For Source code,

In “preprocess\_data\_and\_train\_model\_from\_scratch.py” file,

Just like what its name says, it preprocesses the original data (data in “action” folder) for better being taken as input data for training DNN models and trains and tests and saves the pose classification DNN model. It prints the training history and plots the confusion matrix of the model for better understanding its performance.

Datasets:

I divided the original data (data in “actions” folder) into Testing, Training and Validation Datasets. Their proportion relations are:

(For the whole dataset, 15% Testing, 85% Rest)

(For the Rest, 15% Validation, 85% Training)

Usage:

Create a virtual environment, download all the related libraries and modules it needs according to the imports in the beginning of the script, put the original dataset (folders names and structure should be like “actions” folder, heatmap .npy files names can be any) right under the project, same level of the script, and run it to get the results.

It will generate a “data.csv” file which combine all the original data together for better training the model and a saved model called “weights.best.hdf5”, it also prints the training history and plots the confusion matrix of the model showing its accuracy on testing dataset.

In “use\_saved\_model\_and\_predict\_or\_test.py” file,

It loads saved model and use the loaded model to make new predictions (in single heatmap .npy file), it also tests the loaded model on new dataset, it will plot the confusion matrix of the model on new dataset.

Usage:

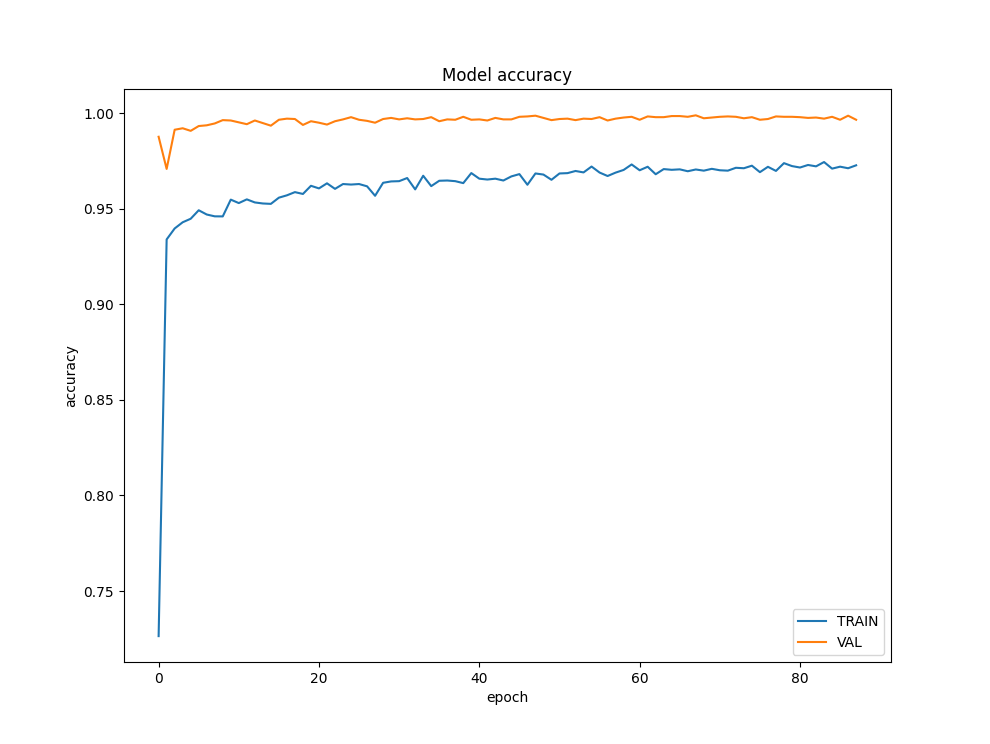
Put the saved model and new dataset folder (names and structures should be like “actions” folder, heatmap .npy files names can be any) right under the project, same level of the script, and run it to get the results.

