deep learning hw #1

Problem1

Network structure

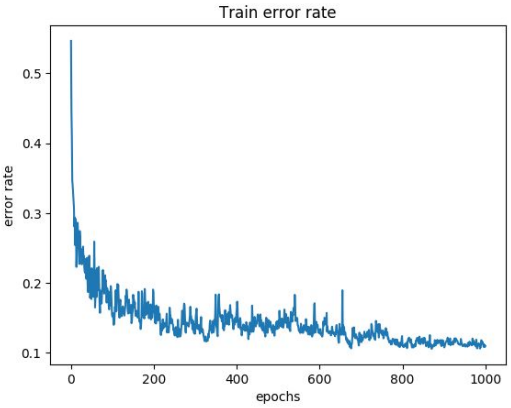
- input (28x28)
- fully connect (60)
- sigmoid (60)
- fully connect (2)
- fully connect (10)
- softmax output (10)

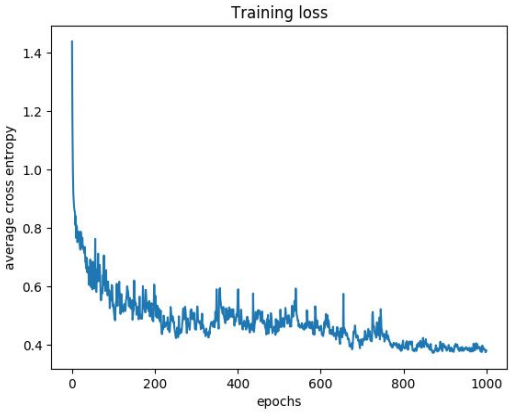
Hyperparameters

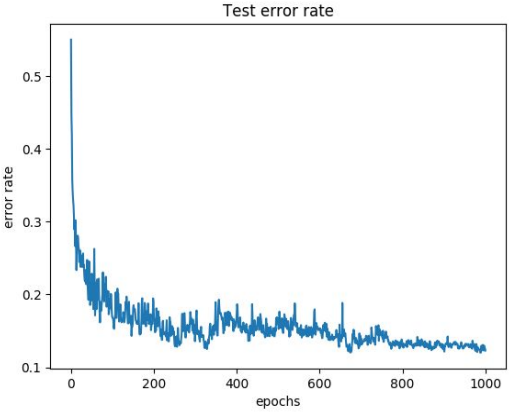
- learning rate (0.4)
- batch size (16)
- epochs (1000)

