COMP90086 Assignment 1
Jiawei Luo
1114028

Question 1:

For output size:

No padding: output size=ceiling((input size - kernel size+1)/stride)

padding: output size=ceiling(input size / stride)

number of channel = number of filters

For No.Para and No.Multiplications

k_c: column of kernel, k_r: row of kernel, K: number of filters

w_o: width of output size, h_o: height of output size, M_o: channel of output

w_i: width of input size, h_i: height of input size, M: channel of input

No.multiplication: M * k_r * k_c * w_o * h_o *K

No.parameters: K * M * k_c * k_r

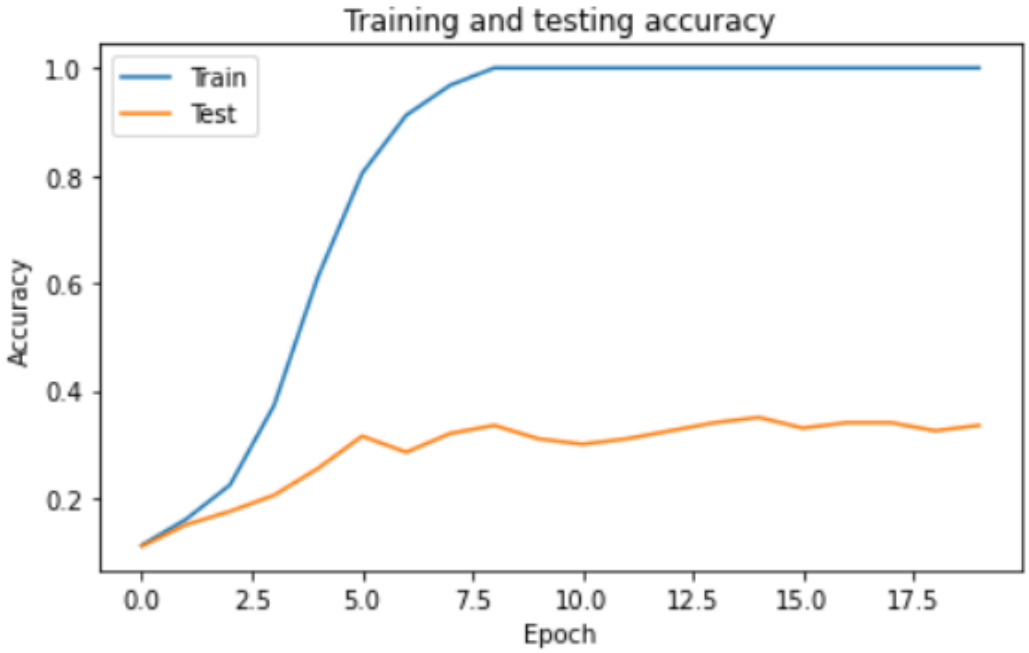
Layer	**Input**	**NO.filter**	**Filter Size**	**Padding**	**Stride**	**Output Shape**	**No. Para**	**No. Multiplications**
First Convolutional Layer	32*32*3	32	5*5	vaild	1*1	28*28*32	32*3*5*5 = 2400	3*5*5*28*28*32 = 1881600
Module 1								
Left Conv 1	28*28*32	32	1*1	same	1*1	28*28*32	32*32*1*1 =1024	32*1*1*28*28*32 =802816
Middle Conv 1	28*28*32	32	1*1	same	1*1	28*28*32	32*32*1*1 =1024	32*1*1*28*28*32 =802816
Middle Conv 2	28*28*32	64	3*3	same	1*1	28*28*64	32*64*3*3 =18432	32*5*5*28*28*64 =40140800
Right max pooling 1	28*28*32		3*3	same	1*1	28*28*32	0	0
Right Conv 2	28*28*32	32	1*1	same	1*1	28*28*32	32*32*1*1 =1024	32*1*1*28*28*32 =802816
Concatenate						28 * 28 * (32 + 32 + 64) = 28 * 28 * 128		
Module 2								
Left Conv 1	28 * 28 * 128	64	1*1	same	1*1	28 * 28 * 64	64*128*1*1 =8192	128*1*1*28*28*64 =6422528
Middle Conv 1	28 * 28 * 128	64	1*1	same	1*1	28 * 28 * 64	64*128*1*1 =8192	128*1*1*28*28*64 =6422528
Middle Conv 2	28 * 28 * 64	128	3*3	same	1*1	28 * 28 * 128	64*128*3*3 =73728	64*3*3*28*28*128 =57802752
Right max pooling 1	28 * 28 * 128		3*3	same	1*1	28 * 28 * 128	0	0
Right Conv 2	28 * 28 *	64	1*1	same	1*1	28 * 28 * 64	64*128*1*1	128*1*1*28*28*64

	128						=8192	=6422528
Concatenate						28 * 28 * (64 + 64 + 128) = 28 * 28 * 256		
Max Pooling	28 * 28 * 256		2*2	same	2*2	14*14*256	0	0
Flatten	14*14*256					14*14*256 =50176	0	0
FC 256	50176					256	50176*256 =12845056	12845056
FC 10	256					10	256*10 =2560	2560

Question 2

2.1

recognition results:

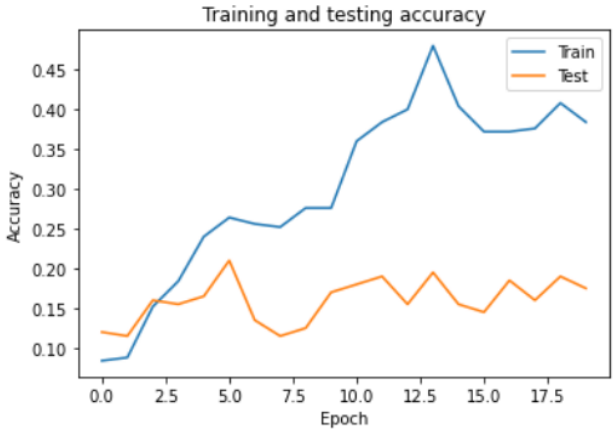


From the results, the model gradually fits the training data, while, the results on the test data set no longer increase at 2-5 epochs, and the training set accuracy gradually goes to 1, the model is quickly **overfitting**.

Too few training instances, our model is able to fit this data very quickly and does not get any boost after fully fitting the data, it also completely fits the noise and irrelevant information in the training data. the model has difficulty recognising new test data and the robustness of the model is very poor. Also, the model fits too quickly, so it needs to be trained slower or regularised to control the complexity of the model.

2.2

recognition results

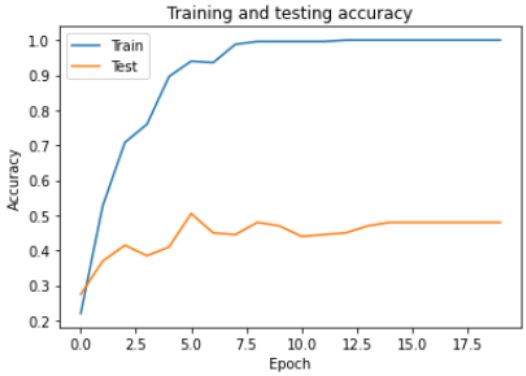


This MLP model simply takes the image flattened and feeds it into the fully connected neural network as inputs. Each node of the fully connected network is connected to each node of the next layer and generates weights, forming a very inefficient network. Image recognition using MLP is not translation invariant, as the input to the image is flattened, the object in the image will form a completely different input if it is translated and rotated, making MLP training and recognition very difficult.

On the other hand, the spatial information of the image after flatten will be lost, resulting in low recognition ability of the model.

2.3

recognition results



One way to extract features from raw images is to use a pre-trained CNN (VGG19) for feature extraction, and to reduce the dimensionality of large data images imformation to small feature image and effectively retain the image features.

The low-dimensional features are fed into the MLP as vectors, VGG can effectively identifies similar images features when they are flipped, rotated or transformed in position. MLP takes the input features flattened, passes them through two fully-connected layers, outputs 10 vectors, and converts them by softmax into the probabilities of ten labels and selects the one with the highest probability as the classification result.

The input layer of the MLP becomes an ordered feature vector input, and the MLP is able to identify similar input features on the basis of the feature vector, while reducing the effects of image translation, flipping and rotation.