CSC420 Assignment 5
1002077726
Guanxiong Liu
liuguanx
1a. Yes, it is a good to train binary classification neural networks since I only have five images per employee. The available data is relatively small, so it would be appropriate to have a binary classifier for each employee. i.e. One vs. All classifier.

1b. TF-IDF is useful because common objects (visual words) in the whole dataset should weigh less when match an image in a query since if every picture has this visual word then this word is not salient enough to be a candidate of matching the two pictures. Also, TF-IDF weights words that occur more often in a document. It will make the document more salient/unique or emphasize its feature.

1c. Please see code in facenet.py

1d. The embeddings must be clustered to calculate mean (separate different faces), thus we can use mean to build inverted index for fast image retrieval.

1e. Choose cluster's center (mean) vector as visual word representation.

1f. Please see code in facenet.py. Here is the dictionary.

{1: ['face_image_0.jpg', 'face_image_100.jpg', 'face_image_101.jpg', 'face_image_105.jpg', 'face_image_106.jpg', 'face_image_108.jpg', 'face_image_119.jpg', 'face_image_122.jpg', 'face_image_135.jpg', 'face_image_139.jpg', 'face_image_156.jpg', 'face_image_161.jpg', 'face_image_165.jpg', 'face_image_180.jpg', 'face_image_186.jpg', 'face_image_192.jpg', 'face_image_194.jpg', 'face_image_197.jpg', 'face_image_198.jpg', 'face_image_199.jpg', 'face_image_32.jpg', 'face_image_33.jpg', 'face_image_45.jpg', 'face_image_52.jpg', 'face_image_54.jpg', 'face_image_57.jpg', 'face_image_62.jpg', 'face_image_77.jpg', 'face_image_80.jpg', 'face_image_83.jpg', 'face_image_84.jpg', 'face_image_89.jpg', 'face_image_98.jpg'],

4: ['face_image_1.jpg', 'face_image_11.jpg', 'face_image_116.jpg', 'face_image_128.jpg', 'face_image_14.jpg', 'face_image_148.jpg', 'face_image_154.jpg', 'face_image_16.jpg', 'face_image_162.jpg', 'face_image_172.jpg', 'face_image_177.jpg', 'face_image_184.jpg', 'face_image_196.jpg', 'face_image_2.jpg', 'face_image_23.jpg', 'face_image_25.jpg', 'face_image_29.jpg', 'face_image_36.jpg', 'face_image_46.jpg', 'face_image_48.jpg', 'face_image_5.jpg', 'face_image_55.jpg', 'face_image_56.jpg', 'face_image_61.jpg', 'face_image_66.jpg', 'face_image_79.jpg', 'face_image_88.jpg'],

5: ['face_image_10.jpg', 'face_image_102.jpg', 'face_image_103.jpg', 'face_image_107.jpg', 'face_image_111.jpg', 'face_image_113.jpg', 'face_image_12.jpg', 'face_image_13.jpg', 'face_image_132.jpg', 'face_image_136.jpg', 'face_image_142.jpg', 'face_image_170.jpg', 'face_image_178.jpg', 'face_image_183.jpg', 'face_image_191.jpg', 'face_image_26.jpg', 'face_image_39.jpg', 'face_image_42.jpg', 'face_image_43.jpg', 'face_image_53.jpg', 'face_image_59.jpg', 'face_image_63.jpg', 'face_image_64.jpg', 'face_image_65.jpg', 'face_image_68.jpg', 'face_image_72.jpg', 'face_image_73.jpg', 'face_image_75.jpg', 'face_image_85.jpg', 'face_image_97.jpg'],

0: ['face_image_104.jpg', 'face_image_114.jpg', 'face_image_120.jpg', 'face_image_123.jpg', 'face_image_150.jpg', 'face_image_152.jpg', 'face_image_164.jpg', 'face_image_166.jpg',

CSC420 Assignment 5
1002077726
Guanxiong Liu
liuguanx

'face_image_168.jpg', 'face_image_195.jpg', 'face_image_21.jpg', 'face_image_3.jpg', 'face_image_35.jpg', 'face_image_49.jpg'],

2: ['face_image_109.jpg', 'face_image_112.jpg', 'face_image_115.jpg', 'face_image_117.jpg', 'face_image_124.jpg', 'face_image_125.jpg', 'face_image_130.jpg', 'face_image_131.jpg', 'face_image_133.jpg', 'face_image_138.jpg', 'face_image_140.jpg', 'face_image_143.jpg', 'face_image_144.jpg', 'face_image_145.jpg', 'face_image_146.jpg', 'face_image_149.jpg', 'face_image_15.jpg', 'face_image_157.jpg', 'face_image_158.jpg', 'face_image_160.jpg', 'face_image_169.jpg', 'face_image_171.jpg', 'face_image_174.jpg', 'face_image_175.jpg', 'face_image_176.jpg', 'face_image_179.jpg', 'face_image_182.jpg', 'face_image_187.jpg', 'face_image_188.jpg', 'face_image_189.jpg', 'face_image_19.jpg', 'face_image_190.jpg', 'face_image_193.jpg', 'face_image_22.jpg', 'face_image_24.jpg', 'face_image_30.jpg', 'face_image_31.jpg', 'face_image_38.jpg', 'face_image_4.jpg', 'face_image_41.jpg', 'face_image_47.jpg', 'face_image_50.jpg', 'face_image_51.jpg', 'face_image_58.jpg', 'face_image_6.jpg', 'face_image_69.jpg', 'face_image_7.jpg', 'face_image_70.jpg', 'face_image_71.jpg', 'face_image_78.jpg', 'face_image_81.jpg', 'face_image_87.jpg', 'face_image_95.jpg', 'face_image_96.jpg'],

3: ['face_image_110.jpg', 'face_image_118.jpg', 'face_image_121.jpg', 'face_image_126.jpg', 'face_image_127.jpg', 'face_image_129.jpg', 'face_image_134.jpg', 'face_image_137.jpg', 'face_image_141.jpg', 'face_image_147.jpg', 'face_image_151.jpg', 'face_image_153.jpg', 'face_image_155.jpg', 'face_image_159.jpg', 'face_image_163.jpg', 'face_image_167.jpg', 'face_image_17.jpg', 'face_image_173.jpg', 'face_image_18.jpg', 'face_image_181.jpg', 'face_image_185.jpg', 'face_image_20.jpg', 'face_image_27.jpg', 'face_image_28.jpg', 'face_image_34.jpg', 'face_image_37.jpg', 'face_image_40.jpg', 'face_image_44.jpg', 'face_image_60.jpg', 'face_image_67.jpg', 'face_image_74.jpg', 'face_image_76.jpg', 'face_image_8.jpg', 'face_image_82.jpg', 'face_image_86.jpg', 'face_image_9.jpg', 'face_image_90.jpg', 'face_image_91.jpg', 'face_image_92.jpg', 'face_image_93.jpg', 'face_image_94.jpg', 'face_image_99.jpg']}

1g. First encode each image to embedding, then feed it into kmeans.predict(). When I got the classification (0~5) then I look up visual word representation in kmeans.cluster_centers_[classification]. Then I calculate the Euclidean distance between the embedding and the visual word, as long as the distance is less than 0.5, I consider they are matched. So, I look up in the inverted index dictionary[classification] and get all images that are belong to this class.

1h. Please see the code and output file matching.csv.

