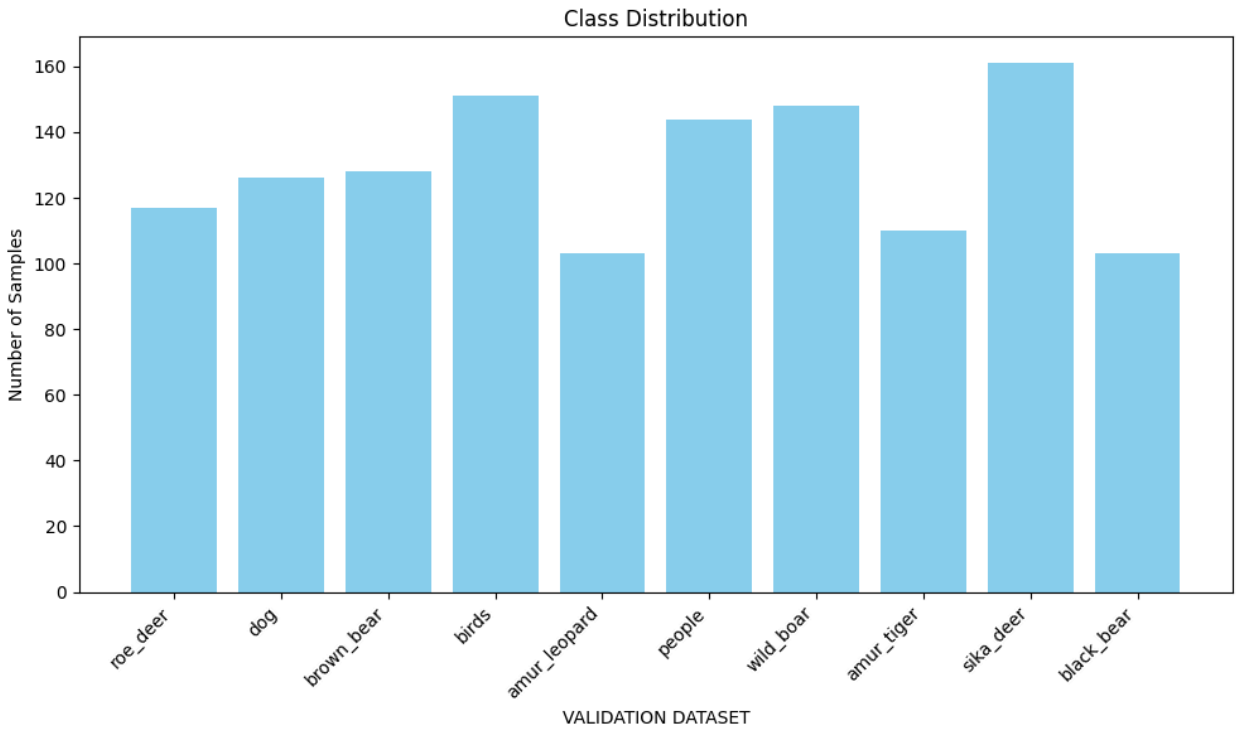
CV ASSIGNMENT 1

Report

Ganeev Singh
2021389

2 1 c) Visualising Data Distribution:



Since I used Stratified Random Split, the training and validation classes have proportional frequencies from different classes

2 2 b)
Trained the custom CNN that i created in part 1 on the data loaders made in 2 1 a)

Graphs for training loss, valuation loss and valuation accuracy vs epochs is graphed on: CNN_train

1. The Training loss function is decreasing at a very high rate and then plateauing at the end, implying that the model is converging
2. The validation loss function is decreasing with time till the 8th epoch, it is plateauing later, Implying that the model is reaching its capacity.
3. The Validation accuracy is increasing at first and plateauing later

2 2 c)
The model is not overfitting
The Model is starting to show subtle signs of overfitting after the 8th epoch.
The validation loss is increasing while the training loss is still decreasing after the 8th epoch, Suggesting that the model might be overfitting.
At the same time the Validation accuracy is not changing after the 8th epoch, which suggests that the model is trained to its capacity, further training will overfit the model

2 2 d)

