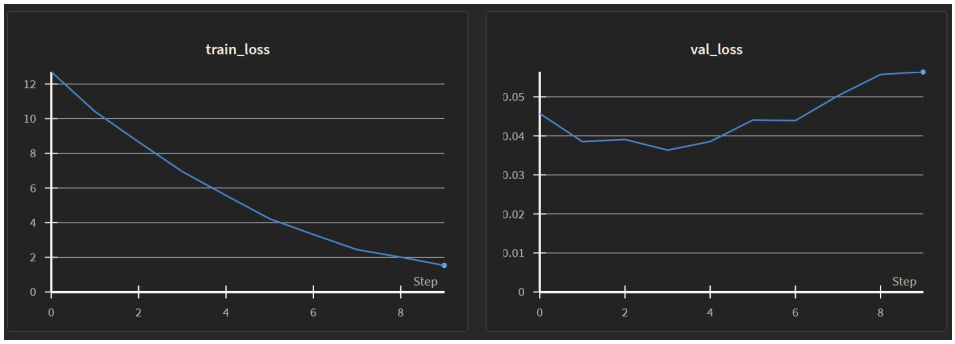
Q2)

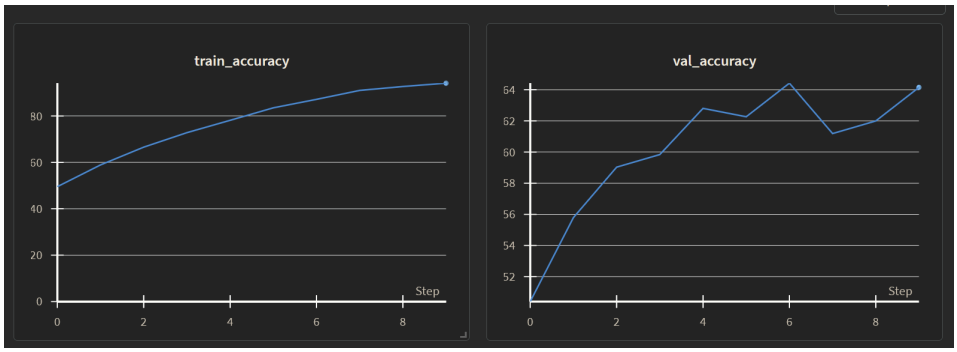
1.

2.

CNN Model

As per the training and validation loss plots and accuracy plots, the model shows clear signs **overfitting**. The accuracy on the train set is much higher compared to the accuracy on the val set.

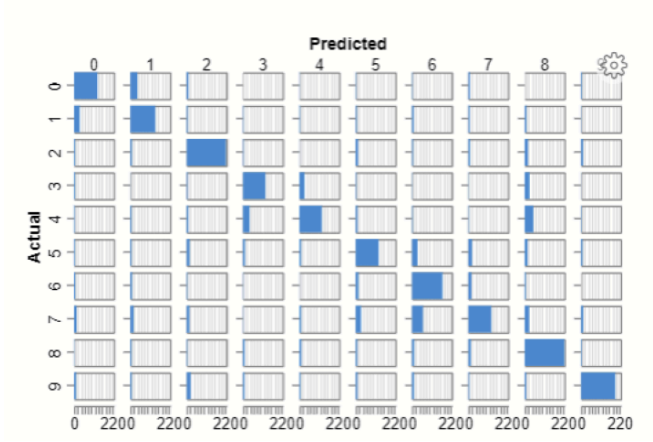




On the test set :

```
Test Accuracy: 65.6269
Test F1-Score: 65.5823
```

Confusion Matrix :



Class Mapping :

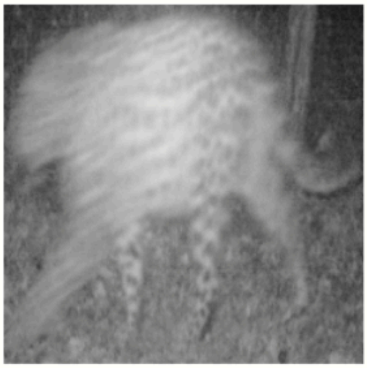
```
class_mapping = {
    "amur_leopard": 0,
    "amur_tiger": 1,
    "birds": 2,
    "black_bear": 3,
    "brown_bear": 4,
    "dog": 5,
    "roe_deer": 6,
    "sika_deer": 7,
    "wild_boar": 8,
    "people": 9,
}
```

Miss classified examples :

