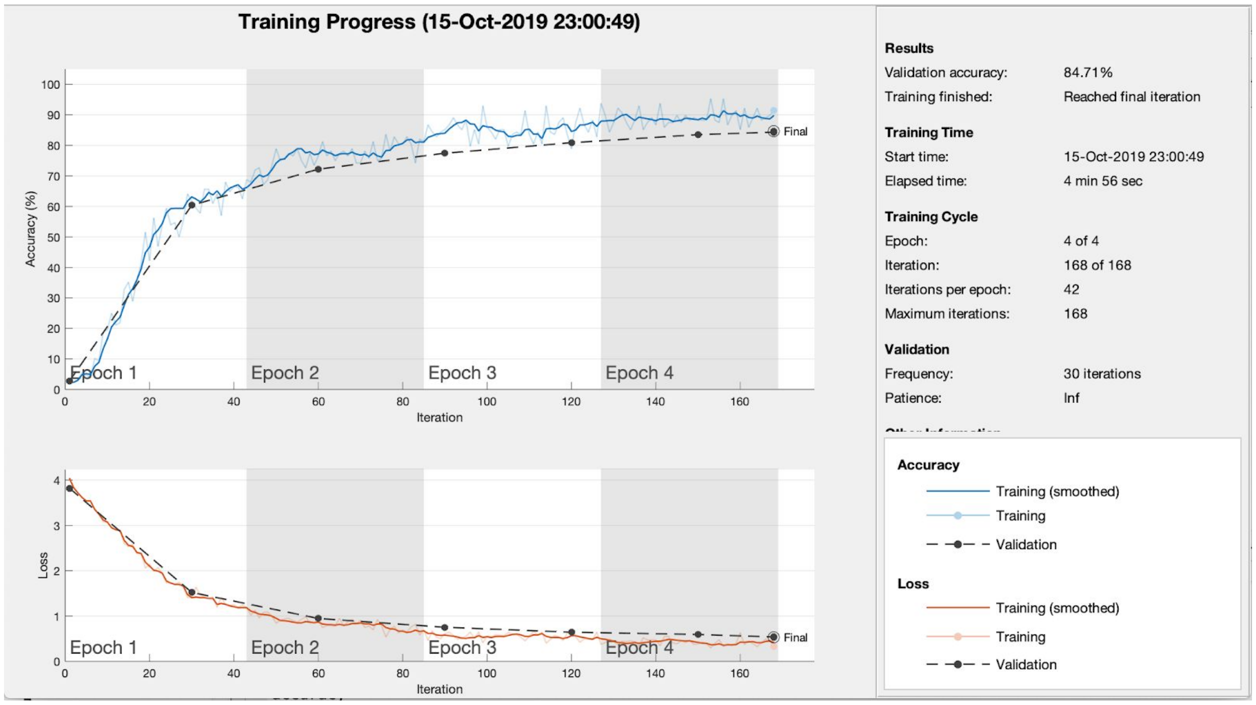
Yanisa Sunthornyotin

MATLAB result

The result from example code



Results

Validation accuracy:	84.71%
Training finished:	Reached final iteration

Training Time

Start time:	15-Oct-2019 23:00:49
Elapsed time:	4 min 56 sec

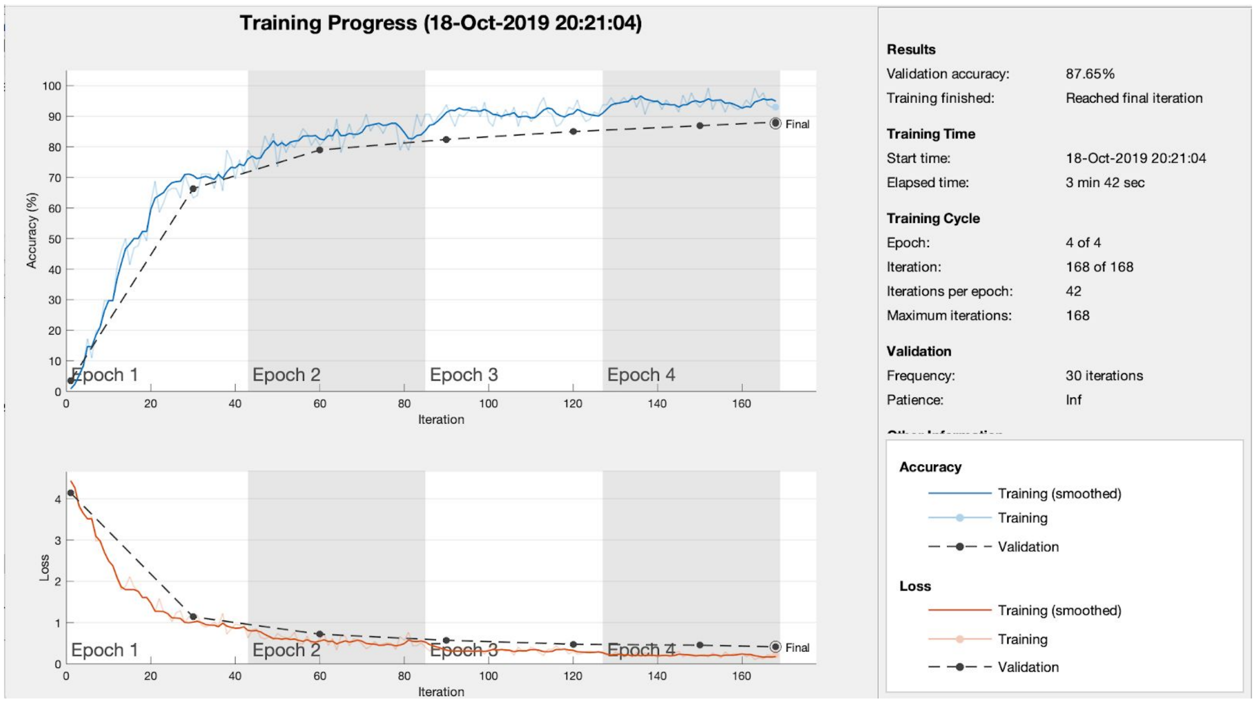
Training Cycle

Epoch:	4 of 4
Iteration:	168 of 168
Iterations per epoch:	42
Maximum iterations:	168

