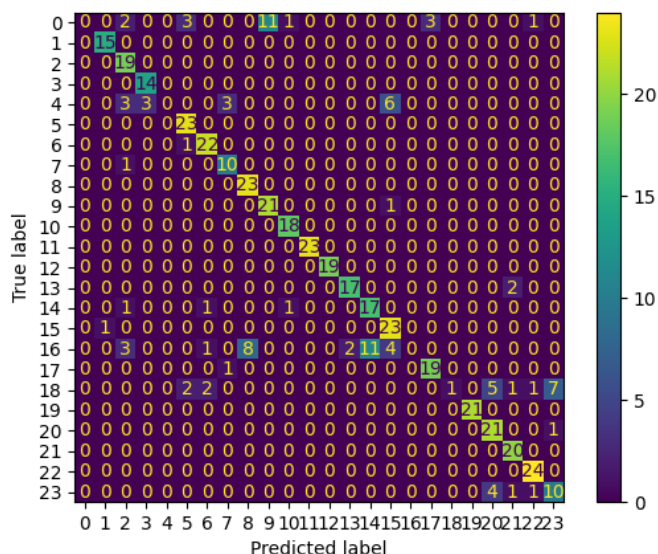


The progress on my project so far

Currently, my progress would include two major parts. One is the figure preprocessing in Golang and the other one is the image classification through neural network in Python via TensorFlow.

For the first part, the raw figure data was stored in a directory, with figure name labels as "A__", this is the segmented character from driver license. Since character "I" and "O" are not existed in the real word, the data would contain 2400 figures, with 24 labels. The first step is to read the input figure and use an array to store the pixels of each figure. Meanwhile, I converted the figure into grey format and normalize the pixel by dividing 255, ensuring each pixel is in the range of [0,1]. Additionally, I converted the first character of file name ("what figure represents") to numerical data and then a 1*24 matrix using one hot encoding, which would be considered as the label. I iteratively went through all the files and append to a 2d array, with shape equals 2400*1152 each array and I denoted as X. Similarly, we obtain all the labels with a 2d matrix, with shape (2400*24). To divide the shape for later training, I shuffled and divided the index of the previous 2400 data, ensuring the training data could be matched to the testing data.

The second part would use the existing neural network platform TensorFlow to conduct a rough image classification to prove this project is feasible. Based on the research, I found the existing TensorFlow of Golang package could not support the training part, so I conducted the previous suggested ideas on Python again. Then, I trained the data on a neural network with three layers, each containing the nodes 512, 128, 24 respectively. For the model compiler, I used the "SGD" as optimization method and mean squared error as the loss function. I then used a confusion matrix to visualize the classification performance. Since this trained model is not tuned on hyperparameters, the model thus doesn't reach its most accurate condition. The overall performance by using this simple three layers could achieve such results which would be acceptable for me.



Any challenges I have met so far

As previously suggested, I need to train the model in Python environment using TensorFlow, but when I tried to read the saved model, I got the encountered problem. I tried to search the Internet, but the resource is limited. In addition, the error information is a general question, many conditions could also raise this kind of scenario, I had no ideas how to debug this for now. Some possible other thoughts would be to convert the stored pixel into other size matrix, rather than flatten all the pixel into one array. Golang does not have convenient matrix manipulation method, therefore, this method could not be ensured to work in the end.

```
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2021-10-20 21:38:03.129281: I tensorflow/cc/saved_model/loader.cc:211] Restoring SavedModel bundle.
2021-10-20 21:38:03.165841: I tensorflow/cc/saved_model/loader.cc:195] Running initialization op on SavedModel bundle at path: model/my_model
2021-10-20 21:38:03.172282: I tensorflow/cc/saved_model/loader.cc:283] SavedModel load for tags { serve }; Status: success: OK. Took 68685 microseconds.
panic: op serving_default_inputs_input not found

goroutine 1 [running]:
github.com/galeone/tfgo.(*Model).Op(0x4243780, {0x42ce75d, 0x1c}, 0x0)
    /Users/yuxuan/coding/go/pkg/mod/github.com/galeone/tfgo@v0.8.0-20210914072119-caabdf338ee3/model.go:80 +0x185
main.main()
    /Users/yuxuan/coding/go/src/finalProj4Program/main.go:56 +0x154

Process finished with the exit code 2
```

For the following part, once the read trained model in go version's TensorFlow, I would focus on building the neural network from scratch in Golang step by step. For instance, matrix helper, including the basic matrix manipulation, some

components including the design of activation function, loss function and possible optimization method.