Initialize

- random init weights

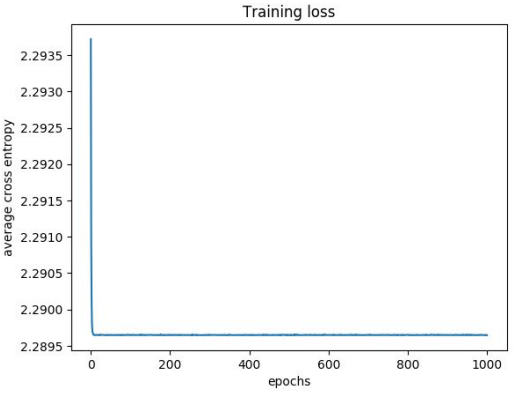


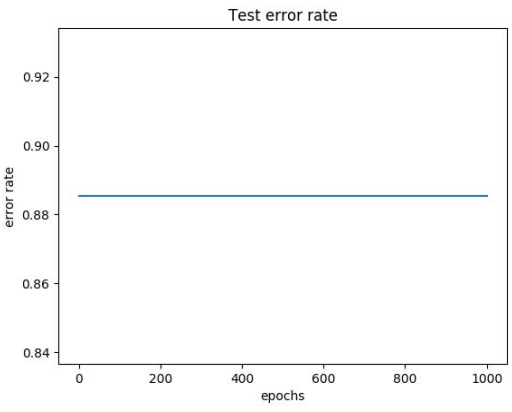


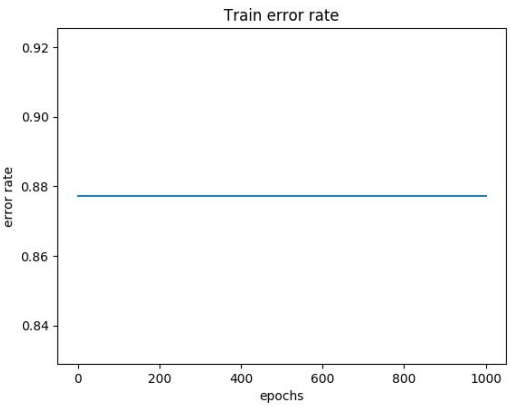


- zero init weights

fail to calculate the gradient using zero init weights