Validation

Frequency:	30 iterations
Patience:	Inf

After adding leakyReluLayer and more convolution2dLayer



Results

Validation accuracy:	87.65%
Training finished:	Reached final iteration

Training Time

Start time:	18-Oct-2019 20:21:04
Elapsed time:	3 min 42 sec

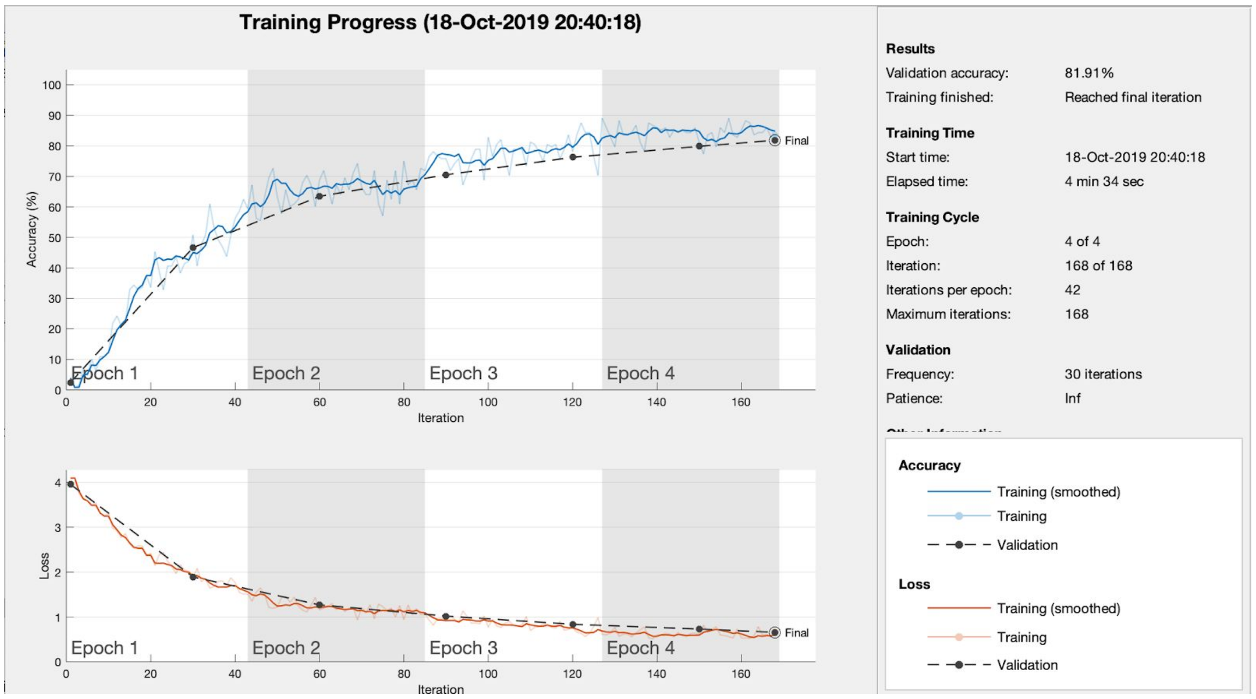
Training Cycle

Epoch:	4 of 4
Iteration:	168 of 168
Iterations per epoch:	42
Maximum iterations:	168

Validation

Frequency:	30 iterations
Patience:	Inf

After changing from ReLu to Tanh



Results

Validation accuracy:	81.91%
Training finished:	Reached final iteration

Training Time

Start time:	18-Oct-2019 20:40:18
Elapsed time:	4 min 34 sec

Training Cycle

Epoch:	4 of 4
Iteration:	168 of 168
Iterations per epoch:	42
Maximum iterations:	168

