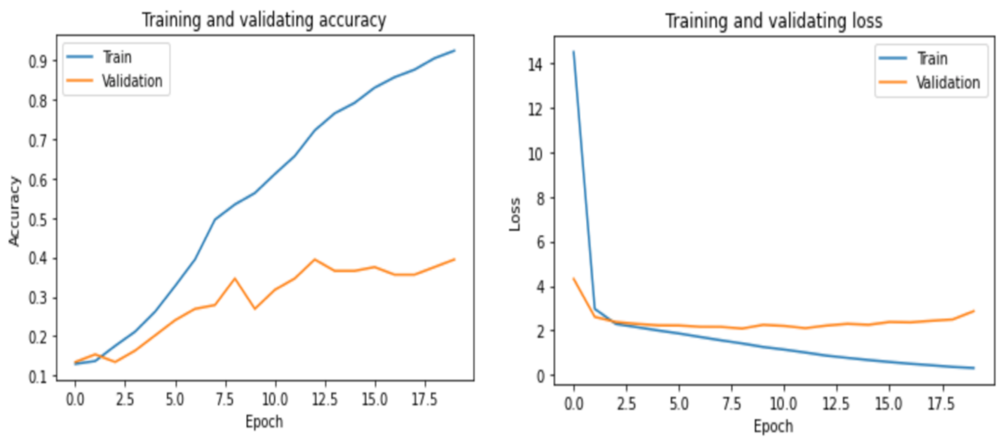
COMP90086 Assignment 2
QINGTAN SHEN
1130945

1.1

Here is the accuracy and loss graph I get from the original CNN structure:



From the train and validation accuracy plot, we can see that after 17 epochs, the train accuracy has arrived nearly 0.9, while the validation accuracy has already stopped increasing, and it's just around 0.4.

From the training and validating loss plot, we can see that after 17 epochs, the train loss has already become zero, while the validation loss is around 2, and it increases smoothly after 15 epochs.

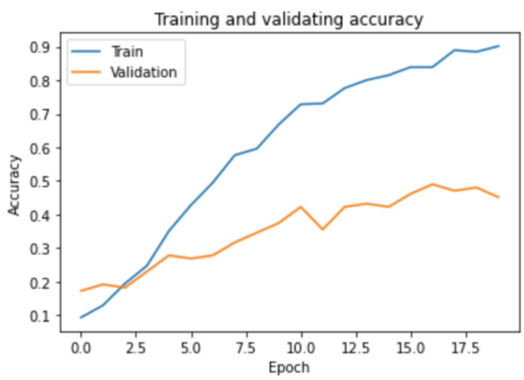
So, there is a overfitting problem we can get from these two plots, and data augmentation will also be used to make the accuracy higher.

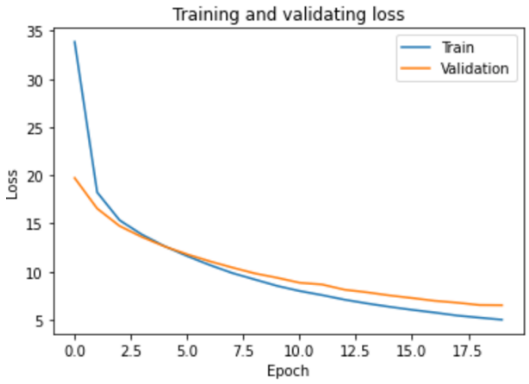
1.2 (1)

In this part, we consider regularization firstly. We consider two methods in this part, which are L1_regularizasion and drop out method. The reason why I didn't choose early stop method is that the largest accuracy is only 0.4, and the validation accuracy decreases not obvious at the last part of training, also we will use data augmentation later, so we will increase the number of epochs, so early stop method is not suitable.

Then I choose L1 regularization experiment instead of L2, because our image input size is small, so some of the parameters in the deeper CNN structure can be zero. Also drop out method is another good way to avoid overfitting.