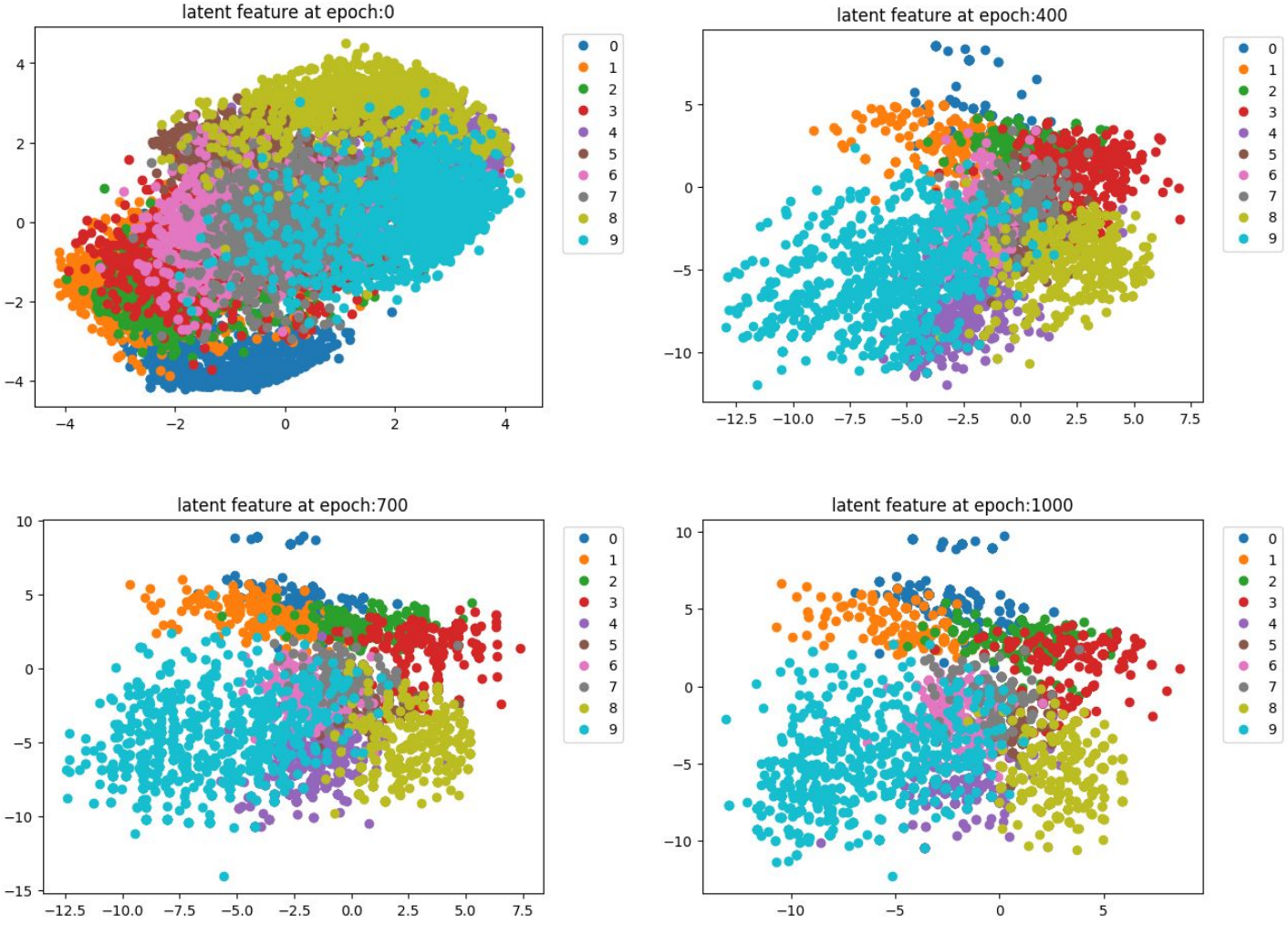






Latent space visualization

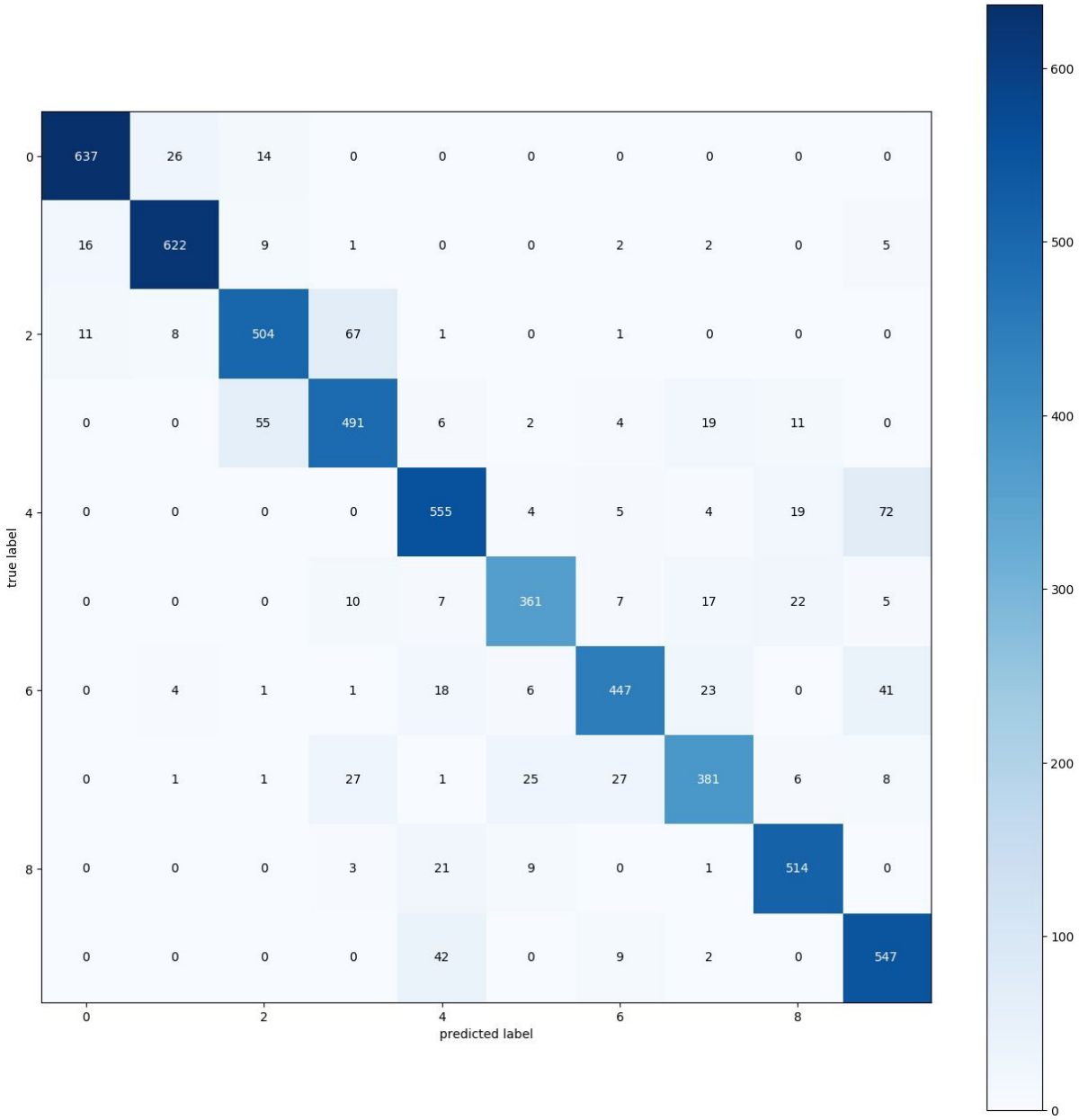
As training goes on. Different class of input gradually groups together.