Misclassified Images for Class 0

Predicted: 1

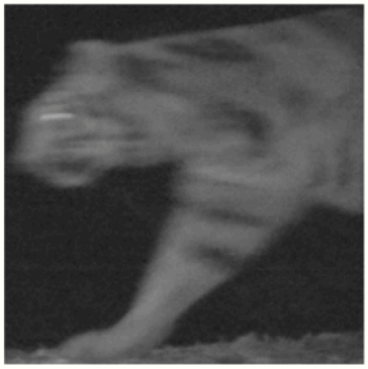
Predicted: 6



Predicted: 1

Misclassified Images for Class 1

Predicted: 5



Predicted: 6

Predicted: 5

Misclassified Images for Class 2

Predicted: 8

Predicted: 9

Predicted: 5

Misclassified Images for Class 3

Predicted: 9

Predicted: 4

Predicted: 4

Misclassified Images for Class 4

Predicted: 3

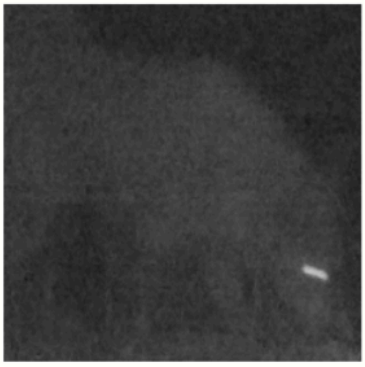
Predicted: 7

Predicted: 0

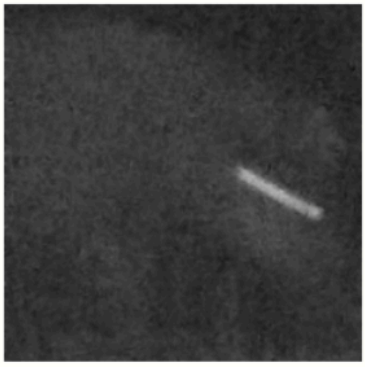
Misclassified Images for Class 5

Predicted: 8

Predicted: 6



Predicted: 6



Misclassified Images for Class 6

Predicted: 7

Predicted: 1

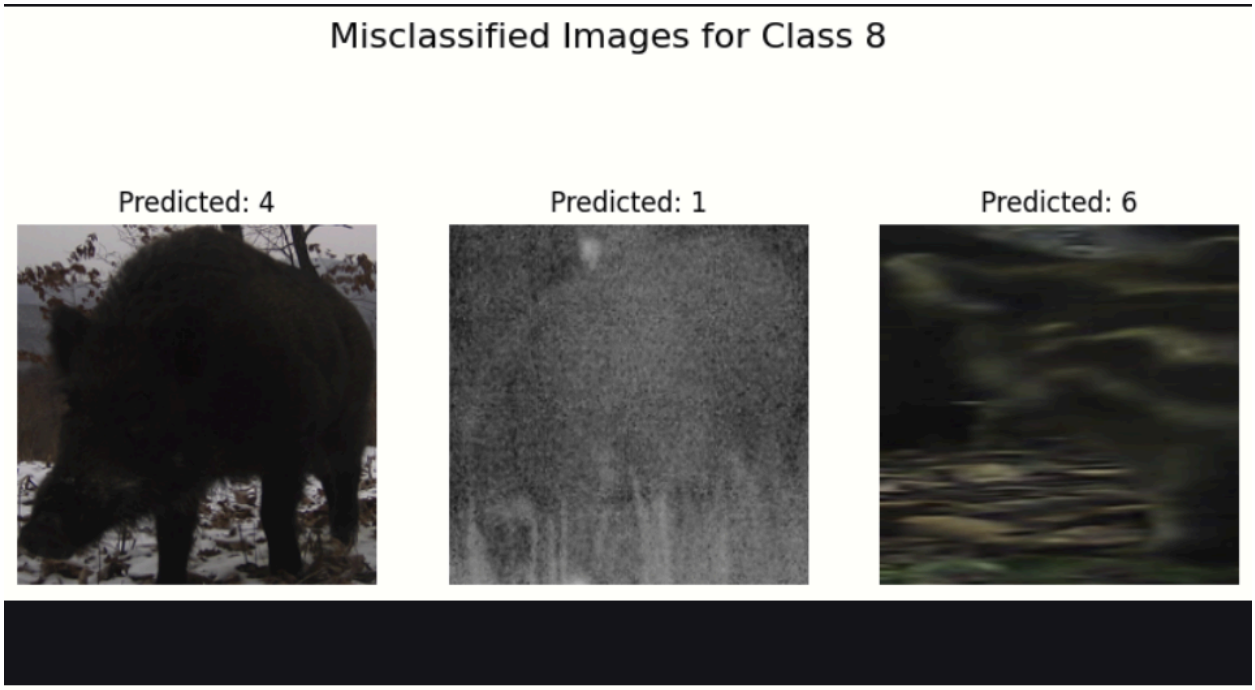
Predicted: 8

Misclassified Images for Class 7

Predicted: 6

Predicted: 6

Predicted: 5



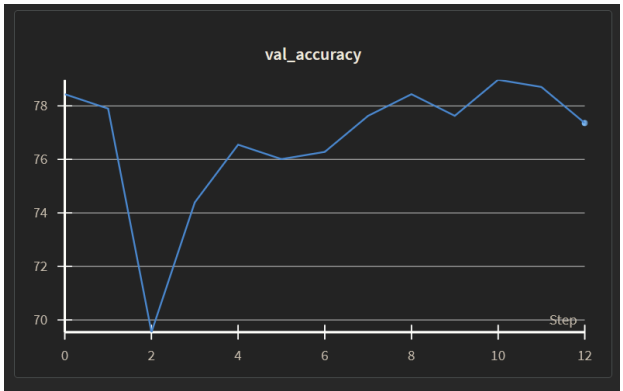
Possible reasons for misclassification of these images :

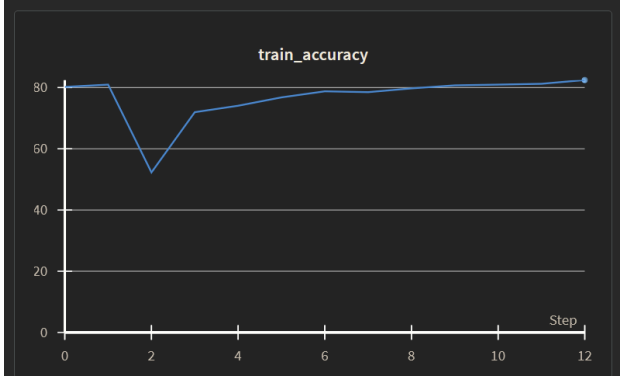
a) Image has low visibility or partial visibility of ground truth class
b) The image resembles examples from another true label.

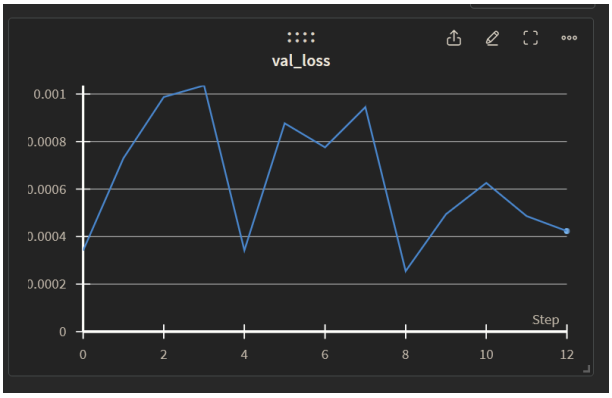
Workaround :
Use more such training examples so that the model learns more discriminative features.

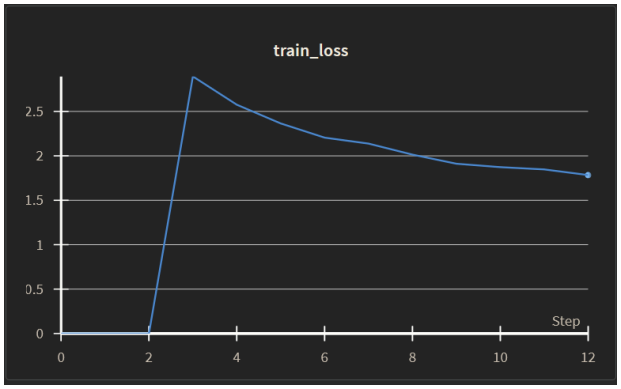
3.

