

# COMPUTER VISION ASSIGNMENT 1 REPORT

Gurmehak kaur

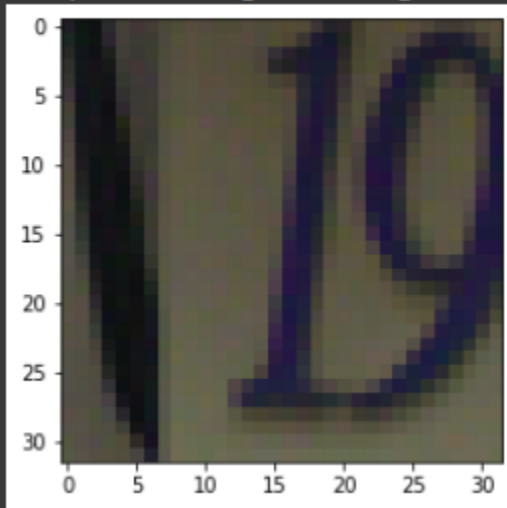
2020298

1.

1. The dataset was loaded, and the corresponding data loader was created.

```
168] plt.imshow(df[:, :, :, 0])
```

<matplotlib.image.AxesImage at 0x7f24dce7f1c0>



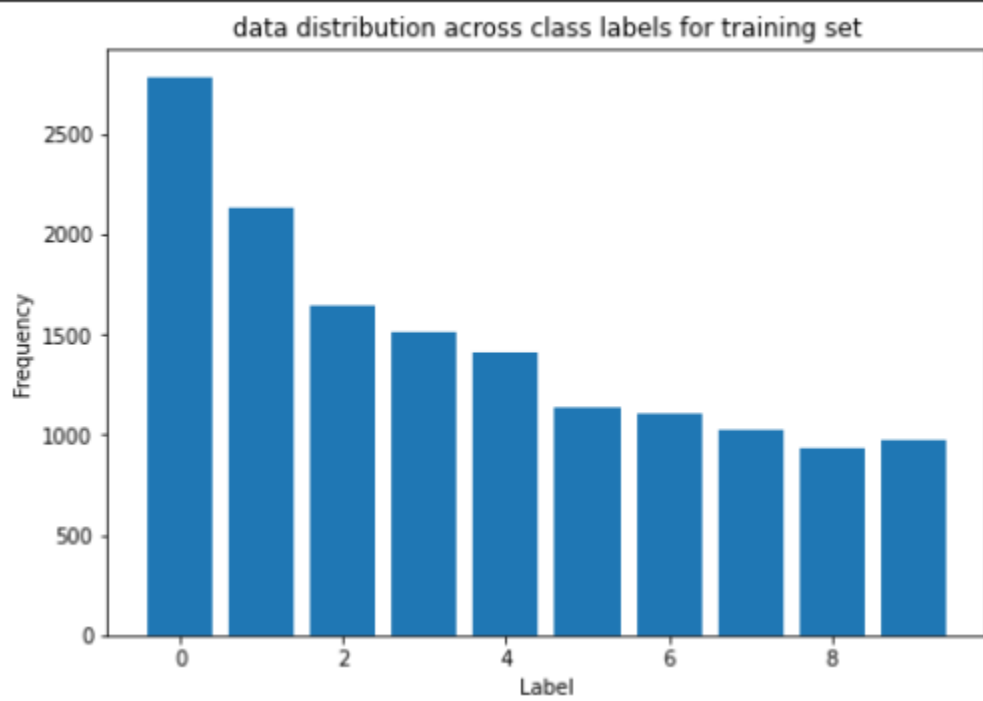
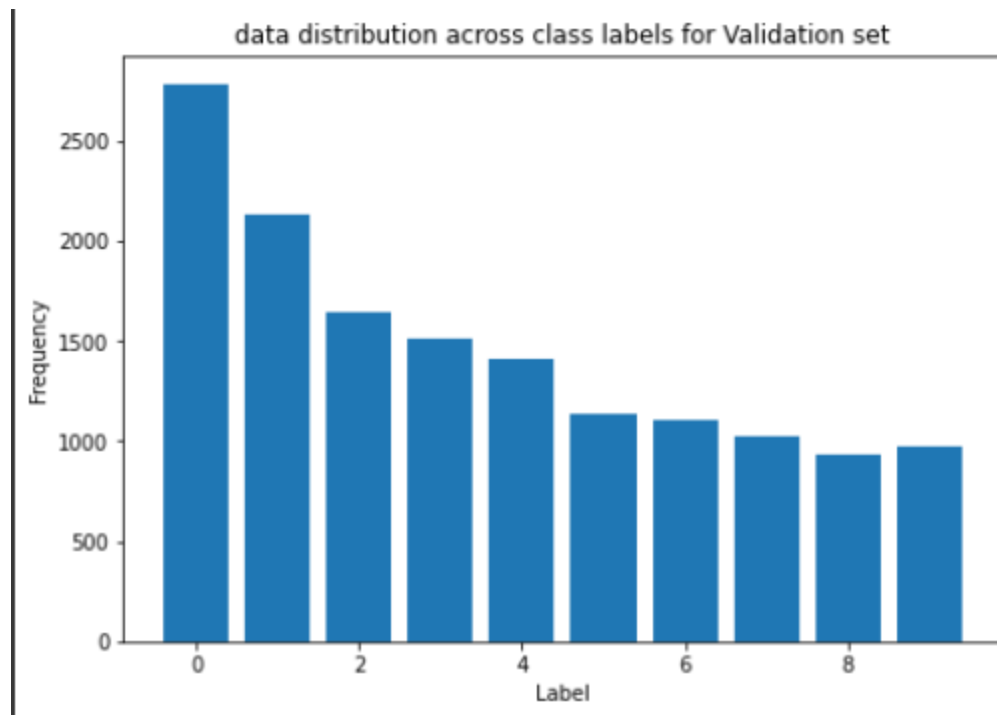
```
class CustomDataset():
    def __init__(self, dataset ,label,transform= None):
        self.label = label
        self.dataset = dataset
        self.n_samples = dataset.shape[3]
        self.transform = transform
        self.count=5;

    def __len__(self):
        return self.n_samples

    def __getitem__(self, idx):
        if self.transform!=None:
            return self.transform(self.dataset[:, :, :, idx]),self.label[idx]
        return self.dataset[:, :, :, idx],self.label[idx]
```

```
custom_transform = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225]))
dataset1 = CustomDataset(x_train,y_train,custom_transform)
dataset2 = CustomDataset(x_val,y_val,custom_transform)
dataset3 = CustomDataset(x_test,y_test,custom_transform)
```