Confusion matrix

most class are correctly classified.
number 2 have a few cases being recognized as 3
number 3 have a few cases being recognized as 2

number 4 have a few cases being recognized as 9
number 9 have a few cases being recognized as 4



Problem2

Image preprocess

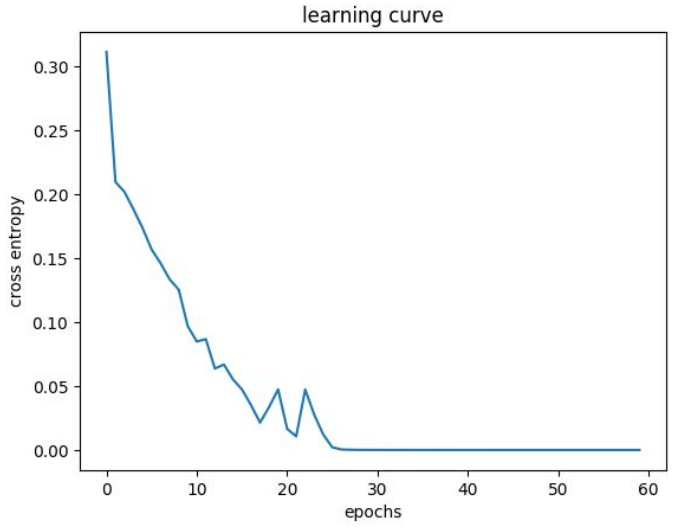
crop every face in bounding box, and resize to 80x80x3 to have consistent input, and keep the detail of every image.

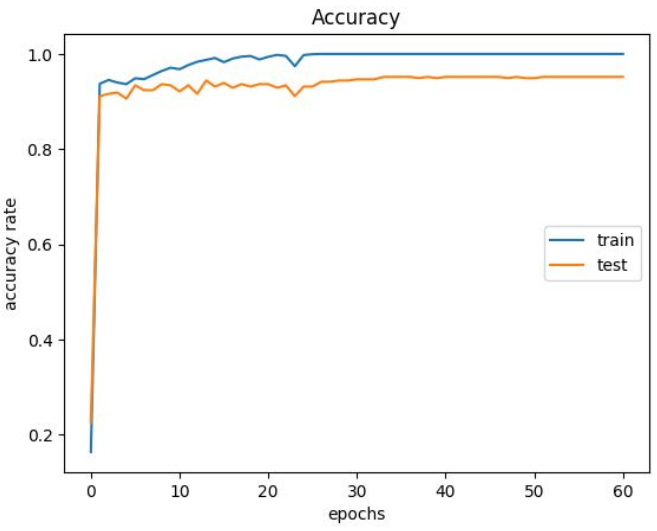
Network structure & Hyper parameter

- learning rate (0.001)
- batch size(8)
- epoch (60)

setting 1

- input(80x80)
- conv2D(input channel: 3, output channel: 6, kernel size: 5, stride: 1)
- maxpool2D(stride: 2)
- conv2D(input channel: 6, output channel: 16, kernel size: 5, stride: 1)
- fully connected(120)
- relu(120)
- fully connected(84)
- relu(84)
- fully connected output (3)