ResNet Model :









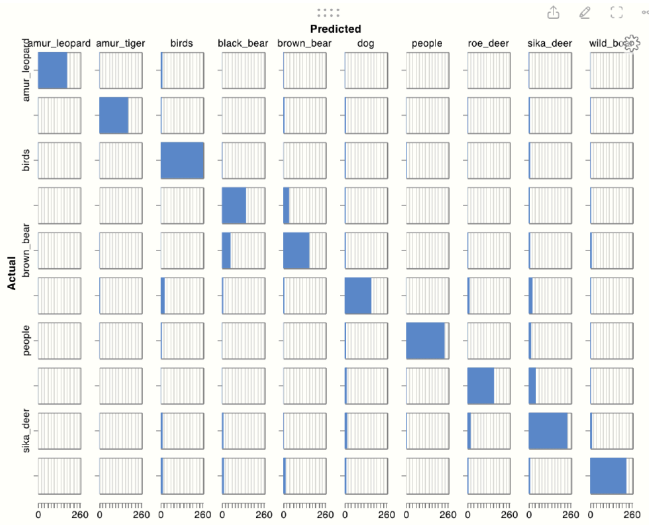
On test set :

```
Test Accuracy: 84.9430
Test F1-Score: 84.9598
```

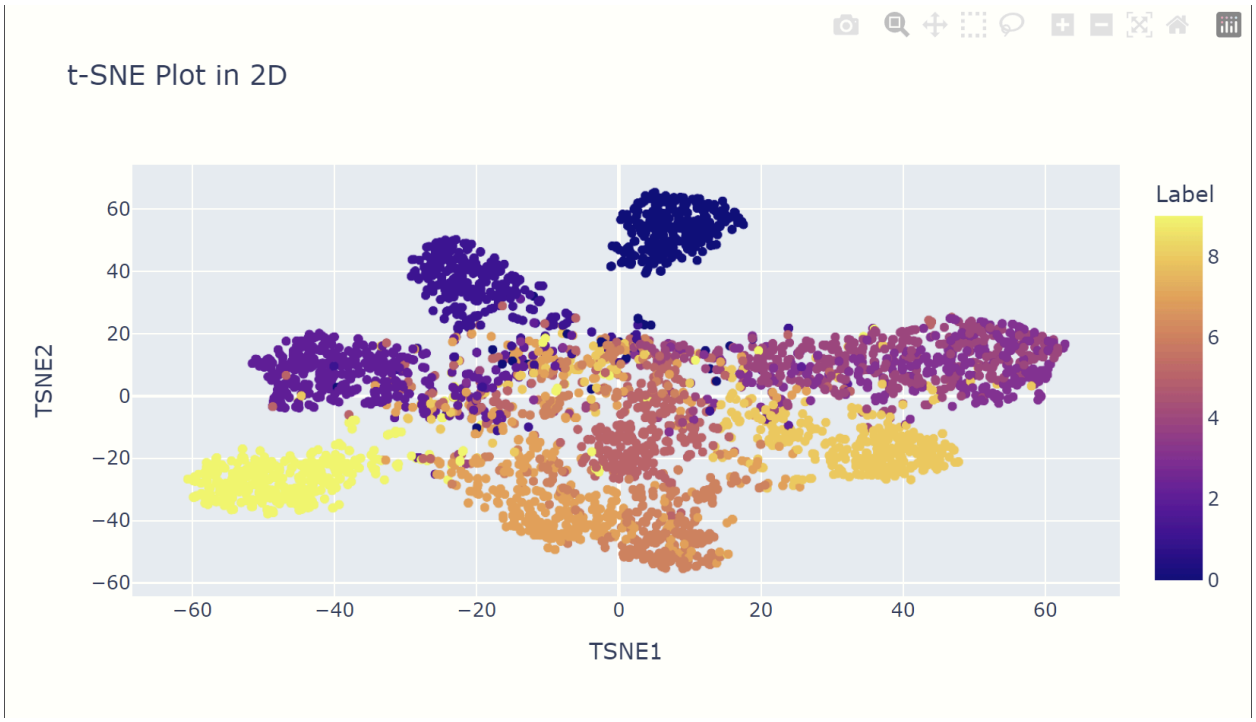
The ResNet model is not overfitting

The ResNet model does not overfit the train data, as the difference between train accuracy and val accuracy is within an acceptable range.

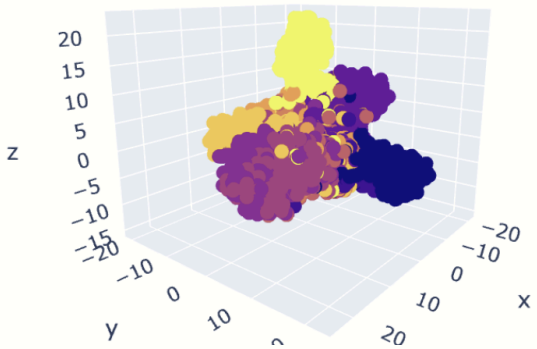
Confusion Matrix :



TSNE plots (train + val data):



t-SNE Plot in 3D



4.

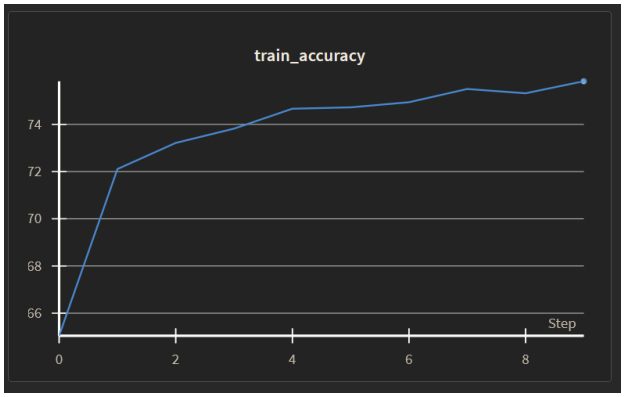
ResNet18 on augmented data :

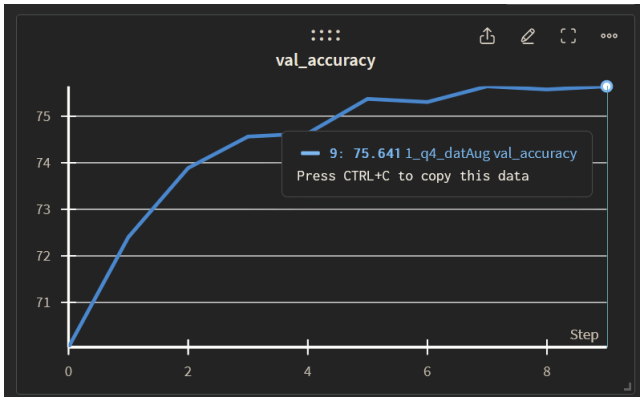
Augmentations used :

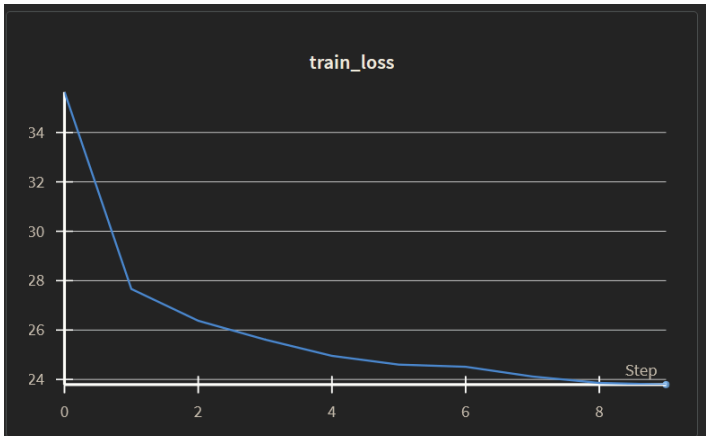
```
import albumentations as A



aug = A.Compose([
    A.VerticalFlip(p=0.5),
    A.RandomRotate90(p=0.5),
    A.HorizontalFlip(p=1),
    A.Transpose(p=1)
])
```