c)



2.

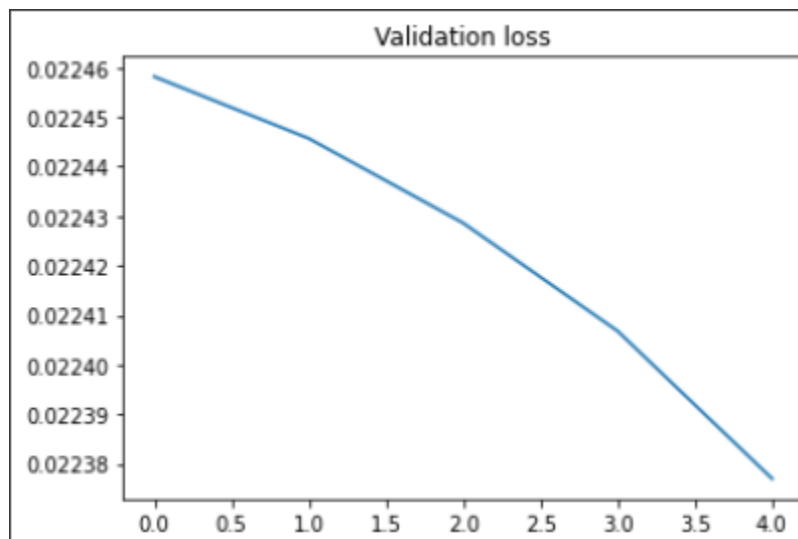
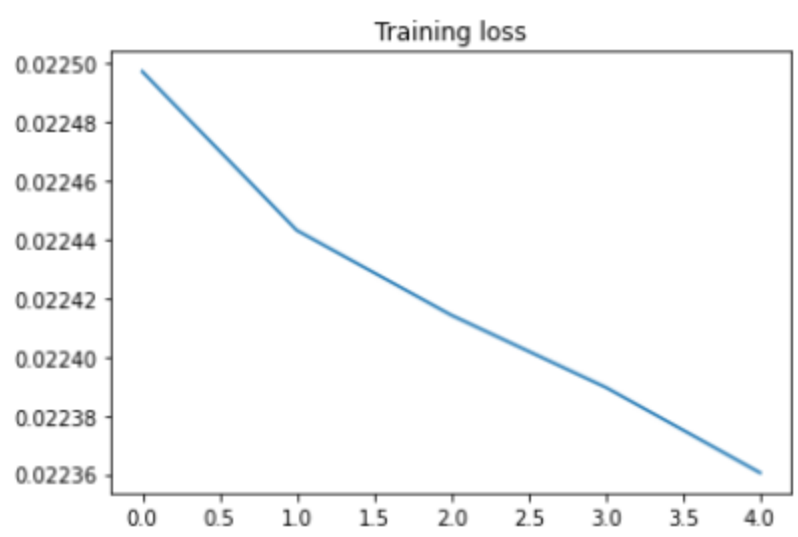
a)

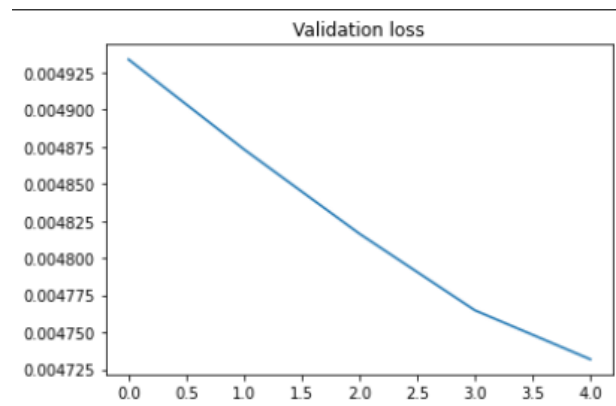
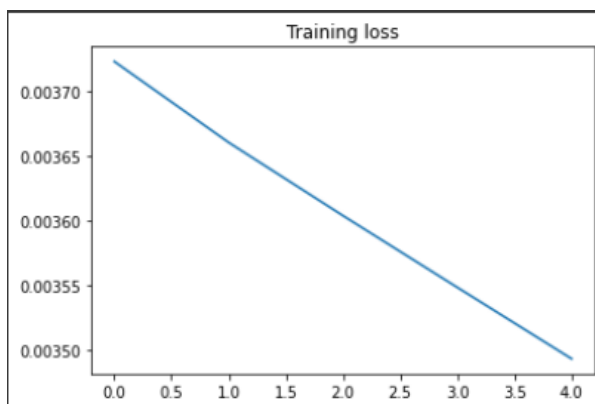
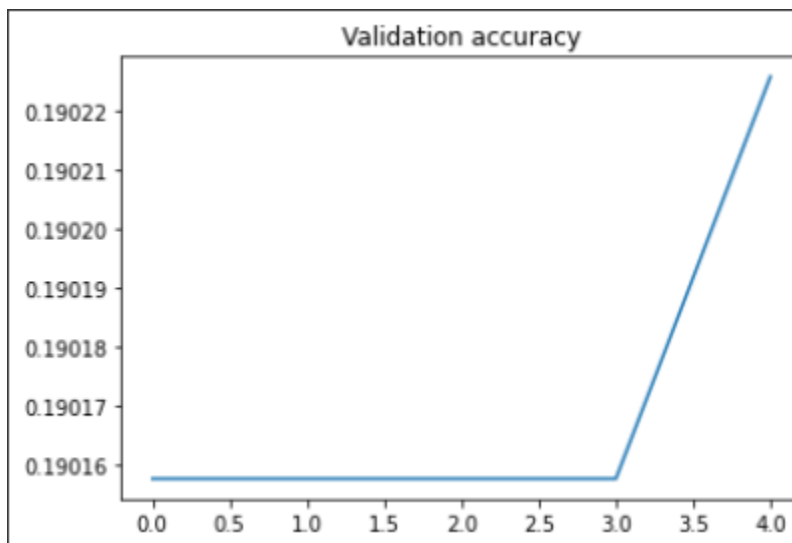
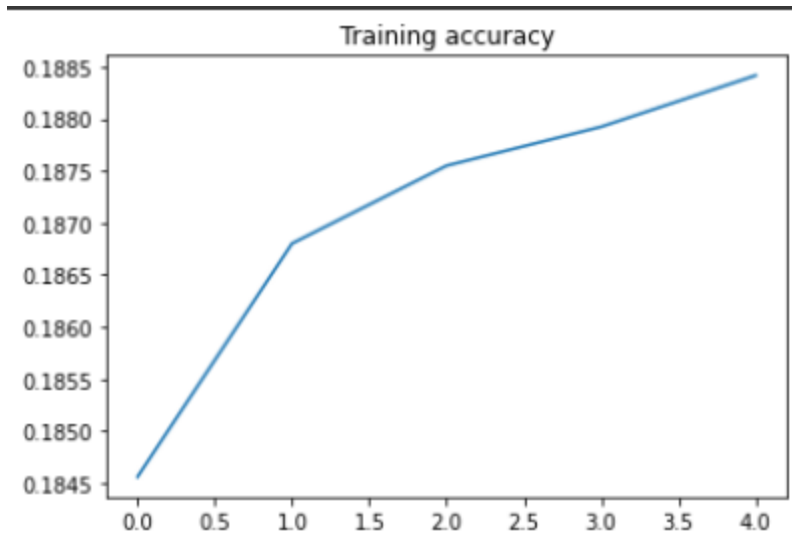
```
class Net(nn.Module):
    def __init__(self):
        super(Net,self).__init__()
        self.conv1 = nn.Conv2d(3, 32, 3,padding=1)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(32, 64, 3,padding=1)
        self.fc1 = nn.Linear(64 * 8 * 8, 64)
        self.fc2 = nn.Linear(64, 10)
        self.relu = nn.ReLU()

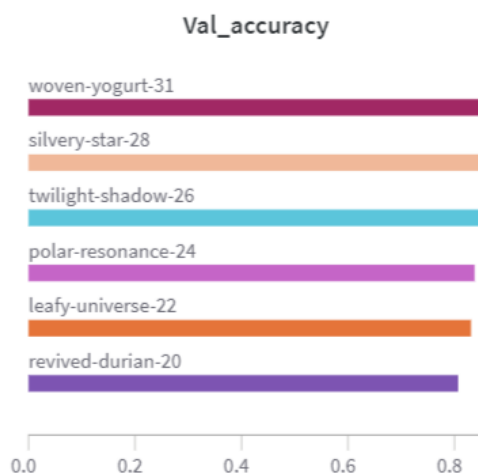
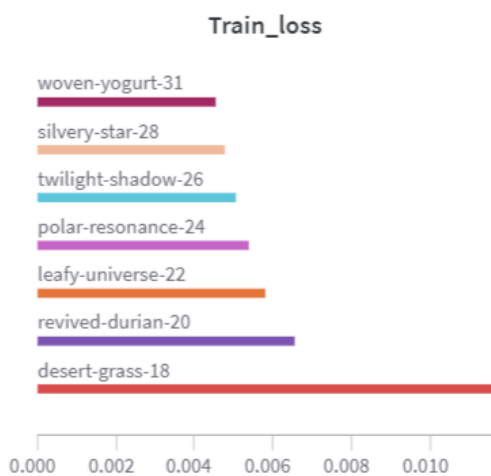
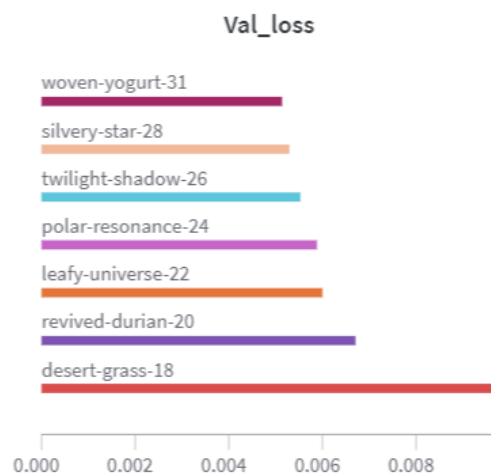
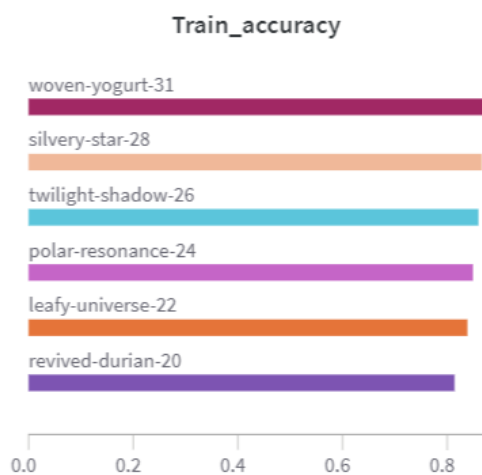
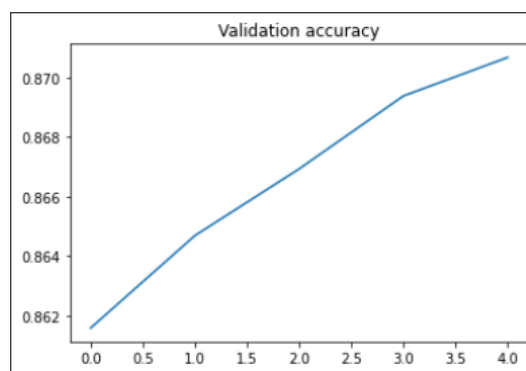
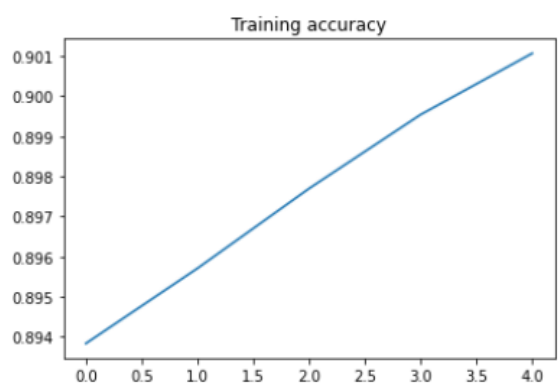
    def forward(self, x):
        x = self.conv1(x)
        x = self.relu(x)
        x = self.pool(x)
        x = self.conv2(x)
        x = self.relu(x)
        x = self.pool(x)
        x = x.view(-1, 64 * 8 * 8)
        x = self.fc1(x)
        x = self.relu(x)
        return self.fc2(x)

model = Net()
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(model.parameters(), lr=0.001,momentum=0.9)
```

b)







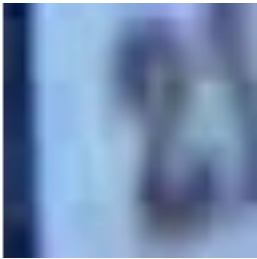
c)

classification_report for test set:				precision	recall	f1-score	support
0	0.84	0.90	0.87	1380			
1	0.71	0.94	0.81	1031			
2	0.65	0.79	0.71	868			
3	0.90	0.80	0.85	712			
4	0.87	0.68	0.76	723			
5	0.78	0.76	0.77	574			
6	0.89	0.76	0.82	553			
7	0.86	0.55	0.67	520			
8	0.80	0.63	0.70	470			
9	0.76	0.76	0.76	494			
accuracy			0.79	7325			
macro avg	0.81	0.76	0.77	7325			
weighted avg	0.80	0.79	0.79	7325			

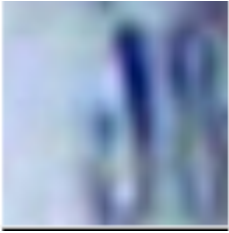

Accuracy of the network on the number of test images=7325 is =0.7877133105802048



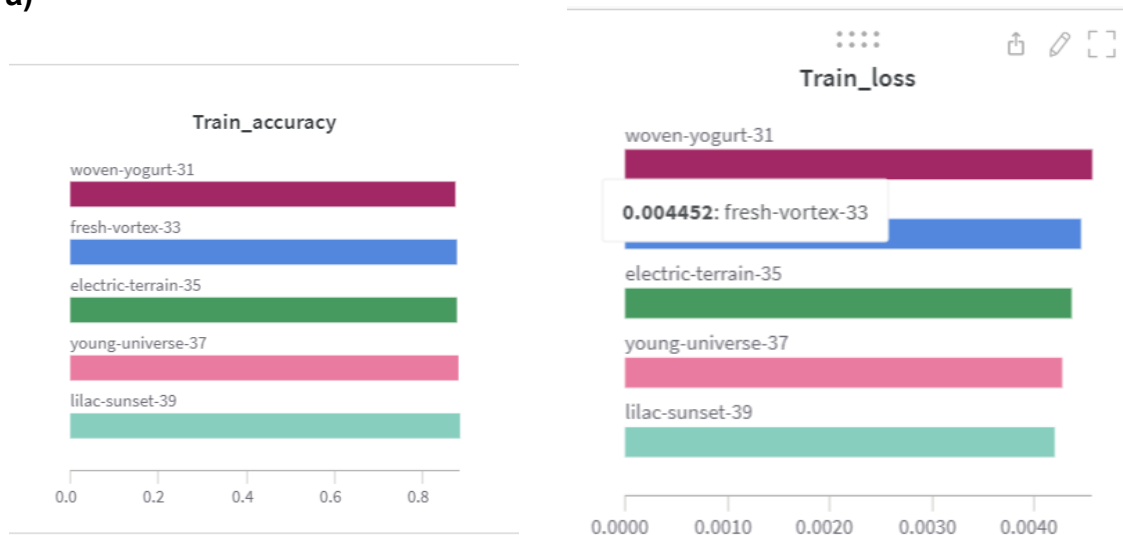
d)

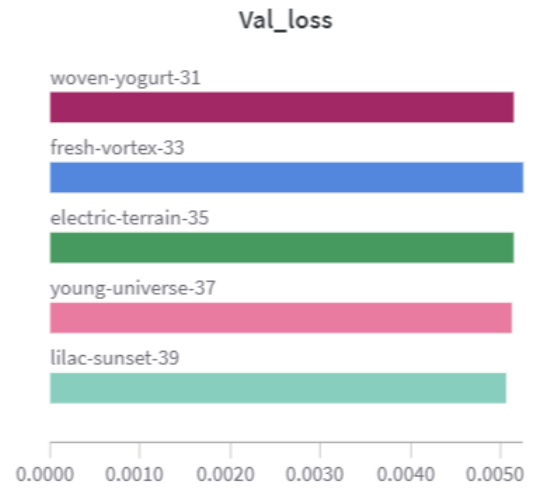
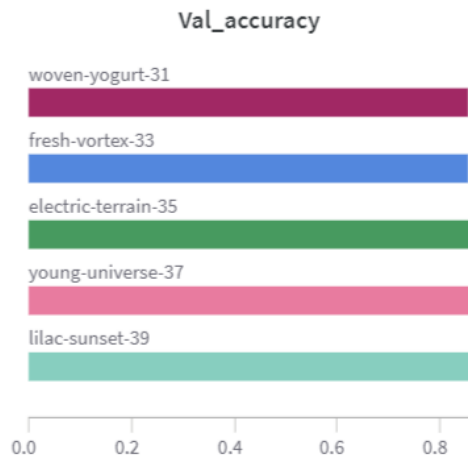
image	predicted	truth	reason
	8	2	poor quality of image



	8	9	similar looking 8 and 9
	4	1	1 got cut from below looks shorter and hence looks more similar to 4

3.  
a)

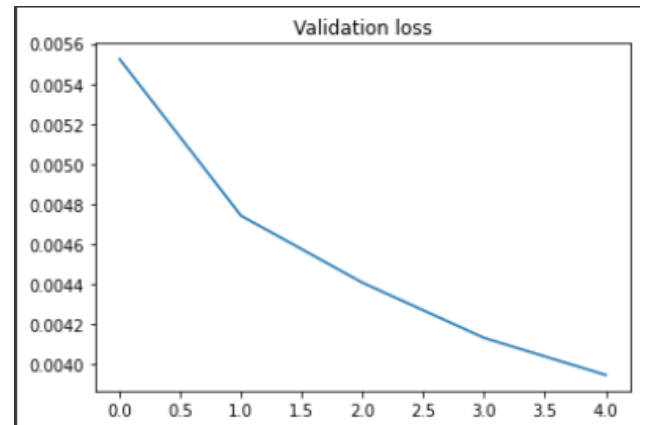
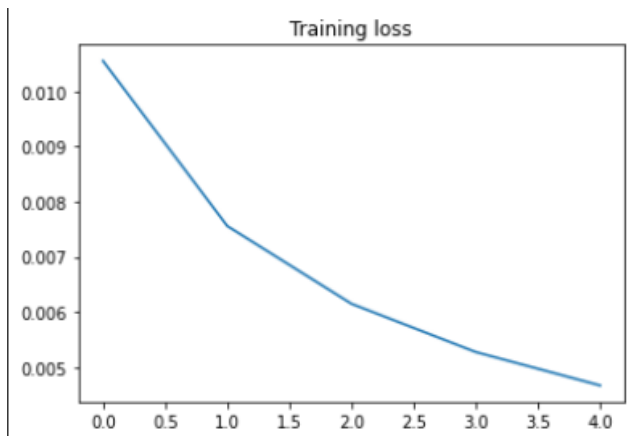