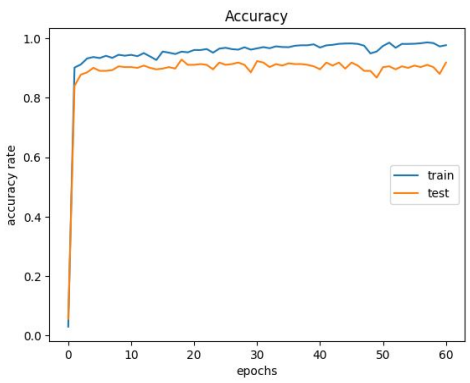


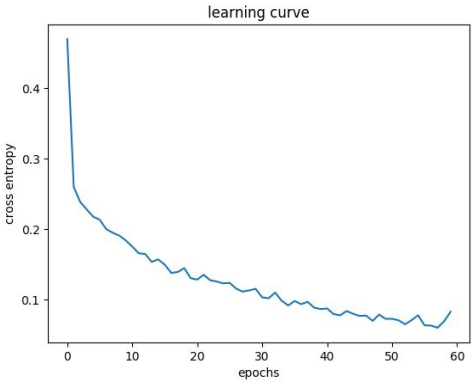


setting 2

change convolution stride size to 3, need longer training time to converge.

- input(80x80)
- conv2D(input channel: 3, output channel: 6, kernel size: 5, stride: 3)
- maxpool2D(stride: 2)
- conv2D(input channel: 6, output channel: 16, kernel size: 5, stride: 3)
- fully connected(120)
- relu(120)
- fully connected(84)
- relu(84)
- fully connected output (3)

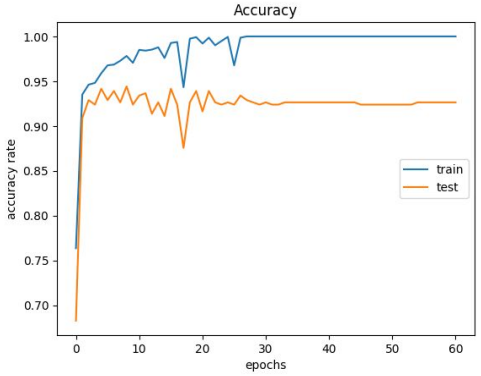


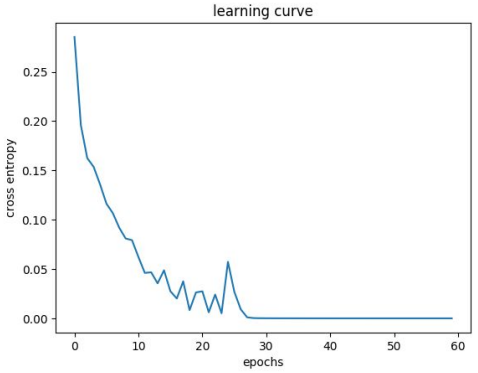


setting 3

change kernel size to 3, result is similar to setting 1.

- input(80x80)
- conv2D(input channel: 3, output channel: 6, kernel size: 3, stride: 1)
- maxpool2D(stride: 2)
- conv2D(input channel: 6, output channel: 16, kernel size: 3, stride: 1)
- fully connected(120)
- relu(120)
- fully connected(84)
- relu(84)
- fully connected output (3)