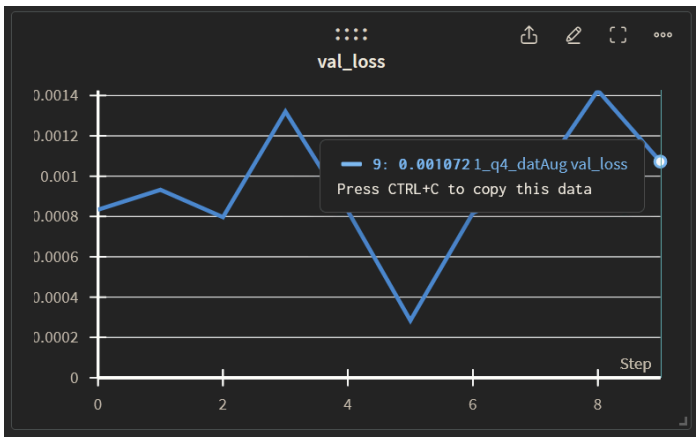
New Data distribution :


Training Set Class Distribution
Number of Samples
2500
0
0
2
4
6
8
Class Label
Test Set Class Distribution
Number of Samples
1000
0
0
2
4
6
8
Class Label
Validation Set Class Distribution
Number of Samples
100
0
0
2
4
6
8
Class Label


train_accuracy
74
72
70
68
66
Step
0
2
4
6
8


val_accuracy
75
74
73
72
71
9: 75.641 1_q4_datAug val_accuracy
Press CTRL+C to copy this data
Step
0
2
4
6
8



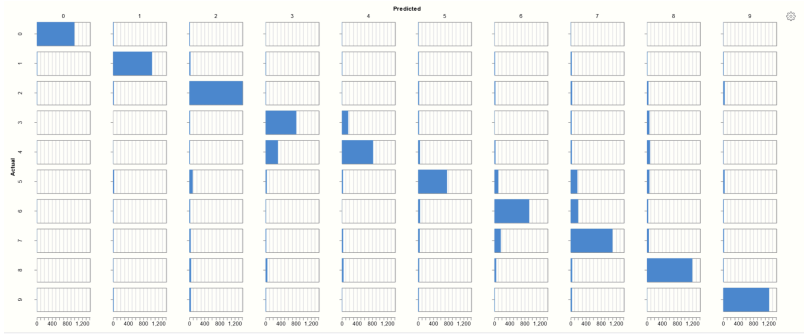


On test set :

```
Test Accuracy: 77.0379
Test F1-Score: 77.1443
```

The ResNet model is not overfitting

Confusion Matrix :



5.

For CNN model :

True class : 1 | Predicted class : 0 | Euclidean Distance : 38.34449768066406
True class : 7 | Predicted class : 6 | Euclidean Distance : 47.652523040771484
True class : 2 | Predicted class : 4 | Euclidean Distance : 10.348711967468262
True class : 1 | Predicted class : 7 | Euclidean Distance : 17.39586067199707
True class : 6 | Predicted class : 1 | Euclidean Distance : 11.211783409118652
True class : 2 | Predicted class : 9 | Euclidean Distance : 29.04446792602539
True class : 4 | Predicted class : 3 | Euclidean Distance : 79.87519836425781
True class : 1 | Predicted class : 7 | Euclidean Distance : 35.29945373535156
True class : 9 | Predicted class : 2 | Euclidean Distance : 42.56366729736328
True class : 6 | Predicted class : 5 | Euclidean Distance : 24.574419021606445
True class : 9 | Predicted class : 8 | Euclidean Distance : 44.31101608276367
True class : 5 | Predicted class : 2 | Euclidean Distance : 53.4626579284668
True class : 7 | Predicted class : 9 | Euclidean Distance : 16.98307228088379
True class : 4 | Predicted class : 0 | Euclidean Distance : 16.356609344482422
True class : 6 | Predicted class : 7 | Euclidean Distance : 46.811973571777344
True class : 4 | Predicted class : 3 | Euclidean Distance : 52.65633010864258
True class : 5 | Predicted class : 7 | Euclidean Distance : 43.43388366699219
True class : 5 | Predicted class : 6 | Euclidean Distance : 11.694635391235352
True class : 7 | Predicted class : 9 | Euclidean Distance : 70.65845489501953
True class : 8 | Predicted class : 5 | Euclidean Distance : 92.35383605957031
True class : 3 | Predicted class : 4 | Euclidean Distance : 10.841940879821777
True class : 0 | Predicted class : 5 | Euclidean Distance : 32.37778854370117
True class : 2 | Predicted class : 6 | Euclidean Distance : 8.25412654876709
True class : 3 | Predicted class : 5 | Euclidean Distance : 50.738365173339844
True class : 8 | Predicted class : 6 | Euclidean Distance : 10.146210670471191
True class : 3 | Predicted class : 4 | Euclidean Distance : 18.708742141723633
True class : 0 | Predicted class : 2 | Euclidean Distance : 20.11578941345215
True class : 8 | Predicted class : 4 | Euclidean Distance : 10.916414260864258
True class : 0 | Predicted class : 6 | Euclidean Distance : 12.493598937988281
True class : 9 | Predicted class : 2 | Euclidean Distance : 82.06773376464844

For ResNet18 :

True class : 0 | Predicted class : 2 | Euclidean Distance : 13.99483871459961
True class : 5 | Predicted class : 2 | Euclidean Distance : 13.99283504486084
True class : 9 | Predicted class : 2 | Euclidean Distance : 13.984682083129883
True class : 0 | Predicted class : 2 | Euclidean Distance : 13.99399471282959
True class : 4 | Predicted class : 2 | Euclidean Distance : 14.01704216003418
True class : 9 | Predicted class : 6 | Euclidean Distance : 13.990453720092773
True class : 4 | Predicted class : 2 | Euclidean Distance : 13.99814510345459
True class : 9 | Predicted class : 6 | Euclidean Distance : 13.955217361450195
True class : 2 | Predicted class : 1 | Euclidean Distance : 13.974799156188965
True class : 7 | Predicted class : 2 | Euclidean Distance : 14.001065254211426
True class : 5 | Predicted class : 0 | Euclidean Distance : 13.988438606262207
True class : 1 | Predicted class : 2 | Euclidean Distance : 14.001882553100586
True class : 4 | Predicted class : 2 | Euclidean Distance : 13.999616622924805
True class : 6 | Predicted class : 2 | Euclidean Distance : 13.993295669555664
True class : 3 | Predicted class : 2 | Euclidean Distance : 13.99374771118164
True class : 6 | Predicted class : 2 | Euclidean Distance : 14.003044128417969
True class : 1 | Predicted class : 2 | Euclidean Distance : 13.986016273498535
True class : 6 | Predicted class : 2 | Euclidean Distance : 13.983423233032227
True class : 7 | Predicted class : 2 | Euclidean Distance : 13.996935844421387
True class : 5 | Predicted class : 2 | Euclidean Distance : 14.032912254333496
True class : 3 | Predicted class : 2 | Euclidean Distance : 14.00444221496582
True class : 0 | Predicted class : 2 | Euclidean Distance : 13.97738265991211
True class : 7 | Predicted class : 2 | Euclidean Distance : 13.982389450073242
True class : 8 | Predicted class : 1 | Euclidean Distance : 13.98779296875
True class : 1 | Predicted class : 2 | Euclidean Distance : 13.970945358276367
True class : 3 | Predicted class : 2 | Euclidean Distance : 14.005023956298828
True class : 8 | Predicted class : 2 | Euclidean Distance : 14.01069450378418
True class : 8 | Predicted class : 0 | Euclidean Distance : 14.013017654418945
True class : 2 | Predicted class : 0 | Euclidean Distance : 13.987284660339355
True class : 2 | Predicted class : 1 | Euclidean Distance : 13.996744155883789