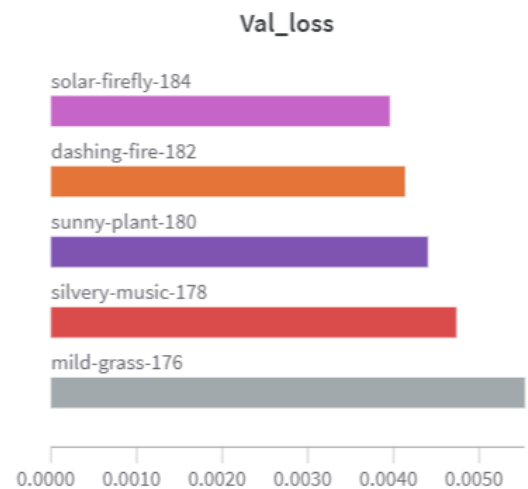
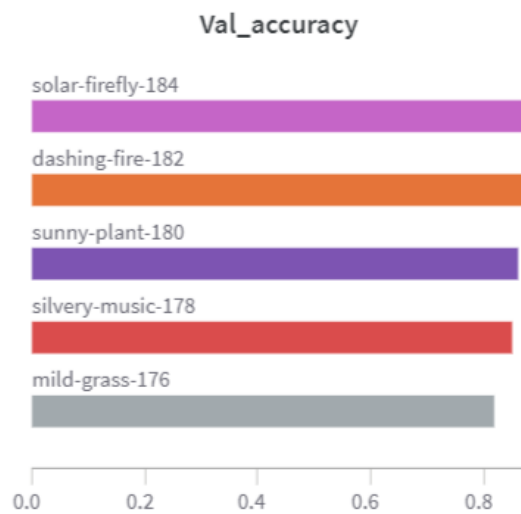
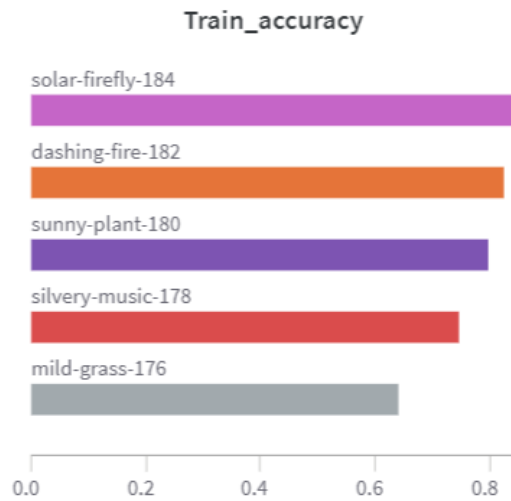
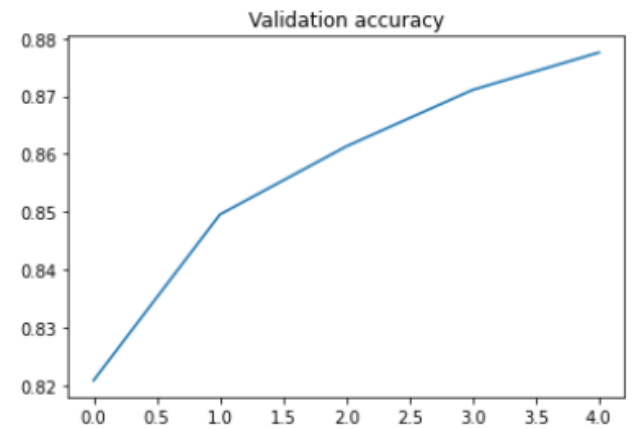
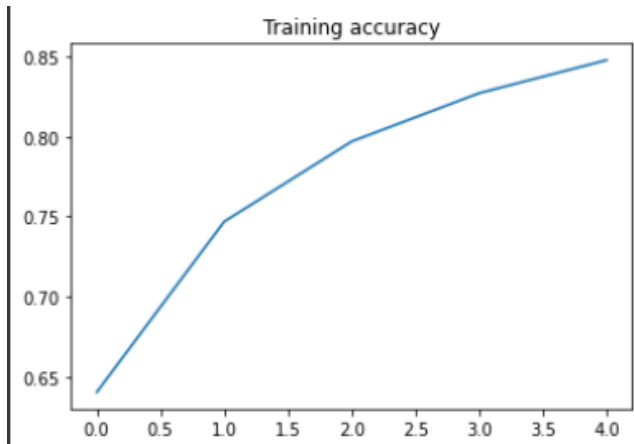


classification\_report for test set:

		precision	recall	f1-score	support
0	0.19	0.99	0.32	0.32	1380
1	0.25	0.02	0.03	0.03	1031
2	0.00	0.00	0.00	0.00	868
3	0.00	0.00	0.00	0.00	712
4	0.00	0.00	0.00	0.00	723
5	0.00	0.00	0.00	0.00	574
6	0.00	0.00	0.00	0.00	553
7	0.00	0.00	0.00	0.00	520
8	0.00	0.00	0.00	0.00	470
9	0.00	0.00	0.00	0.00	494
accuracy				0.19	7325
macro avg	0.04	0.10	0.03	0.03	7325
weighted avg	0.07	0.19	0.06	0.06	7325