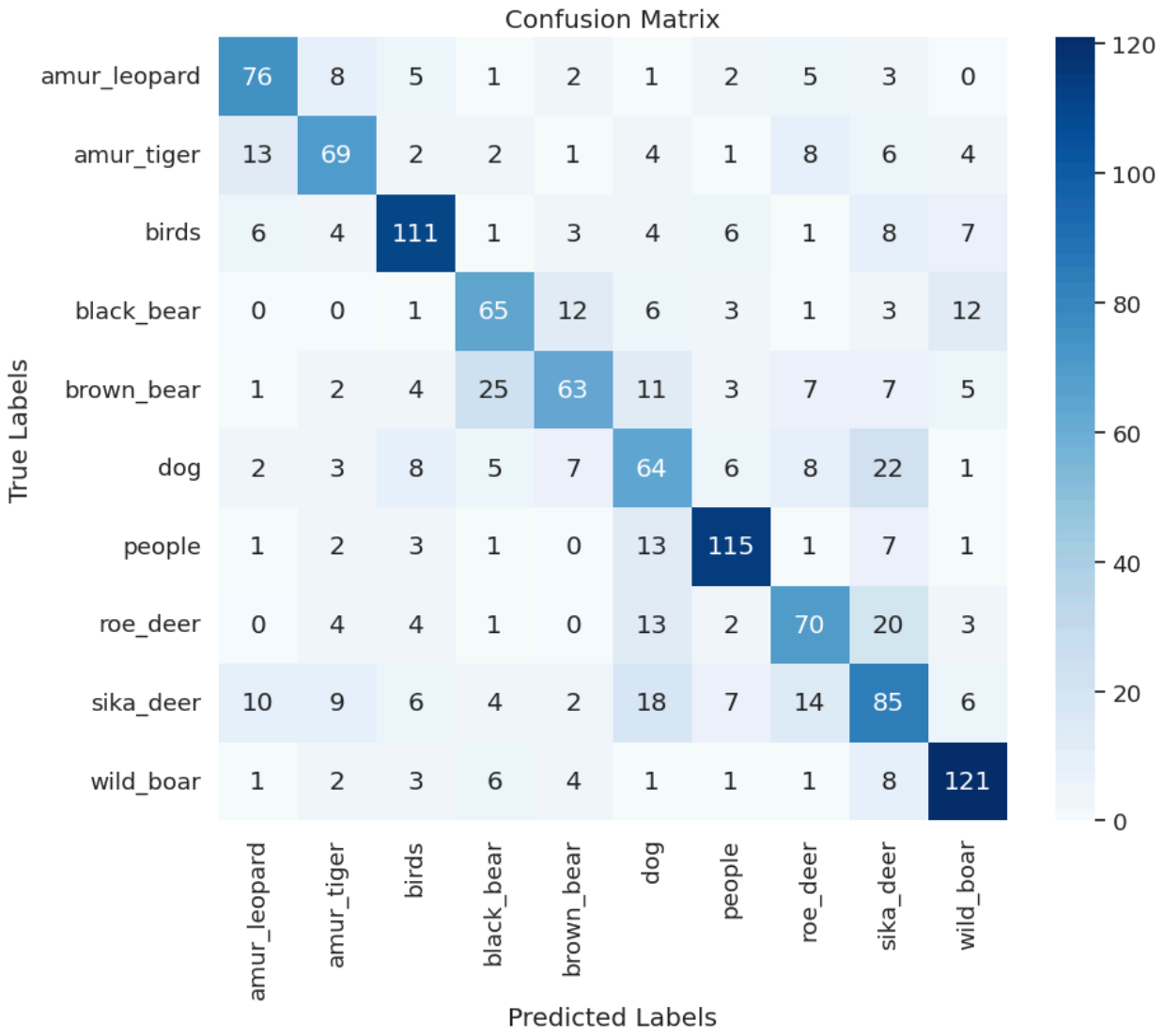
The test accuracy for the custom CNN model after it was trained on the given training dataset is 64.99%. The model is not performing as well as we'd like it to.

The F1 score for the given test is 0.6491. Since the f1 score is close to accuracy, it suggests that the model is almost equally likely to predict a false positive as it is to predict a false negative. Hence the model is homogenous. Again we would like the model to perform better than 0.6491 f1 score.

The confusion matrix for the CNN model

Confusion Matrix:

```
[[ 76   8   5   1   2   1   2   5   3   0]
 [ 13  69   2   2   1   4   1   8   6   4]
 [  6   4 111   1   3   4   6   1   8   7]
 [  0   0   1  65  12   6   3   1   3  12]
 [  1   2   4  25  63  11   3   7   7   5]
 [  2   3   8   5   7  64   6   8  22   1]
 [  1   2   3   1   0  13 115   1   7   1]
 [  0   4   4   1   0  13   2  70  20   3]
 [ 10   9   6   4   2  18   7  14  85   6]
 [  1   2   3   6   4   1   1   1   8 121]]
```



2 3 a)
Trained the pretrained RESNET18 model on the data loaders made in 2 1 a)

Graphs for training loss, valuation loss and valuation accuracy vs epochs is graphed on: [RESNET_train]

1. The Training loss function is decreasing at a very high rate and then plateauing at the end, implying that the model is converging
2. The validation loss function is fluctuating reaching a peak on the 6th epoch
3. The Validation accuracy is Fluctuating and reaches an all time low on 6th epoch