For trained model,

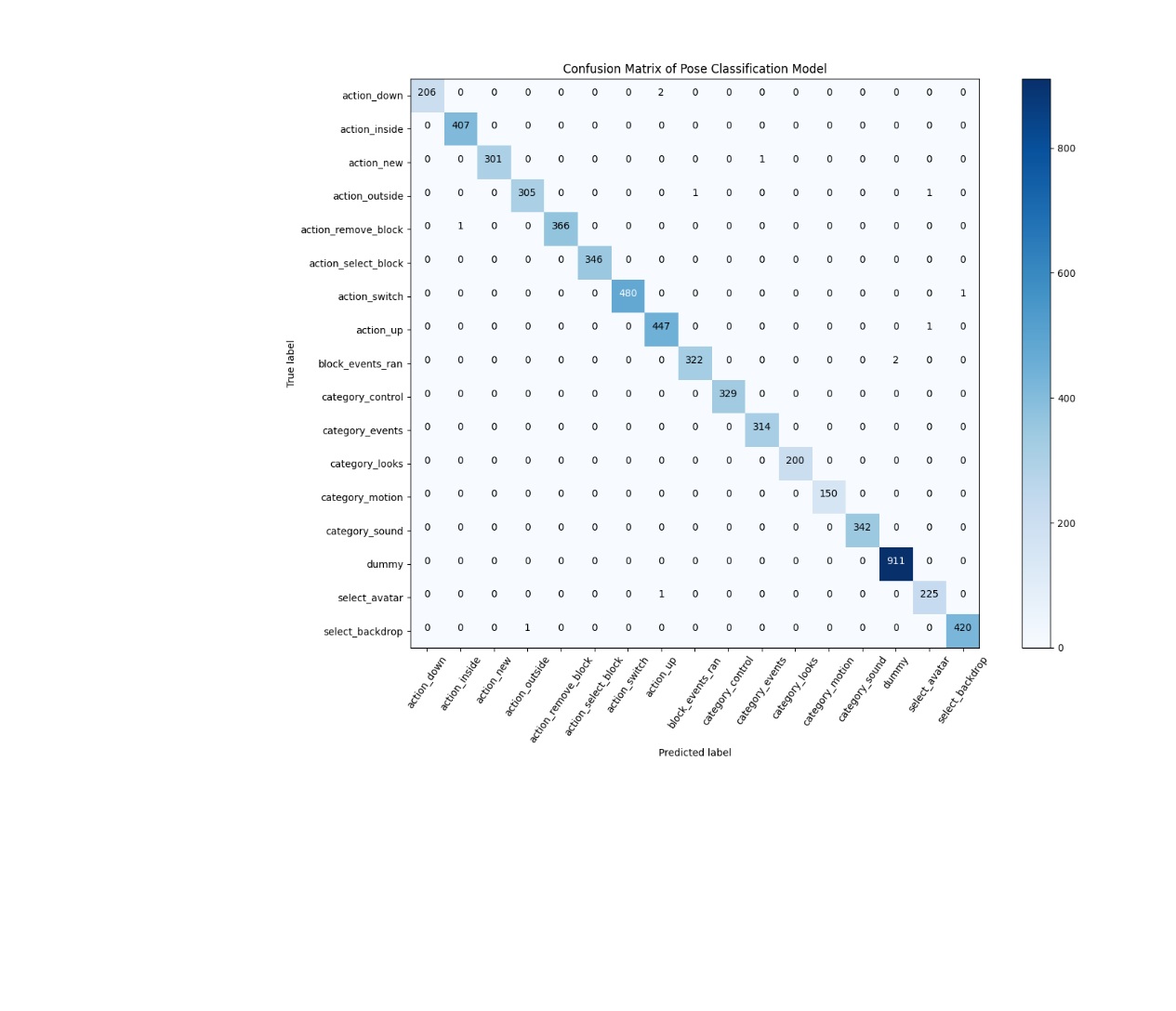
My trained model is “weights.best.hdf5” in Trained model file, it can be loaded and tested and can make new predictions in “use\_saved\_model\_and\_predict\_or\_test.py” python script in Source code file.

Its accuracies are: 97% (Training Dataset), 99.65% (Validation Dataset), 99.80% (Testing Dataset)

Its training history:



Its confusion matrix on testing dataset:



The model is a fully connected neural network with one dense layer of 256 nodes followed by a Dropout layer. It shows as below:

