Student: Chandan Chandel

Student Number: 250914472

Course: Advanced Image Processing

Professor: Dr. Yimin Yang

Assignment 3 – Transfer Learning

Dataset Description: I used the CIFAR 10 dataset which has 10 classes and 60000 images, with 6000 image per class. Each image is 32x32 and is of RGB type as opposed to a black and white image. The CIFAR10 dataset was split into train and test sets along a 5:1 ratio, therefore the training set has 50000 images.

Dataset Preprocessing: The images (x train and test data) were normalized using min-max scaling.

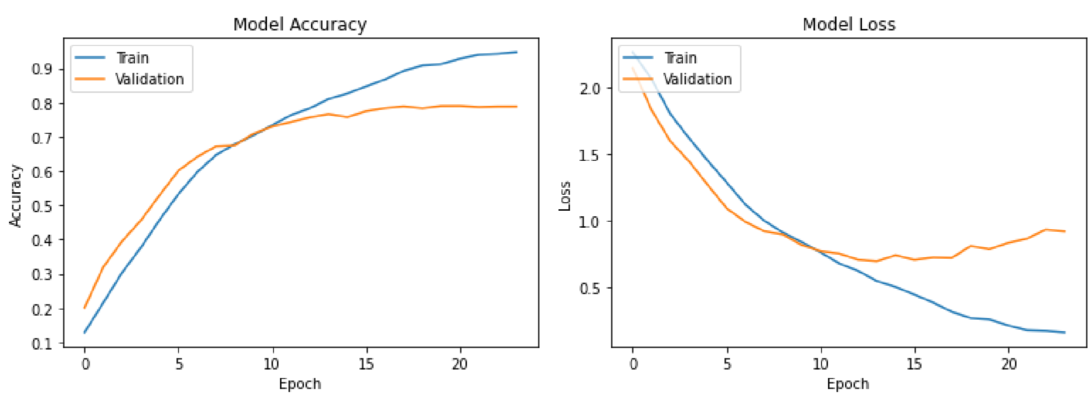
$$x' = (x - \min(x))/(\max(x) - \min(x))$$

The labels were transformed into categorical data using the `keras.utils.to_categorical` command.

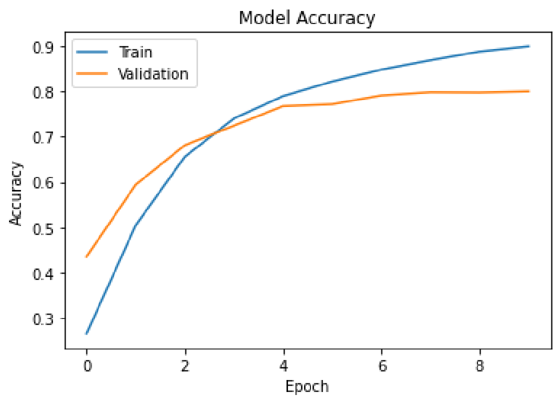
Splitting the dataset: I split the dataset into the test and train datasets using a 5/1 split.

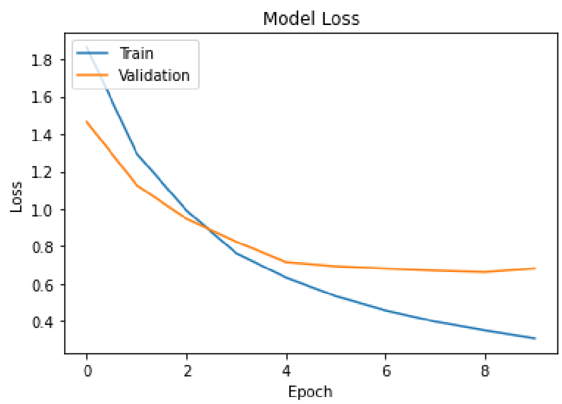
Network Selection: I selected the VGG16 model after consulting CHAT-GPT on which model (out of ResNet50, AlexNet and VGG16) will be best suited to classifying the CIFAR10 dataset. CHAT-GPT ruled out ResNet50 primarily due to it being overkill and said AlexNet and VGG16 may be more suitable due to their shallower architectures and then added that VGG16 in particular had stronger results with small datasets like CIFAR10.

Compare the trained/fine-tuned pretrained DCNN with the DCNN from assignment 2: My DCNN from assignment 2 was only trained on the CIFAR10 dataset and achieved a testing accuracy of 80.1% while the training accuracy was 95.9% - this suggests that there was strong overfitting. I tried to combat this by the following means; increasing the regularisation parameter value in the Dropout Layers, by decreasing the number of free parameters and by implementing early stopping. I also tried to use data augmentation for the images as well but I did not find very helpful results from this process when I used ImageDataGenerator class from tensorflow.keras.preprocessing.image. The training results are shown below.