2 3 b)
YES, the model is overfitting

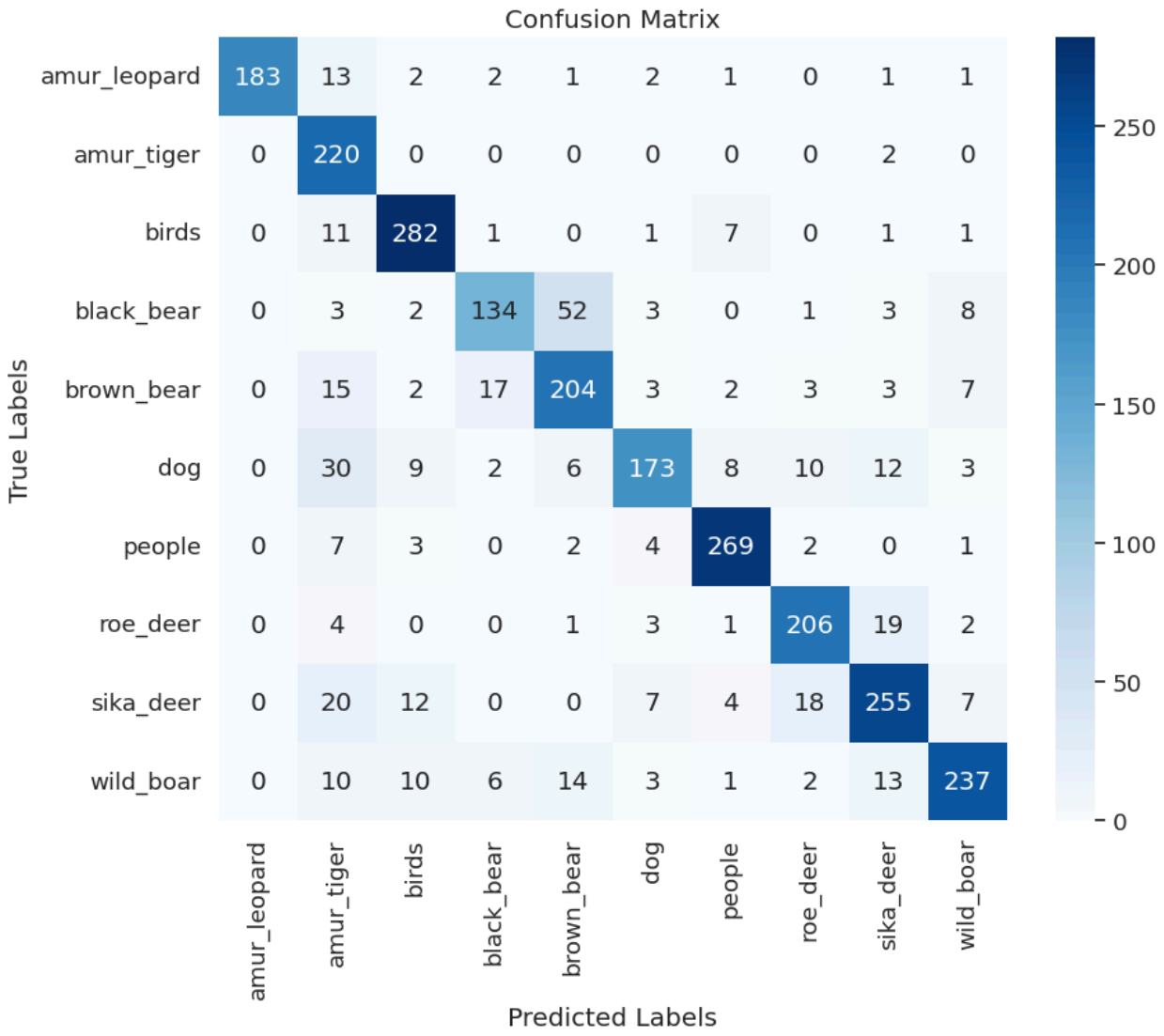
Since the training loss is decreasing till the last epoch but the validation loss is increasing. It shows that the model is overfitting. It is performing better on seen data and worse on unseen data.
It can be seen that the model has started to overfit after the 4th epoch.

2 3 c)
The model has an accuracy of 83.51% on the test data, The model is doing a better job at classification than the custom CNN in the previous part.

The F1 score for the given test is 0.8348. Since the f1 score is close to accuracy, it suggests that the model is almost equally likely to predict a false positive as it is to predict a false negative. Hence the model is homogenous.

The confusion matrix:

```
[[183  13   2   2   1   2   1   0   1   1]
 [  0 220   0   0   0   0   0   0   2   0]
 [  0  11 282   1   0   1   7   0   1   1]
 [  0   3   2 134  52   3   0   1   3   8]
 [  0  15   2  17 204   3   2   3   3   7]
 [  0  30   9   2   6 173   8  10  12   3]
 [  0   7   3   0   2   4 269   2   0   1]
 [  0   4   0   0   1   3   1 206  19   2]
 [  0  20  12   0   0   7   4  18 255   7]
 [  0  10  10   6  14   3   1   2  13 237]]
```