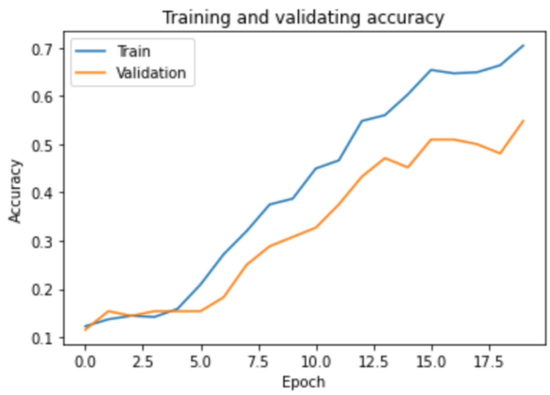
Here is the L1_regularization experiment. Firstly, I choose to add an L1_regularization in the fourth convolutional layer, because this layer has many parameters. And six values are chosen as the L1_regularization value from 0 to 0.3. Here is the plot with best performance in this experiment (L1=0.05).

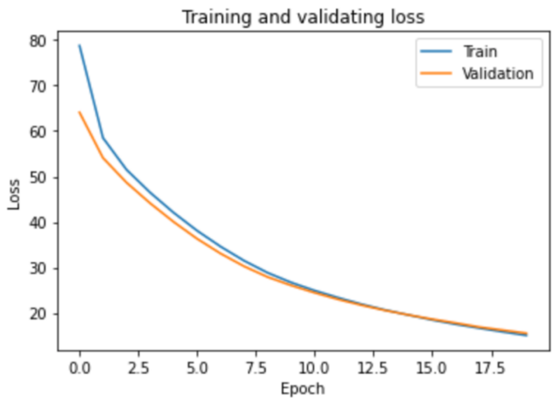




We can see the accuracy is larger, but not obvious, while the validation loss decreases all the time, which is better than the original one.

So, we keep the L1_value in the fourth convolutional layer and also add a L1_regularization to the third convolutional layer, which also has many parameters. Six values were chosen from 0 to 0.3. When L1=0.2, the performance is best, here is the plot:





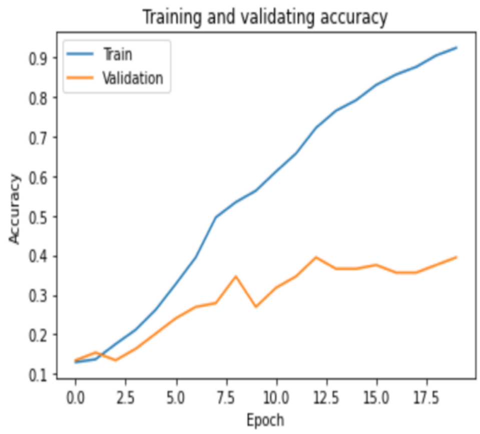
The validation accuracy is around 0.5 after 15 epochs, and the train accuracy is around 0.7, and the validation loss decreases all the time, it has the similar value as train loss after 15 epochs. All the evidence means there is no overfitting problems after adding these two L1 regularization values.

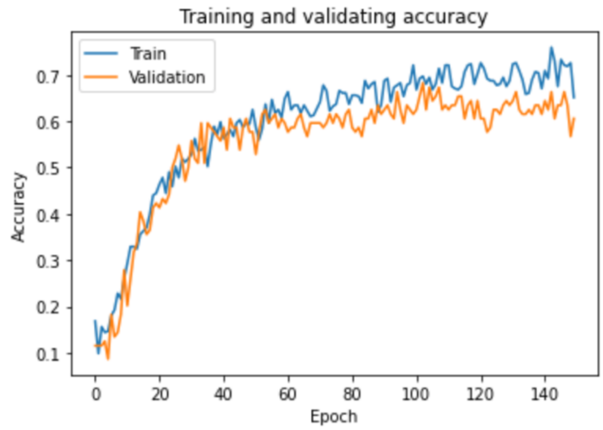
I also try drop out method as an experiment. I try to drop some percentage of relationship between the first and second convolutional layer, and between the third and fourth convolutional layer, but the results are worse than the above (I remain the code in my notebook).

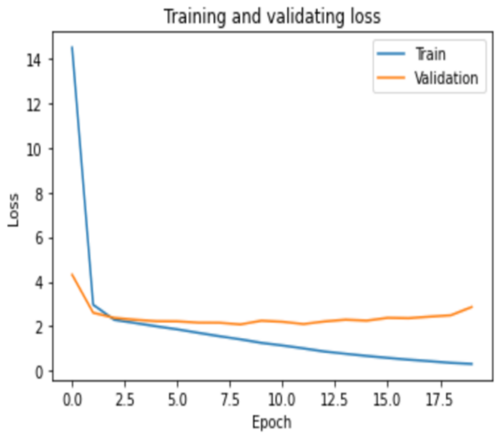
1.2 (2)