For ResNet18 trained on augmented data :

True class : 3 | Predicted class : 4 | Euclidean Distance : 9.510709762573242
True class : 4 | Predicted class : 3 | Euclidean Distance : 9.612300872802734
True class : 1 | Predicted class : 6 | Euclidean Distance : 10.192978858947754
True class : 6 | Predicted class : 7 | Euclidean Distance : 6.827943325042725
True class : 3 | Predicted class : 4 | Euclidean Distance : 5.6907548904418945
True class : 2 | Predicted class : 5 | Euclidean Distance : 14.762918472290039
True class : 2 | Predicted class : 9 | Euclidean Distance : 12.858320236206055
True class : 5 | Predicted class : 4 | Euclidean Distance : 6.6757941246032715
True class : 5 | Predicted class : 0 | Euclidean Distance : 12.00147533416748
True class : 2 | Predicted class : 6 | Euclidean Distance : 8.511661529541016
True class : 3 | Predicted class : 6 | Euclidean Distance : 9.673066139221191
True class : 4 | Predicted class : 3 | Euclidean Distance : 11.033934593200684
True class : 6 | Predicted class : 8 | Euclidean Distance : 11.003650665283203
True class : 6 | Predicted class : 5 | Euclidean Distance : 9.096470832824707
True class : 1 | Predicted class : 9 | Euclidean Distance : 22.615514755249023
True class : 7 | Predicted class : 4 | Euclidean Distance : 6.9136857986450195
True class : 4 | Predicted class : 3 | Euclidean Distance : 13.20931625366211
True class : 7 | Predicted class : 6 | Euclidean Distance : 7.581525802612305
True class : 9 | Predicted class : 2 | Euclidean Distance : 10.76678466796875
True class : 1 | Predicted class : 0 | Euclidean Distance : 10.614912033081055
True class : 8 | Predicted class : 6 | Euclidean Distance : 14.685976028442383
True class : 5 | Predicted class : 7 | Euclidean Distance : 15.218098640441895
True class : 7 | Predicted class : 6 | Euclidean Distance : 12.271258354187012
True class : 8 | Predicted class : 4 | Euclidean Distance : 13.585073471069336
True class : 0 | Predicted class : 6 | Euclidean Distance : 9.986876487731934
True class : 9 | Predicted class : 2 | Euclidean Distance : 7.631399154663086
True class : 8 | Predicted class : 7 | Euclidean Distance : 9.942893028259277
True class : 9 | Predicted class : 7 | Euclidean Distance : 10.257116317749023
True class : 0 | Predicted class : 2 | Euclidean Distance : 12.805451393127441
True class : 0 | Predicted class : 7 | Euclidean Distance : 10.250999450683594

6.

Q3)

1.

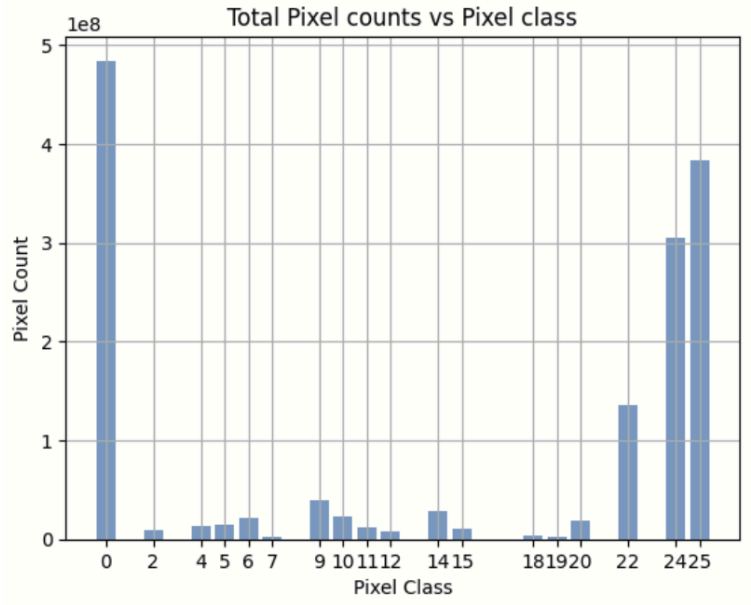
a) Custom Dataloader

```
class dataloader(data.Dataset) :
    def __init__(self,folder_path,to_test=False):
        super(dataloader,self).__init__()
        self.img_files = sorted(glob.glob(os.path.join(folder_path,'image_archive','*.jpg')))
        self.mask_files = sorted(glob.glob(os.path.join(folder_path,'mask_archive','*.jpg')))

        self.pixel_class =  [0,2,4,5,6,7,9,10,11,12,14,15,18,19,20,22,24,25]
        self.pixel_dict = {key: 0 for key in self.pixel_class}

        self.images = []
        self.masks = []
        self.indices = []

        if to_test == True :
            test_size = int(0.001*len(self.img_files))
            random.seed(4)
            self.indices = random.sample(range(len(self.img_files)), test_size)
            self.images = [self.img_files[i] for i in self.indices]
            self.masks = [self.mask_files[i] for i in self.indices]
        else :
            self.indices = np.arange(len(self.img_files)+1)
            self.images = self.img_files
            self.masks = self.mask_files


    # gives the og img, masked img and its corresponding id/index
    def __getitem__(self, index):
        img = self.images[index]
        mask = self.masks[index]

        data_img = Image.open(img).resize((512,512))
        data_tensor = transform(data_img)

        mask_img = Image.open(mask).resize((512,512))
        mask_tensor = transform(mask_img)

        return data_tensor, mask_tensor, self.indices[index]

    def __len__(self) :
        return len(self.images)
```