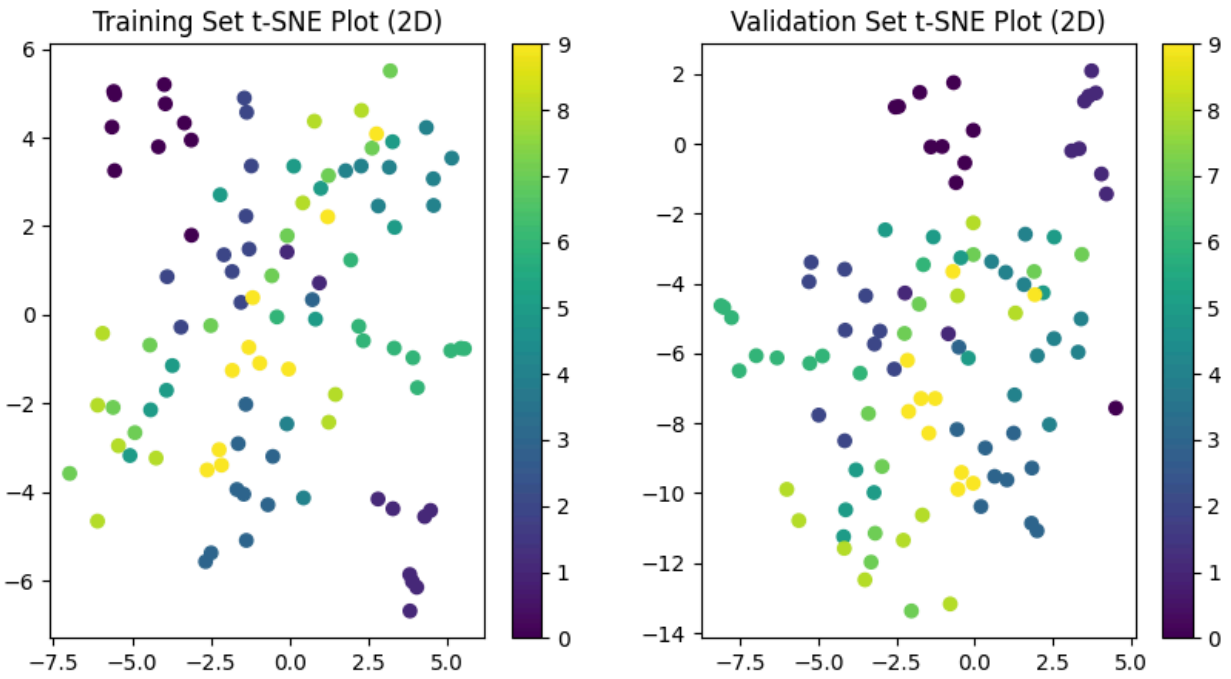


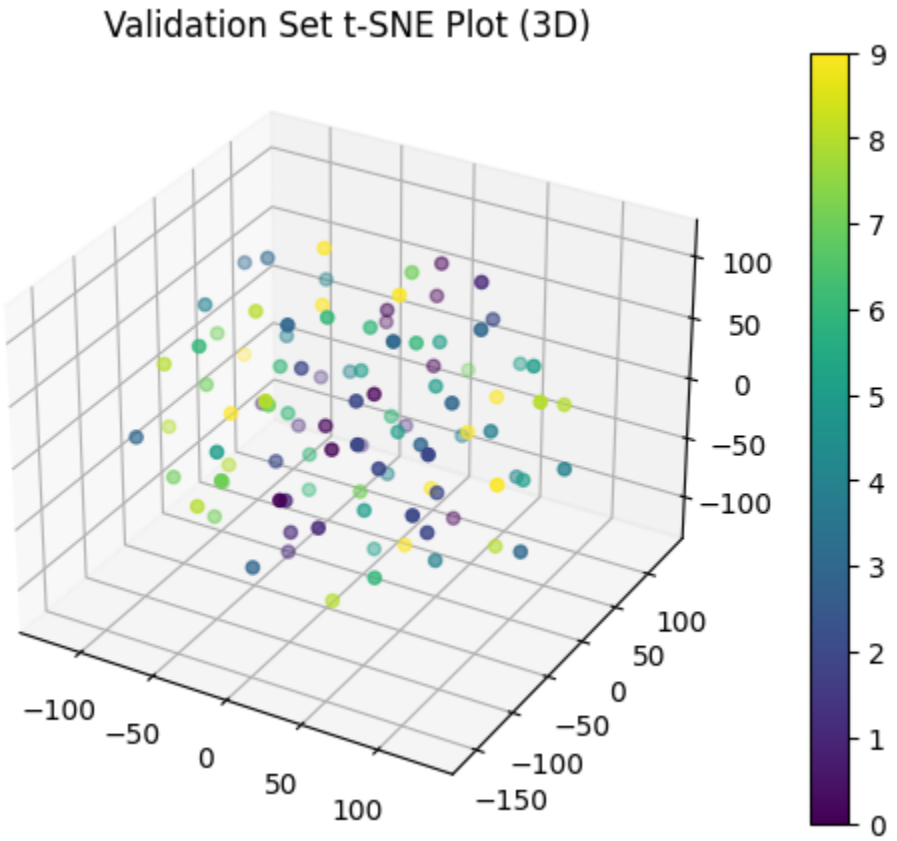
The confusion matrix is homogenous for most values. There is an outlier that catches the eye
The [3][4] block which has a value of 52.
The 3 label is assigned to Black bear and 4 to Brown Bear
The model is misclassifying Black Bears as Brown Bears.

2 3 d)

tSNE plots are basically converting the feature vector of higher dimensions into lower dimensions using Support Vector Machine and grouping algorithms

We would like to view the first layer of the neural network based on the final outcomes, We reduce dimensions with tSNE to view if on 2D, the inputs of similar local area belong to one class





(PLEASE NOTE: I understand the plot should have been better, I made an error rather than loading all training values and running it on the pretrained Resnet18 model, I ran only the toy

dataset i was using to check my code, Considering I don't have a gpu and limited Daily usage on Colab, I could not run the script again)

As we can see inputs of similar color are geographically closer together, we know that the model is classifying about correctly

2 4 a)

I want to increase the dataset by applying filters on the dataset, that could be either increasing brightness, rotate, flip or randomly crop etc.

