System description and user guide

# Overview

This project is to implement a simple system which aims to recognize the hand written numbers from grayscale image. The entire project is implemented with Python and Tensorflow. There are three major python files in the main project directory:

* model\_def.py: it is the definition of the neural network, which will be used by both the training and inference.
* mode\_train.py: it initializes the model defined in model\_def.py, then trains it. The MNIST data will be automatically downloaded and being read by the training. After the training, the checkpoint of the model will be saved in “project\_directory/model/final”.
* mode\_inference.py: it launches the model defined in model\_def.py, then run it and generate the result. In particular, it will use the checkpoints frozened in “project\_directory/model/final” and setting up the parameters for the model.

# Prerequisites to run the training and inference

You need to have the following installed:

* Python 3.6.2
* Tensorflow (cmd: “pip3 install --upgrade tensorflow”)

# How to train the model

The process is simple. Open a command prompt, go to your project directory, then run “py model\_train.py”. The script will automatically download the MNIST data set then start training. The number of training iterations is set up as 5000. If you would like to change it, please open “project\_directory/config/config.yml”, then change the item of num\_iterations to whatever number you would like to be. The checkpoints will be saved for every 1000 training iterations. After the training is done, the final checkpoint will be stored in “project\_directory/model/final”, which will be used by inference. Please note that a sample checkpoint has been uploaded to the github repo for user’s convenience.

# How to do model inference

Running the model to address your real world problem is simple as well. Assume you have trained the model and the checkpoint has been stored in “project\_directory/model/final”, open a command prompt, go to your project directory, then run “py model\_inference.py --i path\_of\_your\_input\_image\_file”. For user’s convenience, we have uploaded several sample images into “project\_directory/test” folder. E.g., if you type “py model\_inference.py --i test\3.jpg” and run, then you would expect something like “The recognized result is 3 and the probability is xyz ”.

Please note that we only support JPEG, PNG, GIF and BMP formats. Please DO NOT input any raw rgb/yuv image. There is no restriction about image resolution. Since the MNIST data set is composed by grayscale image with white number and black background, so the trained model can only recognize the white number in the black background. We strongly suggests to provide the input image with the same features (grayscale, while number and black background).