1i. In this case, each image should have more than one embedding because there are more than one faces. Then we consider multiple visual words appear in the same image. The challenge is that we do not have a visual word representation in this case. When we have only one face in an image, we can simply do clustering to separate different faces, but in this case, we are not able to do clustering to choose a visual word representation, thus is hard to find all faces that are close to the representation. So, my solution would be first, detect the faces in the images and feed them into the algorithm I just implemented(clustering) and then we will have a dictionary of each center (visual representation) with multiple embeddings belong to it. Then if we want to find a face in all images, we can find the cluster first and use double loop to iterate all embeddings in that cluster to match all possible images. The

CSC420 Assignment 5
1002077726
Guanxiong Liu
liuguanx
complexity will be O(number of embeddings in the cluster*number of images). The challenge is to detect face effectively and correctly without losing any useful information.

2a. When we do back propagation (using gradient based optimization techniques) and calculating the gradients of loss with respect to the weights, the gradients is getting smaller and smaller, so the hidden layers that close to input learn from loss very slow. When you have a large neural network, the gradients in the earlier layers tend to be vanished. We can use ReLU activation function to mitigate the problem because the derivative for ReLU is 1 for all positive value and 0 for all negative value.

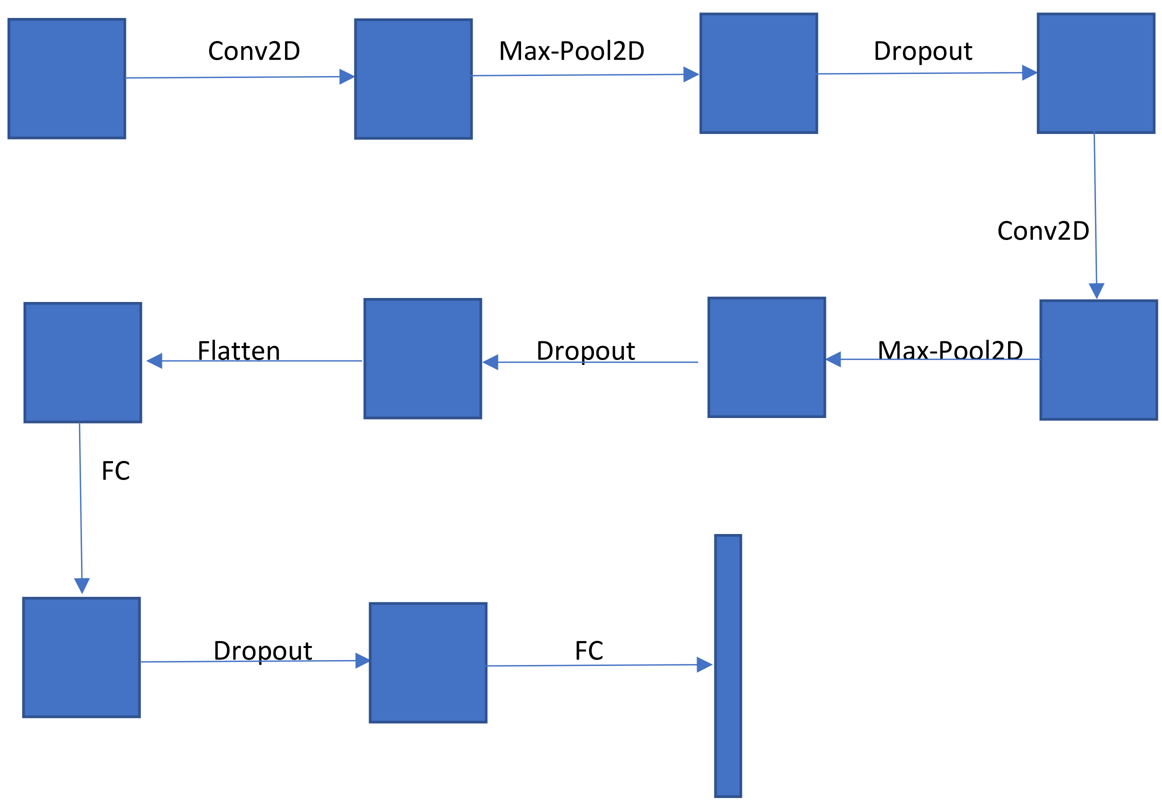
2b. Because the exact location of a feature is less important than its rough location relative to other features in an image, so the max pool exists to reduce the spatial size of the representation and reduce the number of parameters and computational cost.