In Assignment 3 I pretrained my VGG16 model using the ImageNet data source and I retrained on the CIFAR10 dataset achieved a training accuracy of 91.6% while the testing accuracy was 79.9%. The overfitting was less severe and was not accompanied by a reduction in accuracy. The training results are shown below.





Explaining the performance difference:

1. Superior network architecture: VGG16 is a well-know model that is crafted to perform well on the exact datasets which I am working on and used by many professionals, whereas the architecture I designed through trial and error in Assignment 2 is relatively untested and still has poorly understood limitations.
2. Transfer Learning: By pretraining my model I essentially expanded my dataset considerably and allowed the model to pick up rich features that are commonly found in many images. The act of expanding your dataset is typically considered to be a good measure for combatting overfitting, which most likely explains the reduced overfitting in Assignment 3.

Bonus Section:

<u>SIFT features + MLP classification task</u>

In Assignment 1 I achieved relatively good results when I used SIFT for feature extraction and then used an MLP for classification, as shown in the results below. I used images that had been normalized to 300 x 300 and I was classifying 99 classes.

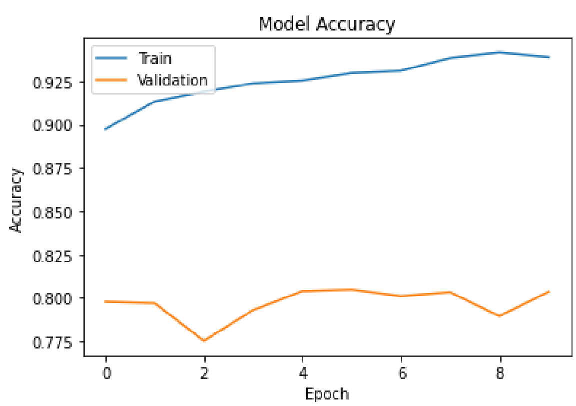
	precision	recall	f1-score	support
accuracy			0.72	297
macro avg	0.69	0.73	0.67	297
weighted avg	0.76	0.72	0.70	297

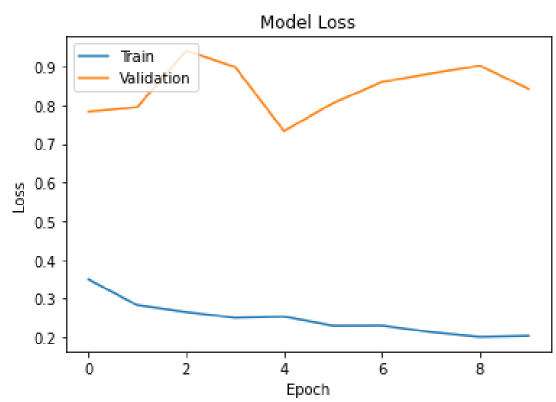
Below is a classification report of the SIFT features + MLP Classification model. The results were much weaker than when I had completed the same task in Assignment 1 and I believe this is because it is my data choice (CIFAR 10) consists of images with shape 32x32. **Images this small are very bad for extracting SIFT features** and that is reflected in the results even though I was only classifying 10% of the classes that I was doing in Assignment 1.

	precision	recall	f1-score	support
0	0.51	0.24	0.32	1000
1	0.50	0.09	0.15	1000
2	0.25	0.02	0.04	1000
3	0.00	0.00	0.00	1000
4	0.14	0.00	0.01	1000
5	0.33	0.02	0.04	1000
6	0.20	0.01	0.02	1000
7	0.40	0.05	0.09	1000
8	0.48	0.12	0.19	1000
9	0.50	0.05	0.09	1000
micro avg	0.45	0.06	0.11	10000
macro avg	0.33	0.06	0.09	10000
weighted avg	0.33	0.06	0.09	10000
samples avg	0.06	0.06	0.06	10000

<u>Deep features + MLP classification task</u>

1. Train VGG16
2. Cut off the FC layers (equivalent to MLP)
3. Replace the old FC layers with a generic MLP
4. Retrain the model





Above are the results of training after cutting off the FC layers and replacing it with new FC layers that needed to be trained again. The validation accuracy was very strong from epoch 1 at 80% accuracy, but this training process didn't yield stronger results upon further training (epoch 10 shows a similar accuracy of 82%). **The validation accuracy was also identical to what was achieved in the original VGG16 training**. This suggests that the convolution layers - which extract features - are incredibly important to model accuracy, meanwhile the FC layers which do the actual classification are not as important to train with much precision.

Comparing SIFT + MLP versus VGG16 convolutional layers + MLP: It wasn't even close. SIFT is useless on the CIFAR 10 dataset and provided unhelpful results. Meanwhile VGG16 results were very impressive and showed that the FC layers are not nearly as hard to train as the convolutional layers in the VGG16.