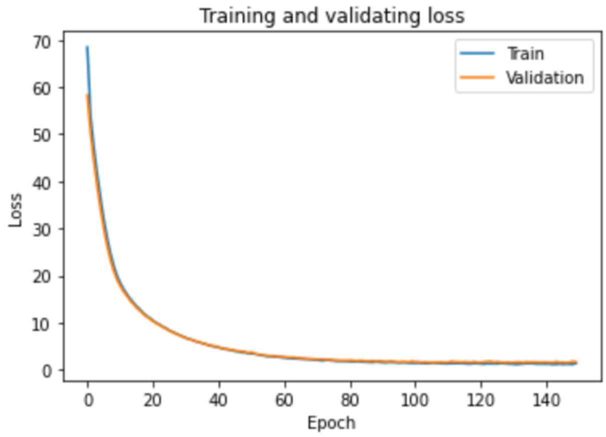
In this part, we consider data augmentation problem, I choose two operations by adding two layers in our CNN network. The first layer I added is flipped horizontal randomly, because I find for the same action, the action directions not always towards the same. The second layer I added is zoom randomly because I find some people in the image is much larger than other images, the parameter I set is 0.4, I found this value is better than others after experiment.

And here is the result comparison for the original CNN (left two) and my CNN after data regularization and data augmentation (right two).





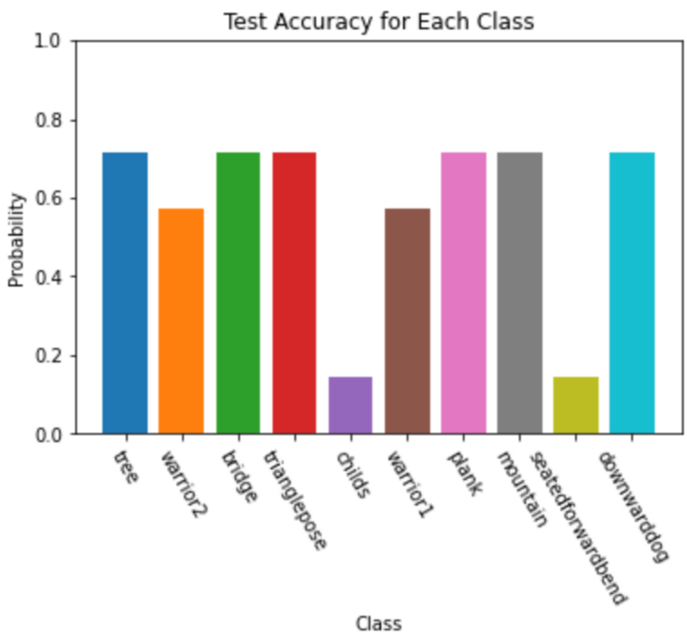




We can see the validation accuracy for the new CNN is around 0.6, which is greater than 0.4 in the original CNN. Also the training and validating loss is almost the same in the new CNN, which means there is no overfitting problem and the model is good.

2.

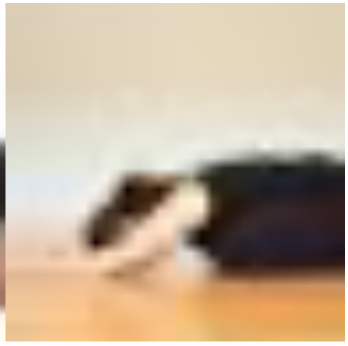
The whole accuracy for our test dataset is 0.5714, and here is the bar chart for average accuracy in each class:



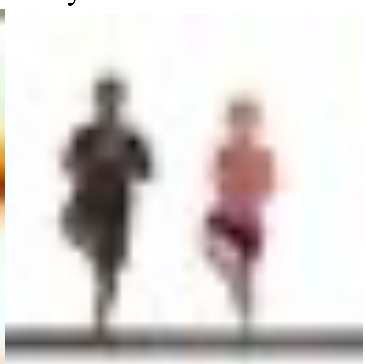
From this chart, we can see that when testing tree, bridge, trianglepose, plank, mountain, dowmwarddog images, the average accuracy is more than 0.7, which is good. When testing warriors2 and warriors1 classes, the average accuracy is almost 0.6, which is also good. However, we get a bad performance when testing childs and seatedforwardbend classes, the accuracy is smaller than 0.2.