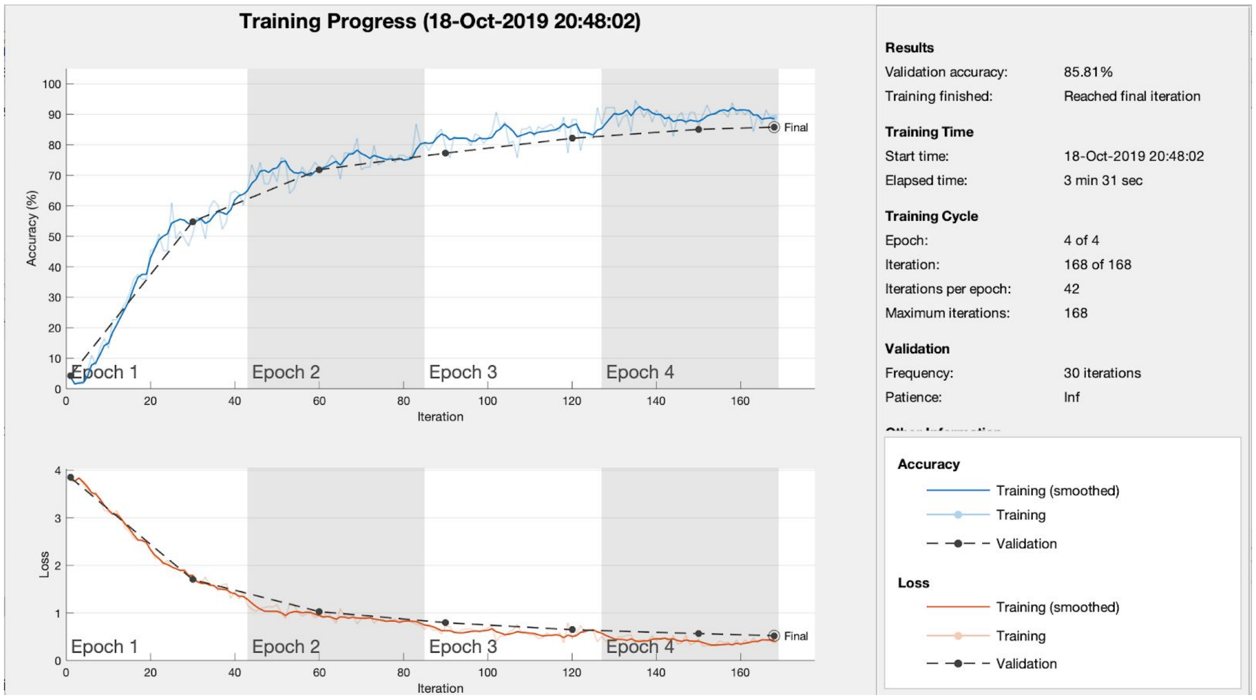
Validation

Frequency:	30 iterations
Patience:	Inf

Accuracy
- Training (smoothed)
- Training
- Validation

Loss
- Training (smoothed)
- Training
- Validation

After adding Dropout to one of the layers in the original architecture



Results

Validation accuracy:	85.81%
Training finished:	Reached final iteration

Training Time

Start time:	18-Oct-2019 20:48:02
Elapsed time:	3 min 31 sec

Training Cycle

Epoch:	4 of 4
Iteration:	168 of 168
Iterations per epoch:	42
Maximum iterations:	168

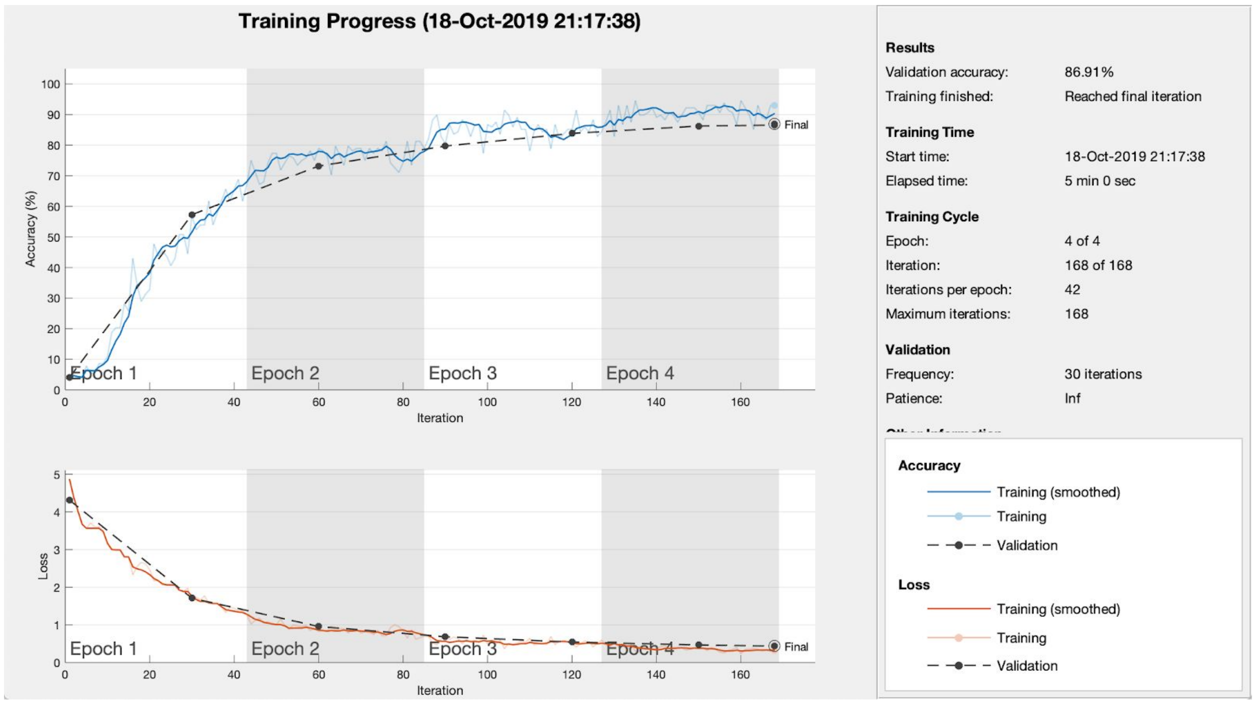
Validation

Frequency:	30 iterations
Patience:	Inf

Accuracy
- Training (smoothed)
- Training
- Validation

Loss
- Training (smoothed)
- Training
- Validation

After adding crossChannelNormalizationLayer set to 3



Results
Validation accuracy: 86.91%
Training finished: Reached final iteration

Training Time
Start time: 18-Oct-2019 21:17:38
Elapsed time: 5 min 0 sec

Training Cycle
Epoch: 4 of 4
Iteration: 168 of 168
Iterations per epoch: 42
Maximum iterations: 168

Validation
Frequency: 30 iterations
Patience: Inf

Accuracy
Training (smoothed)
Training
Validation

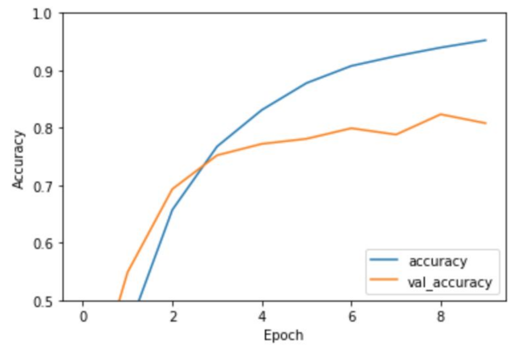
Loss
Training (smoothed)
Training
Validation

SVM linear: Average Accuracy is 68.01%
SVM quadratic: Average Accuracy is 74.60%
SVM gaussian: Average Accuracy is 72.57%
Training and evaluate by the same set of data

Refactor from MATLAB code to Tensorflow in python
Example:

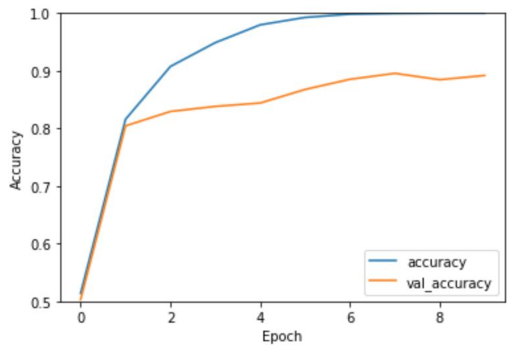
```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_12 (Conv2D)           (None, 26, 26, 32)        608
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 11, 11, 64)        18496
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
conv2d_14 (Conv2D)           (None, 3, 3, 64)          36928
_________________________________________________________________
flatten_6 (Flatten)          (None, 576)               0
_________________________________________________________________
dense_12 (Dense)             (None, 64)                36928
_________________________________________________________________
dense_13 (Dense)             (None, 34)                2210
=================================================================
Total params: 95,170
Trainable params: 95,170
Non-trainable params: 0
_________________________________________________________________
```

```
test accuracy: 0.80808824
1360/1 - 0s - loss: 0.7672 - accuracy: 0.8081
```



```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_16 (Conv2D)           (None, 26, 26, 32)        608
_________________________________________________________________
batch_normalization (BatchNo (None, 26, 26, 32)        128
_________________________________________________________________
max_pooling2d_10 (MaxPooling (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_17 (Conv2D)           (None, 11, 11, 64)        18496
_________________________________________________________________
batch_normalization_1 (Batch (None, 11, 11, 64)        256
_________________________________________________________________
max_pooling2d_11 (MaxPooling (None, 5, 5, 64)          0
_________________________________________________________________
conv2d_18 (Conv2D)           (None, 3, 3, 64)          36928
_________________________________________________________________
batch_normalization_2 (Batch (None, 3, 3, 64)          256
_________________________________________________________________
flatten_7 (Flatten)          (None, 576)               0
_________________________________________________________________
dense_14 (Dense)             (None, 64)                36928
_________________________________________________________________
dense_15 (Dense)             (None, 34)                2210
=================================================================
Total params: 95,810
Trainable params: 95,490
Non-trainable params: 320
_________________________________________________________________
```

```
1360/1 - 1s - loss: 0.6705 - accuracy: 0.8919
test accuracy: 0.89191175
```


Accuracy
Epoch
accuracy
val_accuracy