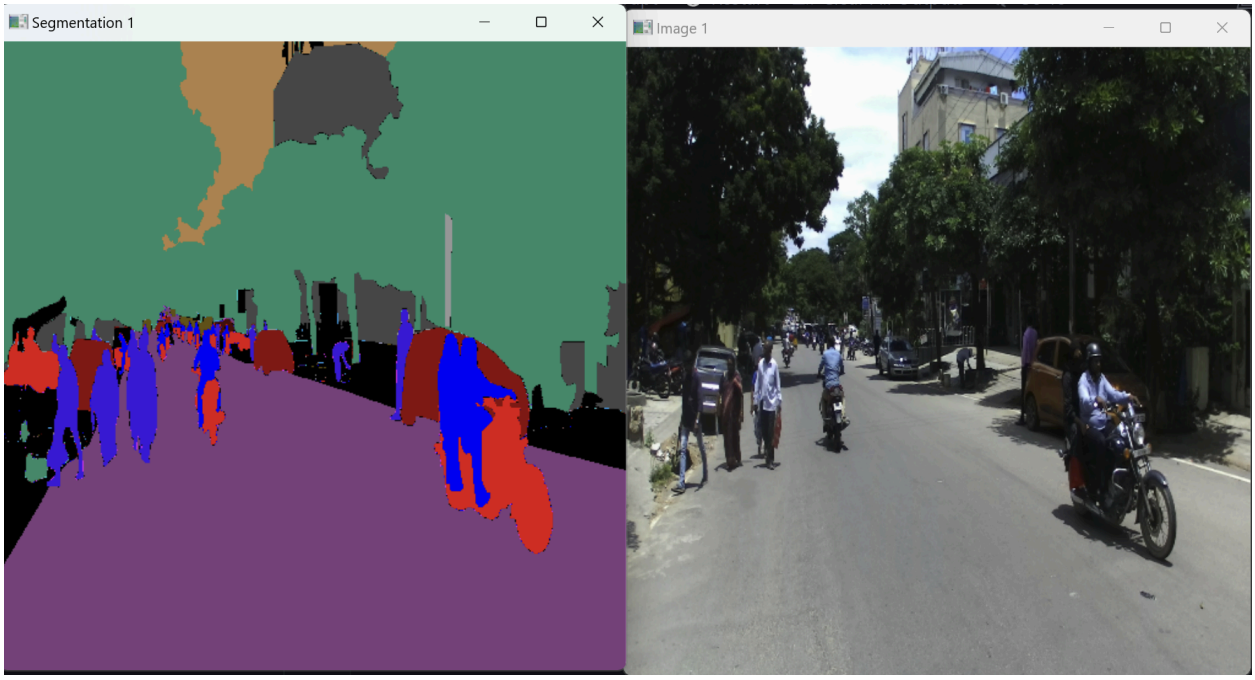
b) Data distribution (pixel count vs class) :

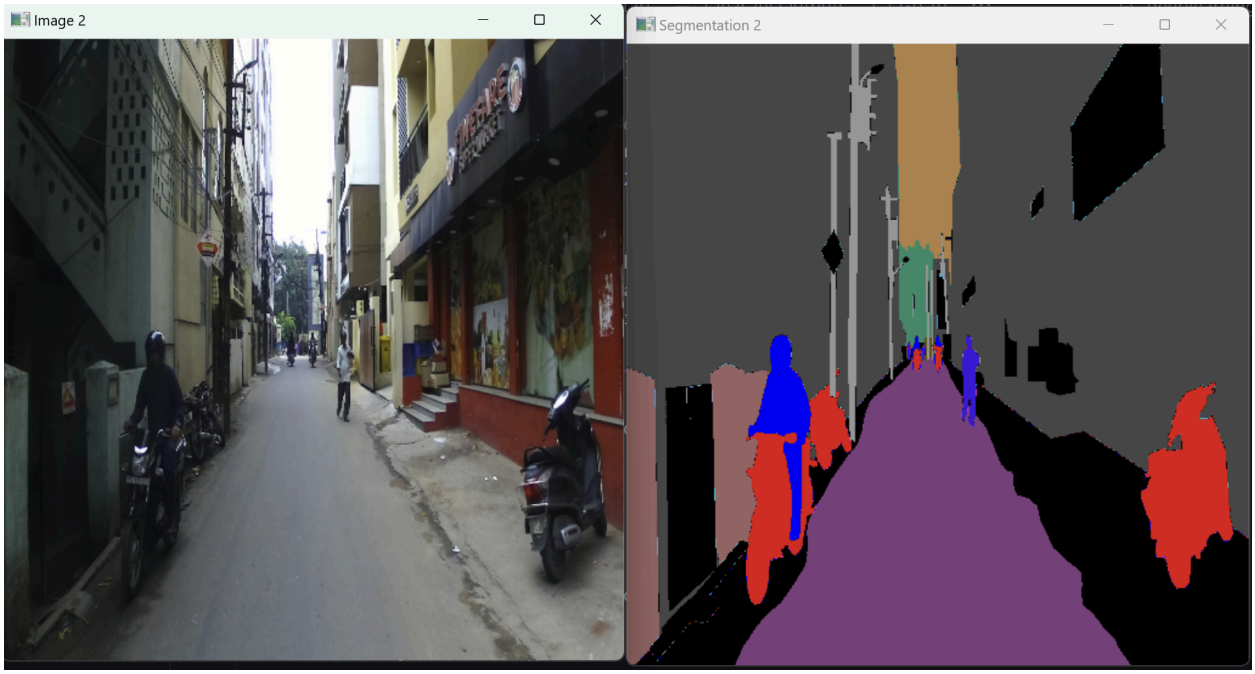
```
color_mapping = {
    0 : [128, 64,128],      # Road
    2: [244, 35,232],      # Sidewalk
    4: [220, 20, 60],    # Person/animal
    5: [255,  0,  0],    # Rider
    6: [0,  0,230],    # Motorcycle
    7: [119, 11, 32],    # bicycle
    9: [0,  0,142],      # Car
    10: [  0,  0, 70],   # Truck
    11: [0, 60,100],   # Bus
    12: [0, 80,100],   # Train/caravan/trailer (heavy vehicals)
    14: [102,102,156],      # Wall
    15: [190,153,153],      # Fence
    18: [220,220,  0],     # Traffic Sign
    19: [250,170, 30],     # Traffic Light
```

```
    20: [153,153,153],    # Pole
    22: [70, 70, 70],   # Building
    24: [107,142, 35],   # Vegetation
    25: [70,130,180],    # Sky
    255: [0, 0, 0]
}
```



c)





2 and 3.

Class wise pixel accuracy :

accuracy for class 0 : 0.8855450948079427
accuracy for class 2 : 0.9852333068847656
accuracy for class 4 : 0.8266658782958984
accuracy for class 5 : 0.9846789042154948
accuracy for class 6 : 0.9832731882731119
accuracy for class 7 : 0.9956995646158854
accuracy for class 9 : 0.9897632598876953
accuracy for class 10 : 0.9876244862874349
accuracy for class 11 : 0.9635670979817709
accuracy for class 12 : 0.9630559285481771
accuracy for class 14 : 0.9759667714436849
accuracy for class 15 : 0.9814726511637369
accuracy for class 18 : 0.9971326192220052
accuracy for class 19 : 0.9975414276123047
accuracy for class 20 : 0.9858779907226562
accuracy for class 22 : 0.8851095835367838
accuracy for class 24 : 0.8593940734863281
accuracy for class 25 : 0.8080043792724609

Class wise precision :

precision for class 0 : 0.7988786524179755
precision for class 2 : 0.5530779164873009
precision for class 4 : 0.00016385872398742756
precision for class 5 : 0.05744877438991249

precision for class 6 : 0.011117728806829461
precision for class 7 : 0.0003114779629341224
precision for class 9 : 0.030610196708149337
precision for class 10 : 0.0
precision for class 11 : 0.3764030279300444
precision for class 12 : 0.0006107191744761503
precision for class 14 : 0.001021266370299171
precision for class 15 : 0.41295244413118287
precision for class 18 : 0.20082860880941997
precision for class 19 : 0.0009115770282588879
precision for class 20 : 0.03353119474614802
precision for class 22 : 0.6088315098063687
precision for class 24 : 0.44956120880997097
precision for class 25 : 0.06348635171952913

Class wise IOU :

iou for class 0 : 0.6652640465002985
iou for class 2 : 0.4534801637724128
iou for class 4 : 0.000161364261483451
iou for class 5 : 0.042019479228781555
iou for class 6 : 0.006345129735241908
iou for class 7 : 0.0002955956251847473
iou for class 9 : 0.01859075947823967
iou for class 10 : 0.0
iou for class 11 : 0.02454635209205733
iou for class 12 : 0.0004988217486282402
iou for class 14 : 0.0008986388264834148
iou for class 15 : 0.40602515236134606
iou for class 18 : 0.1695820290922482
iou for class 19 : 0.0007751937984496124
iou for class 20 : 0.02334784329244163
iou for class 22 : 0.42159137830043436
iou for class 24 : 0.3731604725516428
iou for class 25 : 0.0040729771979236065

Class wise recall :

precision for class 0 : 0.7990999039599759
precision for class 2 : 0.7157660167130919
precision for class 4 : 0.010488676996424315
precision for class 5 : 0.13528734161013695
precision for class 6 : 0.014565632044390498
precision for class 7 : 0.005763688760806916
precision for class 9 : 0.04520527641914925
precision for class 10 : 0.0
precision for class 11 : 0.025586883616941995
precision for class 12 : 0.002715101582248853
precision for class 14 : 0.007428446580729736
precision for class 15 : 0.9603239647109868

precision for class 18 : 0.5215175537938845
precision for class 19 : 0.005154639175257732
precision for class 20 : 0.07139015864479699
precision for class 22 : 0.5782103285395704
precision for class 24 : 0.6870864037742939
precision for class 25 : 0.004333333333333333

Class wise dice coefficient:

precision for class 0 : 0.7988786524179755
precision for class 2 : 0.5530779164873009
precision for class 4 : 0.00016385872398742756
precision for class 5 : 0.05744877438991249
precision for class 6 : 0.011117728806829461
precision for class 7 : 0.0003114779629341224
precision for class 9 : 0.030610196708149337
precision for class 10 : 0.0
precision for class 11 : 0.3764030279300444
precision for class 12 : 0.0006107191744761503
precision for class 14 : 0.001021266370299171
precision for class 15 : 0.41295244413118287
precision for class 18 : 0.20082860880941997
precision for class 19 : 0.0009115770282588879
precision for class 20 : 0.03353119474614802
precision for class 22 : 0.6088315098063687
precision for class 24 : 0.44956120880997097
precision for class 25 : 0.06348635171952913

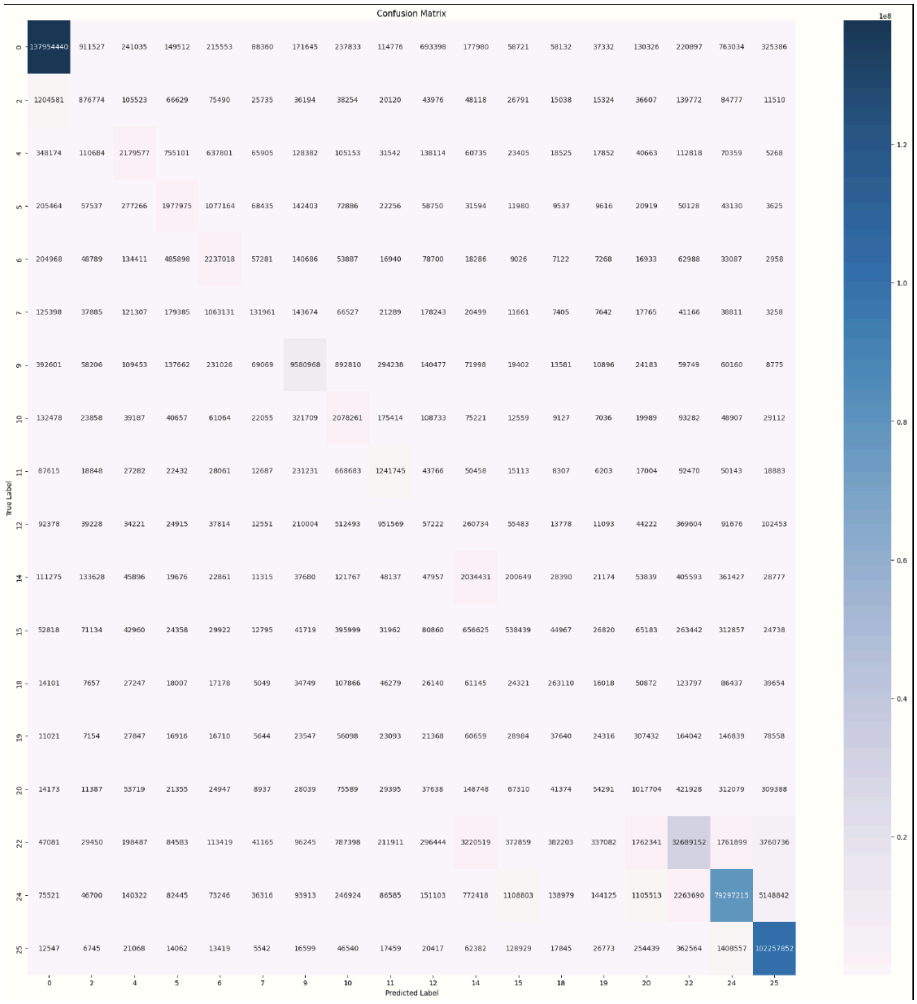
High Performance Classes :

The model achieves high performance on classessuch as sky, vegetation, and building which is evident from their high precision recall and F1score.