Accuracy of the network on the number of test images=7325 is =0.1886689419795222

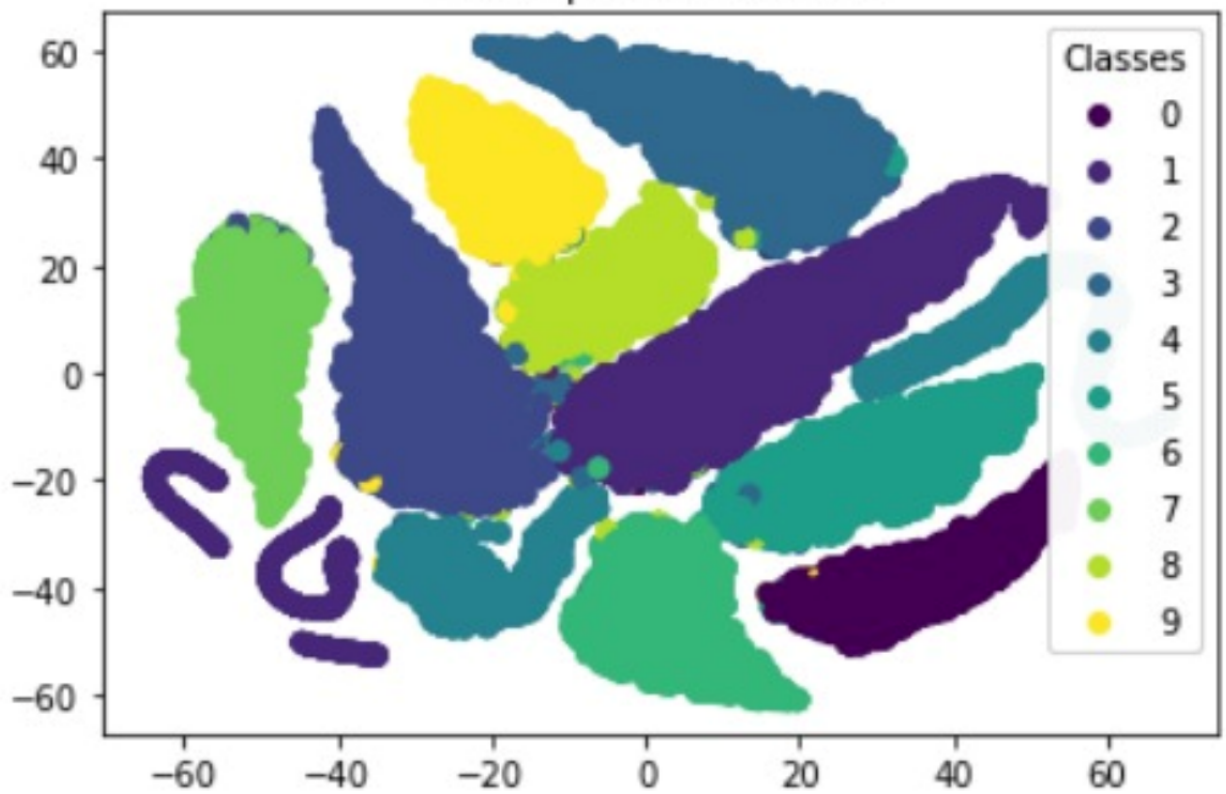


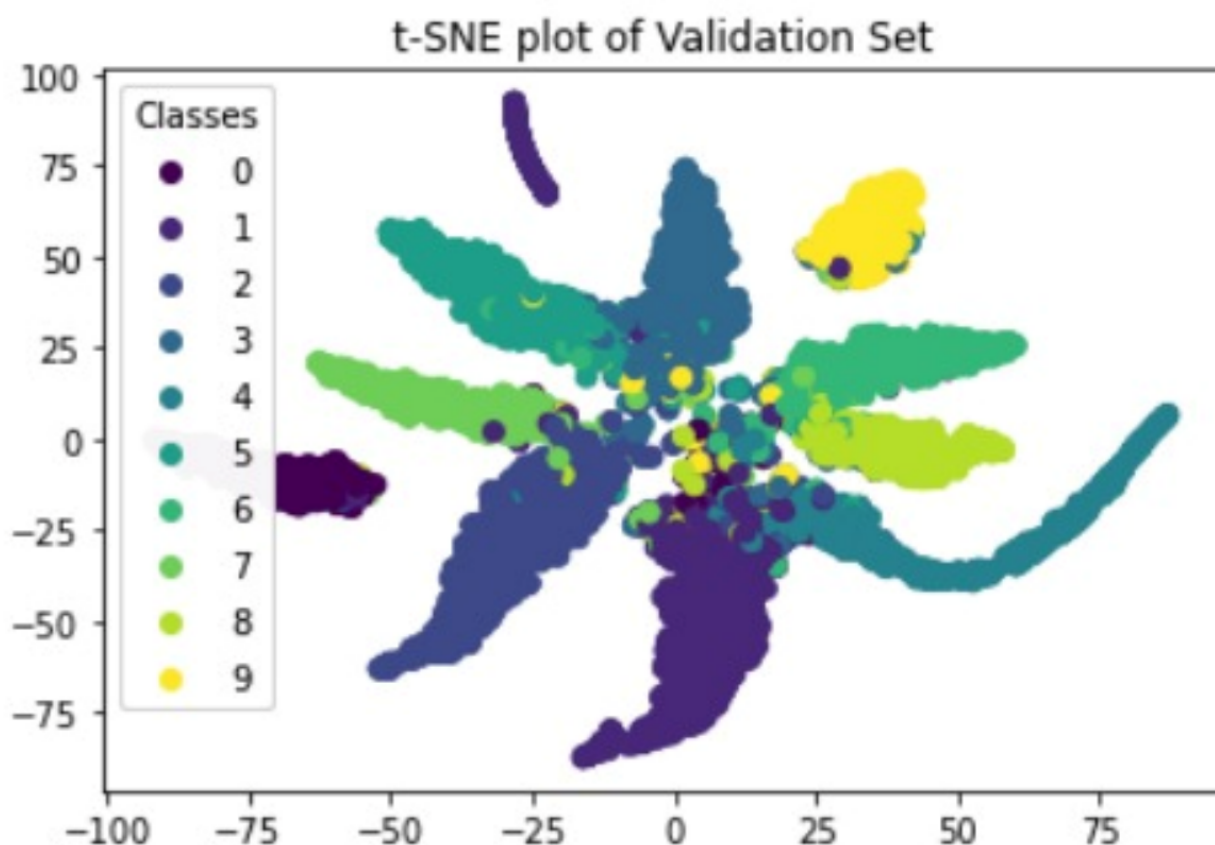


classification_report for test set:					precision	recall	f1-score	support
0	0.94	0.93	0.94	1380				
1	0.94	0.95	0.94	1031				
2	0.81	0.91	0.86	868				
3	0.91	0.96	0.93	712				
4	0.93	0.89	0.91	723				
5	0.91	0.88	0.89	574				
6	0.94	0.88	0.91	553				
7	0.84	0.86	0.85	520				
8	0.91	0.84	0.87	470				
9	0.94	0.89	0.91	494				
accuracy				0.91	7325			
macro avg				0.91	0.90	0.90	7325	
weighted avg				0.91	0.91	0.91	7325	
Accuracy of the network on the number of test images=7325 is =0.9074402730375427								



t-SNE plot of Train Set





```
[0.18785467735330252, 0.20343610444821467, 0.27977664671047925, 0.3831295462079994, 0.463616685192769]  
[0.022370771800196136, 0.02203950727317669, 0.020501319867716222, 0.017963928289008394, 0.01586943516881625]  
[0.19015766841853798, 0.23657088253361544, 0.3571542329306304, 0.45662412122039453, 0.5222578663572452]  
[0.022260608465449735, 0.02153647077766248, 0.018798406792997546, 0.01632786800666731, 0.014521929136958678]
```

```
training acc [0.6406521188010686, 0.7469724448604692, 0.7969344175978471, 0.8270539597106028, 0.847645234891476]  
training loss [0.010550434751907124, 0.007557684528515922, 0.006147157399708337, 0.0052725539865961435, 0.0046675886276750286]  
validation acc [0.820831342570473, 0.8495324551225172, 0.8613063954678861, 0.8710838850590403, 0.8775510204081632]  
validation loss [0.005523518353352798, 0.004741317377640651, 0.004407120102409769, 0.004131709552147463, 0.00394690775017341]
```

confusion\_matrix



Confusion Matrix

Epochs	CNN with just normalisation	Resnet18 without data augmentation	Resnet18 with data augmentation-normalisation, centercrop, RandomAutoContrast