2c. Please see code in fashion.py.

2d. I chose ReLU for the layers in the middle and SoftMax in the final layer because ReLU mitigates gradient vanishing problem and SoftMax assigns decimal probabilities to each class in a multi-class problem. The probabilities add up to 1. This constraint helps training converge more quickly.

2e. $L(y, \hat{y}) = -\sum_i y_i log\hat{y}_i$ where y is the ground truth and $\hat{y}$ is the prediction.

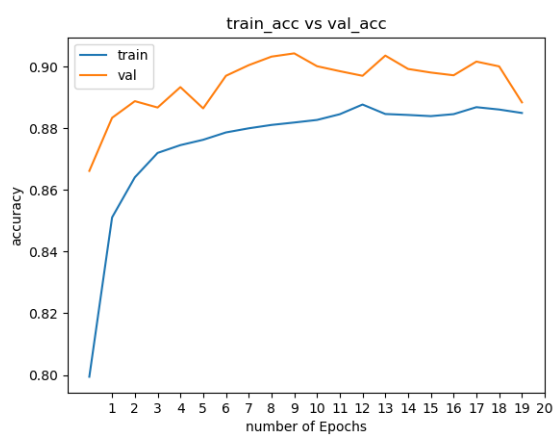
2f.



CSC420 Assignment 5
1002077726
Guanxiong Liu
liuguanx
Model summary:

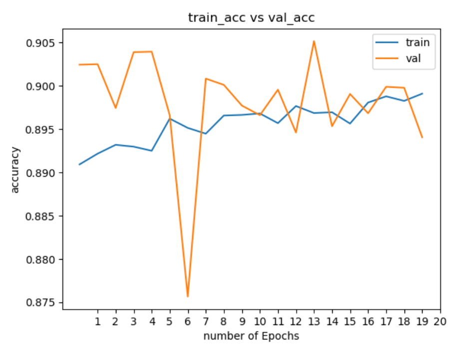
| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d (Conv2D) | (None, 28, 28, 64) | 320 |
| max_pooling2d (MaxPooling2D) | (None, 14, 14, 64) | 0 |
| dropout (Dropout) | (None, 14, 14, 64) | 0 |
| conv2d_1 (Conv2D) | (None, 14, 14, 32) | 8224 |
| max_pooling2d_1 (MaxPooling2D) | (None, 7, 7, 32) | 0 |
| dropout_1 (Dropout) | (None, 7, 7, 32) | 0 |
| flatten (Flatten) | (None, 1568) | 0 |
| dense (Dense) | (None, 256) | 401664 |
| dropout_2 (Dropout) | (None, 256) | 0 |
| dense_1 (Dense) | (None, 10) | 2570 |

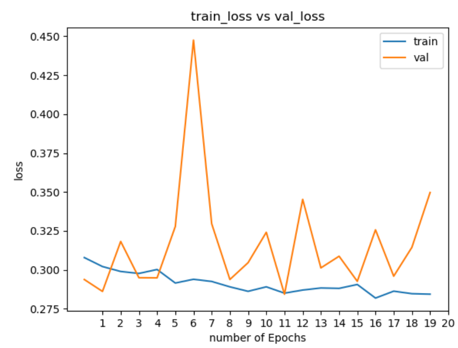
Total params: 412,778
Trainable params: 412,778
Non-trainable params: 0
Learning rate=0.001, batch size=5, number of epochs=20, we should stop training when training accuracy increases, and validation accuracy steadily decreases because it is overfitting. I do not implement early stopping because it is very hard to tell if validation accuracy is at global maximum or it is fluctuating. Final training/validation loss are: 0.3215/0.3702, final training/validation accuracy are: 0.8850/0.8884.Test accuracy: 0.8989