2 4 c)

Training plots for
Cropped_Augmented: Cropped_Augmented
Rotated_Augmented: Rotated_Augmented
ColorJitter_Augmented: ColourJitter_Augmented

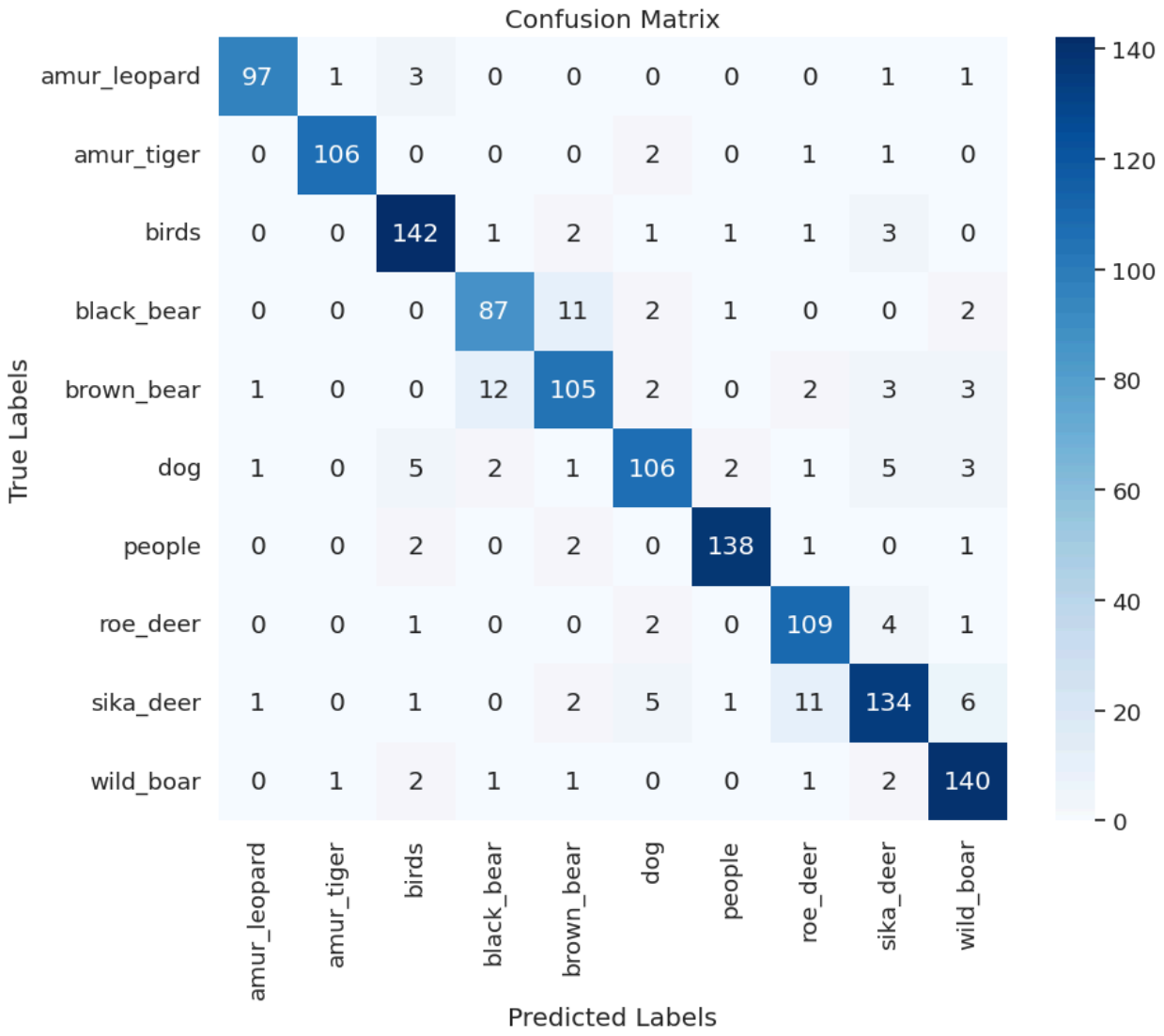
Yes, we have solved the problem of overfitting, even during the last epoch of ColorJitter, the validation loss is going down, the validation accuracy is increasing and training loss is going down.

(d) The model has an accuracy of 90.16% on the test data, The model is doing a better job at classification than the preTrainedResnet18 in the previous part.

The F1 score for the given test is 0.90. Since the f1 score is close to accuracy, it suggests that the model is almost equally likely to predict a false positive as it is to predict a false negative. Hence the model is homogenous.

The confusion matrix:
Confusion Matrix:

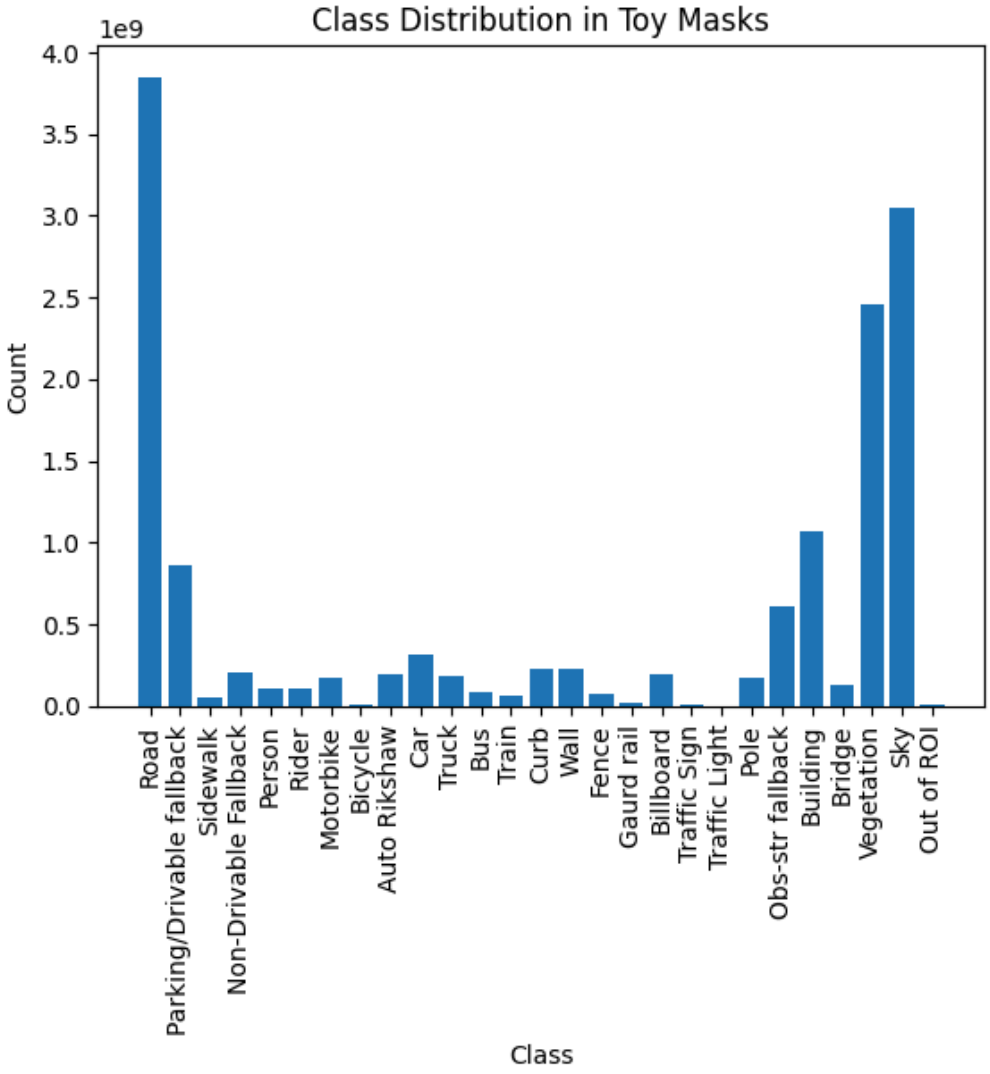
```
[[ 97   1   3   0   0   0   0   0   1   1]
 [  0 106   0   0   0   2   0   1   1   0]
 [  0   0 142   1   2   1   1   1   3   0]
 [  0   0   0  87  11   2   1   0   0   2]
 [  1   0   0  12 105   2   0   2   3   3]
 [  1   0   5   2   1 106   2   1   5   3]
 [  0   0   2   0   2   0 138   1   0   1]
 [  0   0   1   0   0   2   0 109   4   1]
 [  1   0   1   0   2   5   1  11 134   6]
 [  0   1   2   1   1   0   0   1   2 140]]
```



We can see that the model is no longer misclassifying black bear with a brown bear. The model has improved a lot.

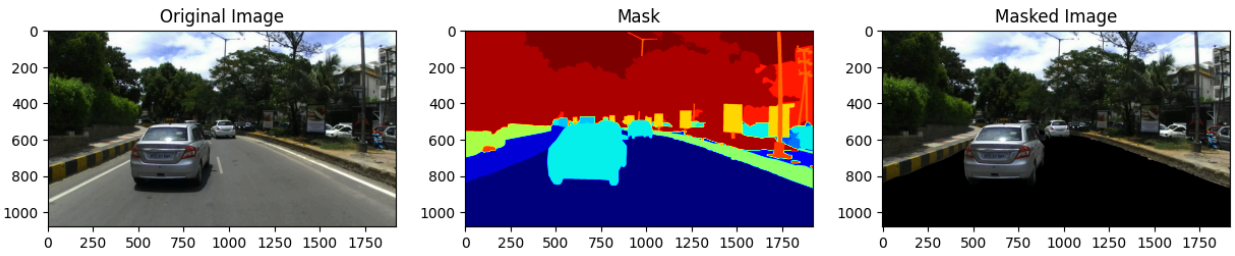
Q3 Report

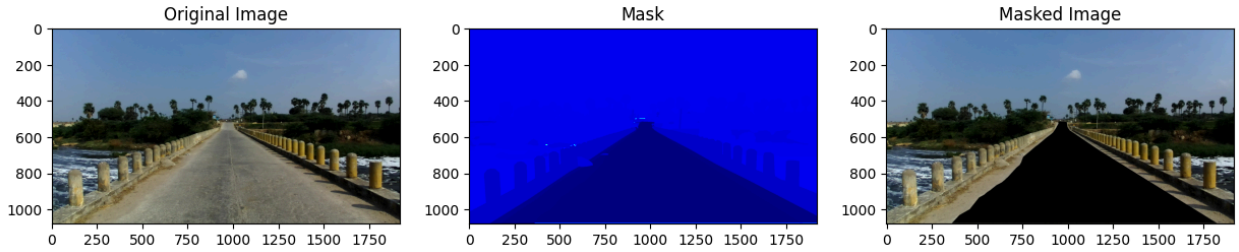
3 1 a)



Performed a class vs pixel count analysis on the given dataset

3 1 b)





Visualized images of Original Image and its corresponding masks

3 2 a)

Predicted the masks for 30 percent of the dataset I created.
I added the masks to my drive so that I can use them later. The mapping for the IDD Dataset and Cityscape dataset are different: I used a dictionary to map between the 2. The saved file is of the IDD dataset classification.