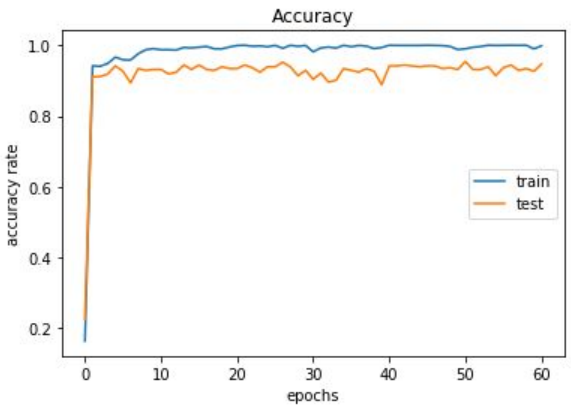


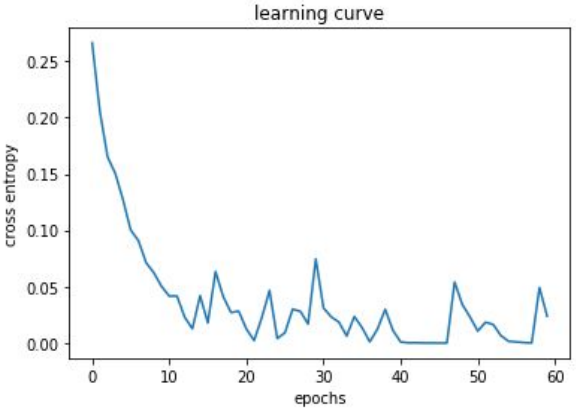


setting 4

add batch nomalization, result is similar to setting 1. But no overfit.

- input(80x80)
- BatchNormalization
- conv2D(input channel: 3, output channel: 6, kernel size: 5, stride: 1)
- maxpool2D(stride: 2)
- BatchNormalization
- conv2D(input channel: 6, output channel: 16, kernel size: 5, stride: 1)
- fully connected(120)
- relu(120)
- fully connected(84)
- relu(84)
- fully connected output (3)