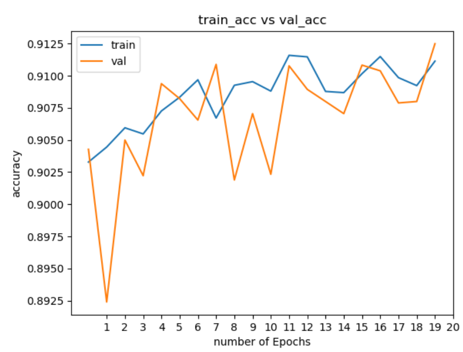


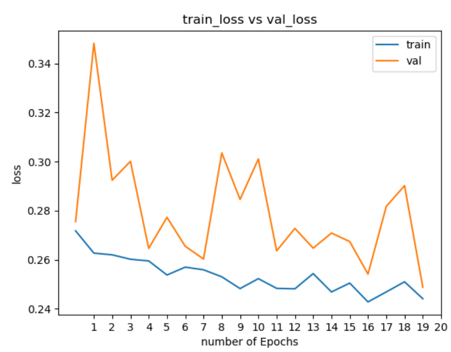

CSC420 Assignment 5
1002077726
Guanxiong Liu
liuguanx

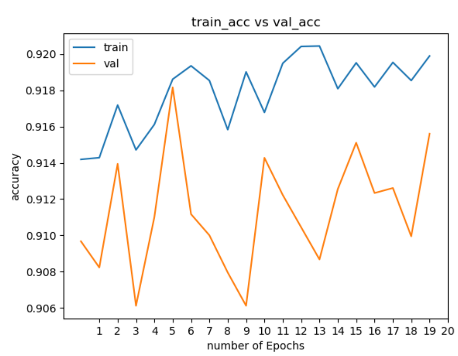
2g. I found that the larger the batch size the higher the test accuracy. This makes sense because when we have larger batch size, we can get more correct gradient. So, when we look at the contour map, it moves to the minima more steadily. If we have relatively small batch size, it will fluctuate more on the contour map thus less accurate. As the batch size gets larger, the fluctuation of the validation is less. Plots are below.

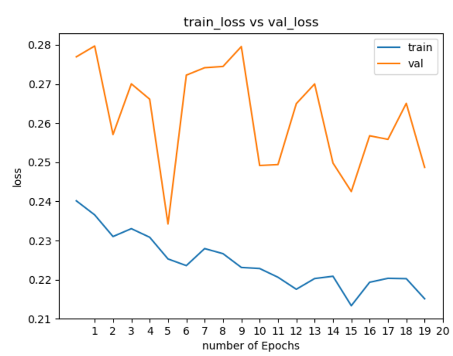
<table>
<tr><th>Batch size</th><th>plot</th></tr>
<tr><td>8</td><td>




Test accuracy: 0.8948</td></tr>
<tr><td>16</td><td>


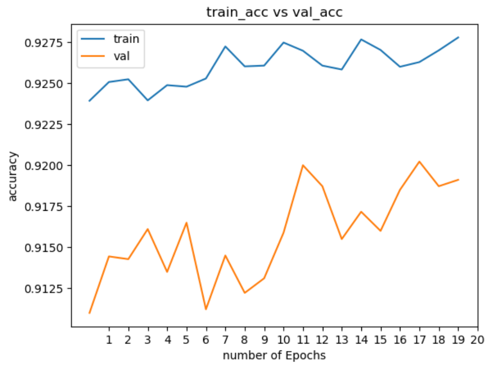


Test accuracy: 0.9063</td></tr>
<tr><td>32</td><td>




Test accuracy: 0.91</td></tr>
</table>

CSC420 Assignment 5
1002077726
Guanxiong Liu
liuguanx

<table>
<tr>
<td>64</td>
<td>


Test accuracy: 0.9181
</td>
</tr>
</table>