Difficulty for images:

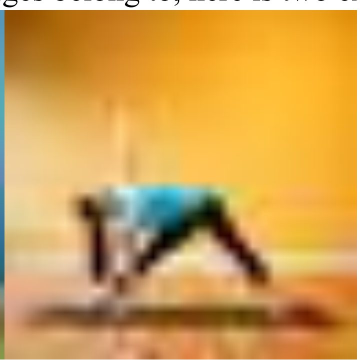
In some test images, the person is large, sometimes we can only see half pose in these images, although it can be zoomed out, we still cannot show the whole pose, such as these two test images:

Also, in some test images, we can get two people with the same pose, or another pose was formed by a mirror, and it is really hard for us to test these images, here are two examples:

Also, in some test images, they have colorful or noisy backgrounds, which is also different to justify what class these images belong to, here is two examples:

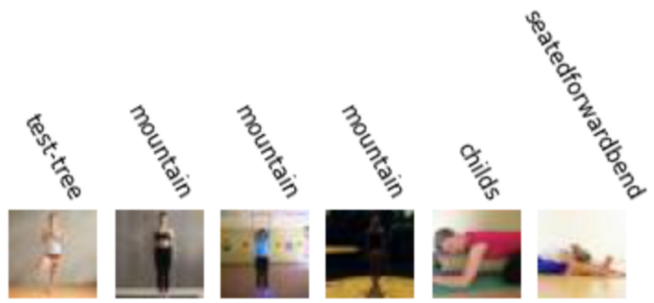



Difficulty for classes:

We know that our model performance is bad when testing 'childs' and 'seatedforwardbend' classes. There are some commons in these two poses, people cannot stretch their bodies to different conditions when doing these poses. Their bodies are not separated obviously with the background. And they all have clothes with different colors. The color area of their clothes is larger than other classes, which may affect the detection. These reasons can lead to a lower performance.

3.

(1) In nearest neighbor analysis, background is an important feature space in my model.

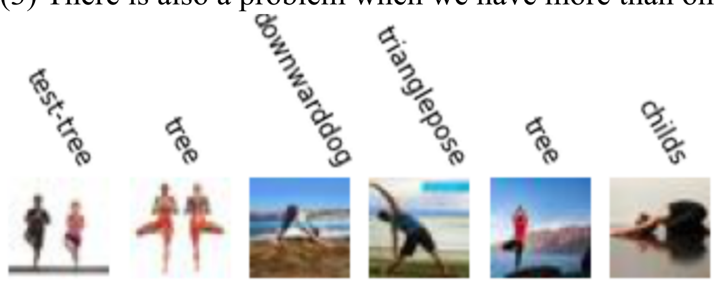


In the plot, the first one in each line is our test sample, the second one is our nearest train sample. We can see in the first line, the nearest one is from mountain class just because they have a similar background. So, background is an important feature space in my model.

(2) The clothes color for a person is also an important feature space.

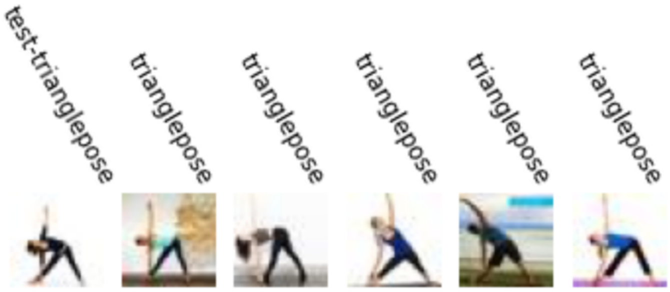
The reason why we get the first and second nearest neighbor belong to class 'mountain' is that they all wear pink clothes. So clothes color is also an important feature space after background.

(3) There is also a problem when we have more than one person in an image.



The test sample has two people, so the nearest neighbor also has two people, but it is hard to recognize other top-nearest images, so it is a restriction in our model.

(4) In some conditions we can get good results.



Like this set of images, we can find the top-five nearest neighbors are all correct, so our model can classify some poses with small noises.

In conclusion, the test accuracy for my model is 0.5714, although my model has a bad performance when testing 'childs' and 'seatedforwardbend' classes, it has a better better performance when testing other classes. And 0.5714 is also a reasonable accuracy value because we have 10 classes. But some of the feature space of my model is related to the background or other unimportant objects, which is a shortage in my model.