3 2 b)
Finding pixel wise accuracy, dice coefficient, mAP, and IoU for all given classes.
Created a dictionary for each IOU, pixel_wise_accuracy, dice_coefficient, Mean_precision and Mean Recall (see Colab)

Segmentation Model Performance Report
The metrics employed for evaluation include pixel wise accuracy, Dice coefficient, mean average precision (mAP), and Intersection over Union (IoU). Each metric has been calculated at IoU thresholds ranging from 0 to 1, at intervals of 0.1.

So I have 10 IOU values for each class at thresholds 0.1, 0.2, 0.3…
Classes are evaluated based on their mean metric scores across these thresholds

The model's performance varied across classes, demonstrating a diverse range of effectiveness in segmenting different objects. The results highlighted specific areas where the model excelled and others where improvements are necessary.

High-Performance Classes:
Classes such as Sky, Vegetation, and Buildings showcased high precision, recall, and F1 scores across nearly all IoU thresholds, indicating the model's robustness in these categories.

Areas Needing Improvement:
Some classes like Bicycles, Traffic Lights, and Trains exhibited lower precision, recall, and F1 scores, suggesting difficulties in segmentation which could be attributed to similarities with other classes or inherent complexities in their shapes and appearances.

Balanced Performance:

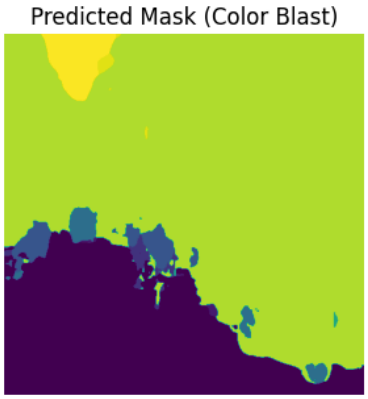
Classes such as Roads, Persons, and Cars demonstrated moderate precision, recall, and F1 scores, signifying satisfactory segmentation but also indicating potential for further refinement.

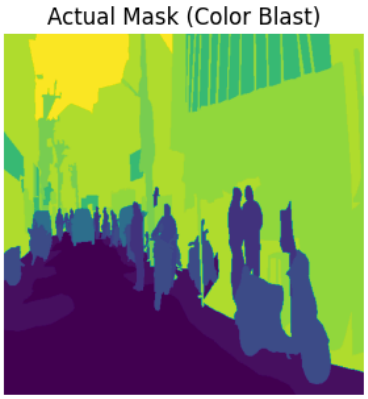
Attention Required:
Classes including Sidewalks, Traffic Signs, and Fences showed lower performance metrics, highlighting the need for model improvements in these areas, perhaps through more nuanced feature extraction or additional training data.

3 2 c)
Visualizing the images with low IOU of each class to identify if the image is being occluded or miss classified in the given environment
I have taken out one example from the visualized images to analyze

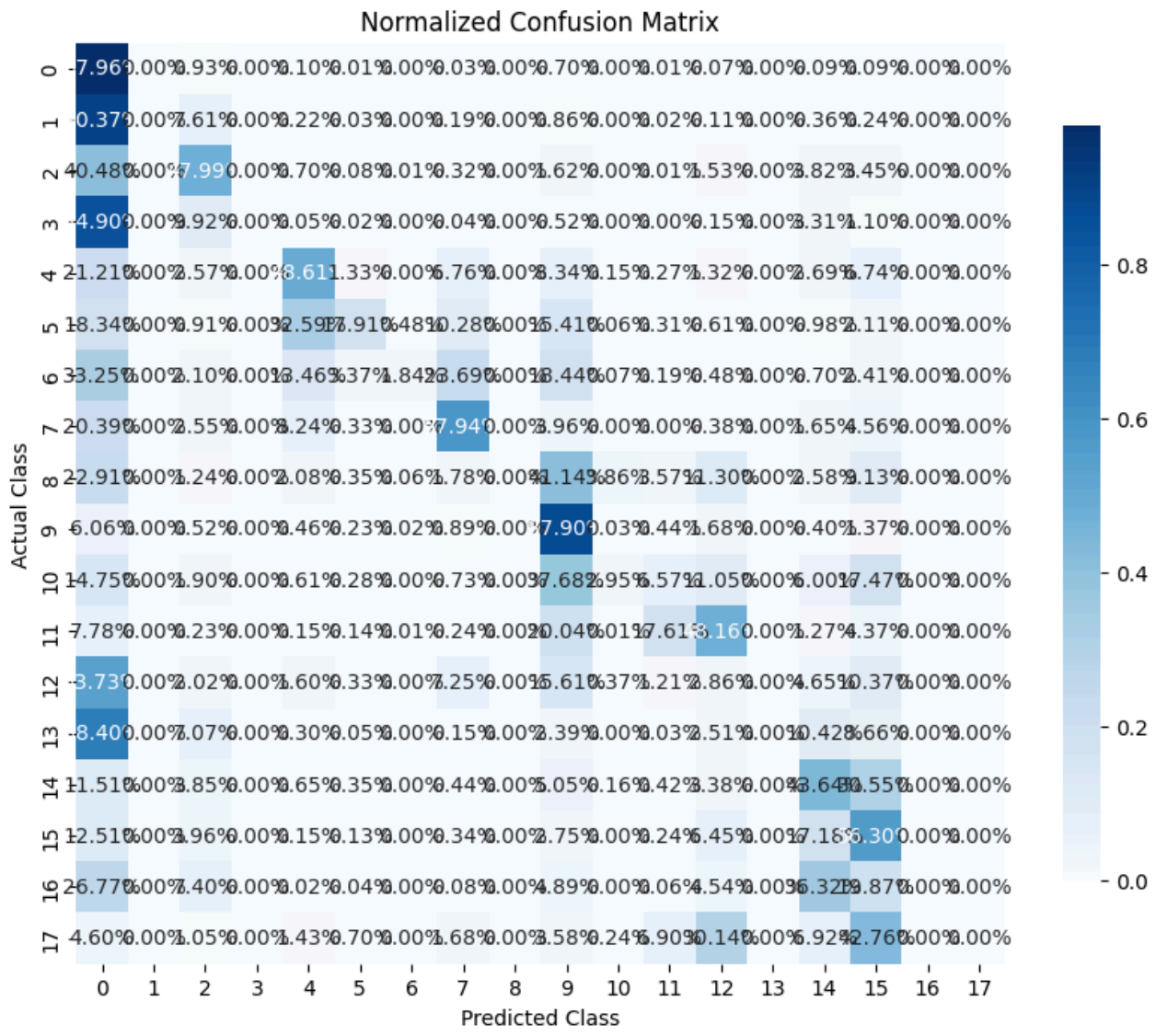
In this, we can see that Cityscape dataset is unable to classify a scooty, as seen in the pic: A possible reason could be that the Cityscape dataset on which the model was trained did not have that many entries of 2 wheelers and was unable to classify it