1.	Training accuracy- <b>0.1878</b> Training Loss- <b>0.02237</b> Validation accuracy- <b>0.190157</b> Validation Loss- <b>0.02226</b>	Training accuracy- <b>0.18455</b> Training Loss- <b>0.02249</b> Validation accuracy- <b>0.19015</b> Validation Loss- <b>0.02245</b>	Training accuracy- <b>0.6406</b> Training Loss- <b>0.01055</b> Validation accuracy- <b>0.82083</b> Validation Loss- <b>0.00552</b>
2.	Training accuracy- <b>0.20343</b> Training Loss- <b>0.02203</b> Validation accuracy- <b>0.23657</b> Validation Loss- <b>0.02153</b>	Training accuracy- <b>0.1868</b> Training Loss- <b>0.02244</b> Validation accuracy- <b>0.19015</b> Validation Loss- <b>0.022445</b>	Training accuracy- <b>0.7469</b> Training Loss- <b>0.00755</b> Validation accuracy- <b>0.84953</b> Validation Loss- <b>0.0047</b>
3.	Training accuracy- <b>0.27977</b> Training Loss- <b>0.0205</b> Validation accuracy- <b>0.35715</b> Validation Loss- <b>0.01879</b>	Training accuracy- <b>0.18754</b> Training Loss- <b>0.02241</b> Validation accuracy- <b>0.1906</b> Validation Loss- <b>0.02242</b>	Training accuracy- <b>0.79683</b> Training Loss- <b>0.00614</b> Validation accuracy- <b>0.86130</b> Validation Loss- <b>0.00440</b>

4.	Training accuracy- <b>0.38312</b> Training Loss- <b>0.01796</b> Validation accuracy- <b>0.4566</b> Validation Loss- <b>0.01632</b>	Training accuracy- <b>0.18792</b> Training Loss- <b>0.02238</b> Validation accuracy- <b>0.1906</b> Validation Loss- <b>0.022406</b>	Training accuracy- <b>0.82705</b> Training Loss- <b>0.00527</b> Validation accuracy- <b>0.871108</b> Validation Loss- <b>0.00413</b>
5.	Training accuracy- <b>0.46361</b> Training Loss- <b>0.01586</b> Validation accuracy- <b>0.5222</b> Validation Loss- <b>0.0145</b>	Training accuracy- <b>0.18841</b> Training Loss- <b>0.02236</b> Validation accuracy- <b>0.190225</b> Validation Loss- <b>0.0223</b>	Training accuracy- <b>0.8476</b> Training Loss- <b>0.00466</b> Validation accuracy- <b>0.87755</b> Validation Loss- <b>0.00394</b>
Testing Accuracy F1-score:	<b>0.787713</b> <b>0.778</b>	<b>0.18766</b> <b>0.189</b>	<b>0.90744</b> <b>0.90</b>

Ques 2 -  
2.1 done





```
dataset1.plotMask(5)
```



3.  
done till 3.1