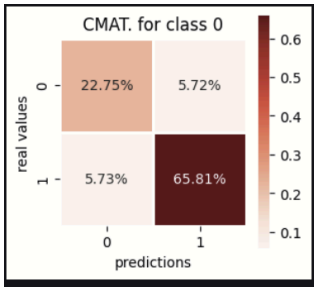
Classes which need improvement :

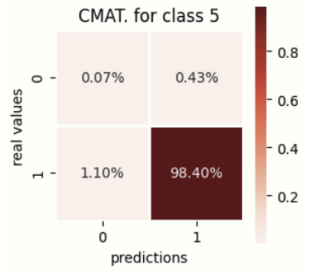
The model exhibits relatively poor performance on classes usch as bicycle, sidewalk, fence, and people. Hence the model could be refined to give better performance on these classes.

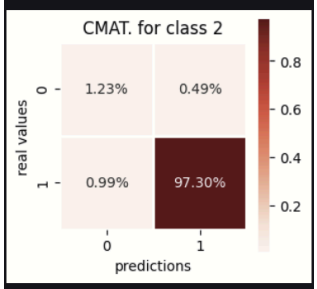
Confusion Matrix :

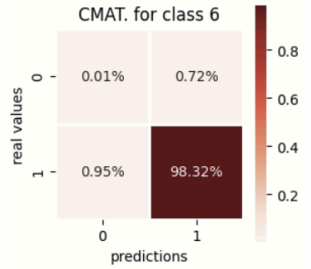


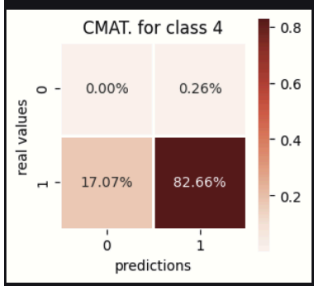
Confusion matrix for each class :

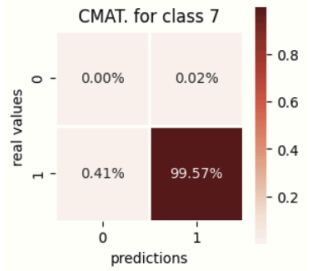

CMAT. for class 0
22.75%
5.72%
5.73%
65.81%
real values
predictions

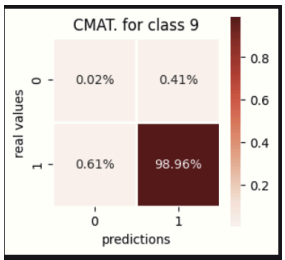

CMAT. for class 5
0.07%
0.43%
1.10%
98.40%
real values
predictions

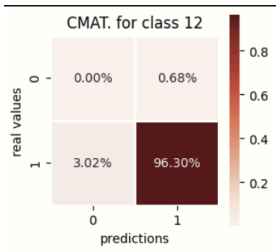

CMAT. for class 2
1.23%
0.49%
0.99%
97.30%
real values
predictions

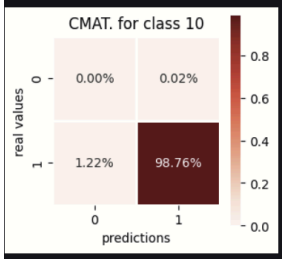

CMAT. for class 6
0.01%
0.72%
0.95%
98.32%
real values
predictions

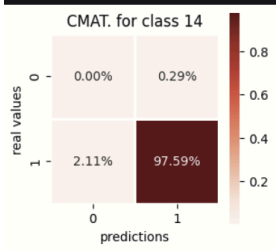

CMAT. for class 4
0.00%
0.26%
17.07%
82.66%
real values
predictions

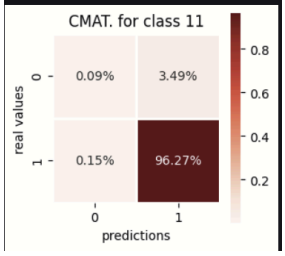

CMAT. for class 7
0.00%
0.02%
0.41%
99.57%
real values
predictions

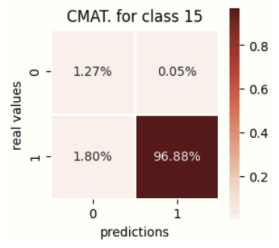

CMAT. for class 9
0.02%
0.41%
0.61%
98.96%
real values
predictions

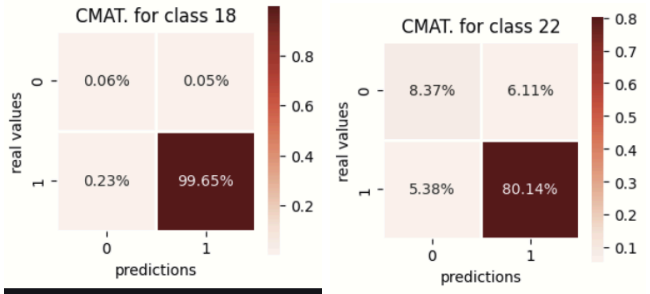

CMAT. for class 12
0.00%
0.68%
3.02%
96.30%
real values
predictions

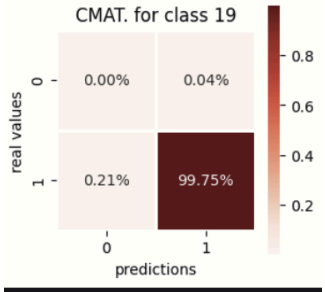

CMAT. for class 10
0.00%
0.02%
1.22%
98.76%
real values
predictions

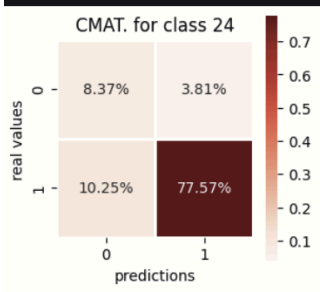

CMAT. for class 14
0.00%
0.29%
2.11%
97.59%
real values
predictions

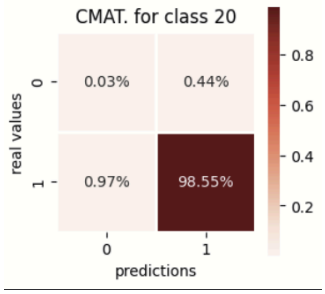

CMAT. for class 11
0.09%
3.49%
0.15%
96.27%
real values
predictions

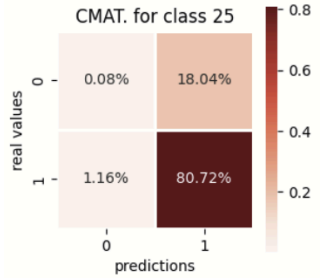

CMAT. for class 15
1.27%
0.05%
1.80%
96.88%
real values
predictions











The model is able to accurately identify classes such as roads, vegetation, sky, which is apparent from their high IoU and precision. Classes such as bicycle and fence suffere due to their small size in images.