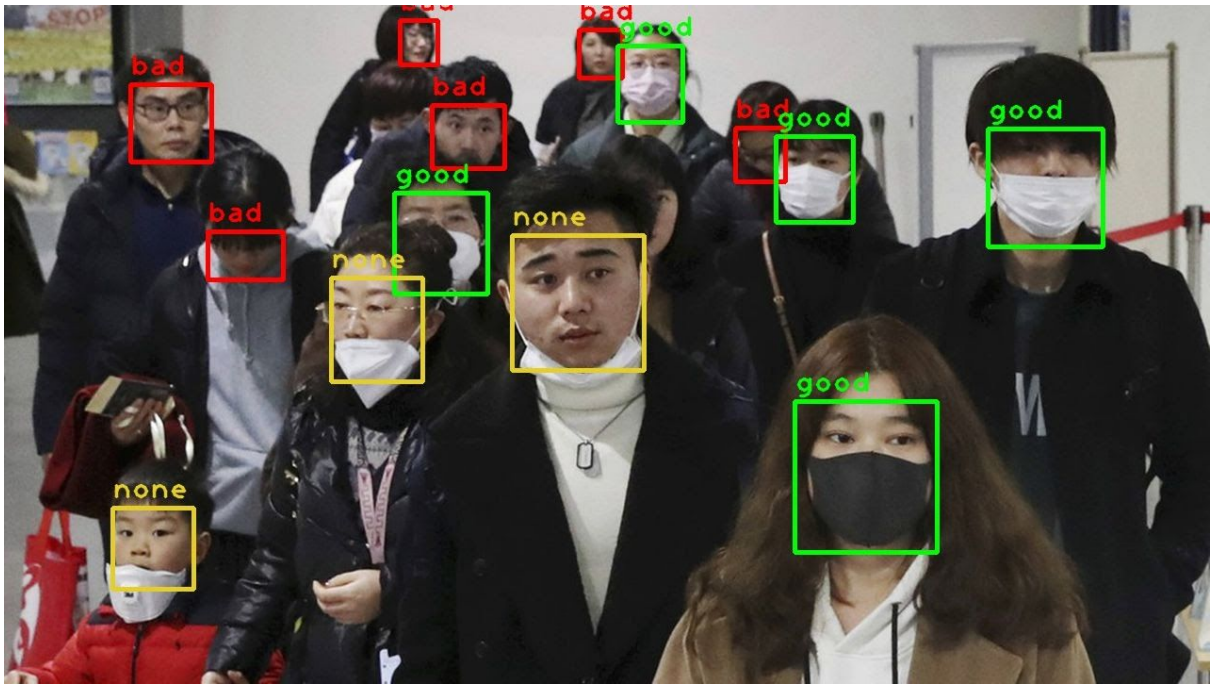




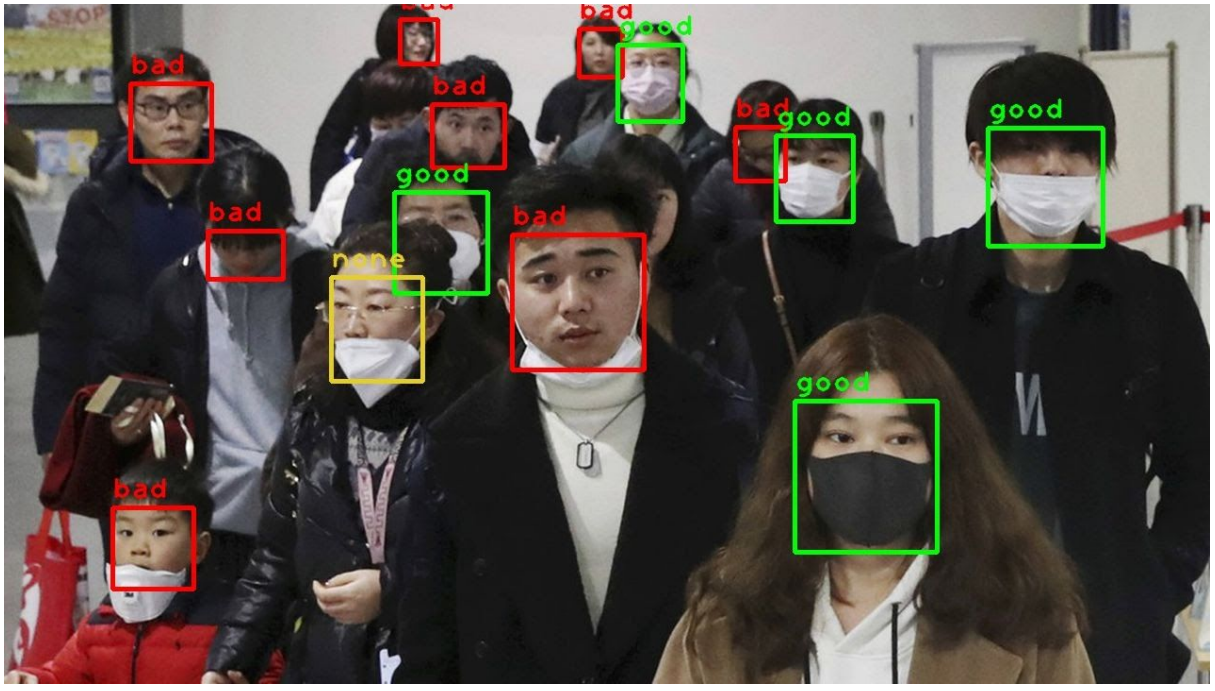
Visualized example

visualize example of test dataset using model trained by setting 1

- label



- model predict



- label

- model predict

- label

- model predict

Accuracy

accuracy of model trained by setting 1

class	Train accuracy	Test accuracy
Good	1	0.9788
None	1	0.5
Bad	1	0.9775

None class has lowest accuracy because model usually recognized none as good or bad. Sometimes none class looks like there is a mask or looks like the top of the shirts.