We can also see that the cityscape model is unable to classify people on the side of the road: A possible reason for this could be that the Cityscape Dataset does not have entries where humans are so close to the road and in between road and building

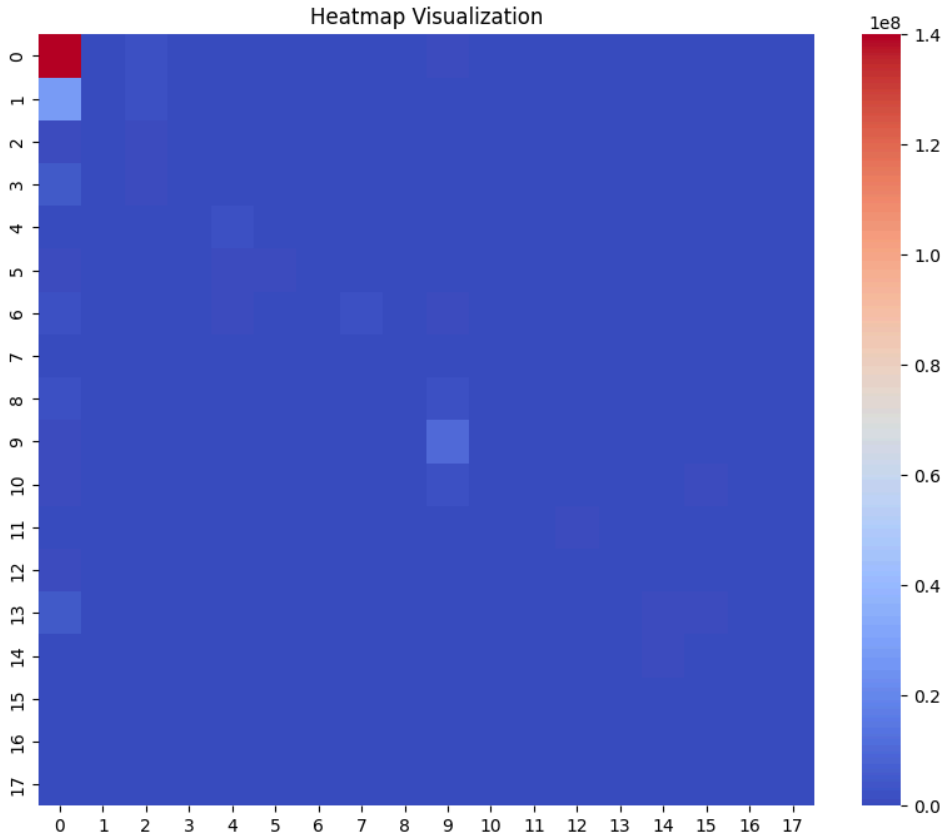




3 3 a



The Confusion matrix tells how the model is able to classify different pixels, an Identity matrix implies that the model is able to identify each pixel perfectly.
We can see that the Model is confused between 0 and 1 labels which correspond to the road and the sidewalk respectively. This highlights the weak points of the dataset. The cityscape dataset that contains mostly urban setting images is unable to segment Indian dataset.
The model is able to classify the roads and cars the best. This should be attributed to the fact that there is a huge dataset present on roads and cars in the cityscape dataset.



3 3 d)

```
Class-wise Performance Metrics:
Class Class 0: Precision: 0.76, Recall: 0.98, F1 Score: 0.86
Class Class 1: Precision: 0.00, Recall: 0.00, F1 Score: 0.00
Class Class 2: Precision: 0.13, Recall: 0.48, F1 Score: 0.20
Class Class 3: Precision: 0.00, Recall: 0.00, F1 Score: 0.00
Class Class 4: Precision: 0.36, Recall: 0.49, F1 Score: 0.41
Class Class 5: Precision: 0.65, Recall: 0.18, F1 Score: 0.28
Class Class 6: Precision: 0.77, Recall: 0.02, F1 Score: 0.04
Class Class 7: Precision: 0.03, Recall: 0.58, F1 Score: 0.05
Class Class 8: Precision: 0.00, Recall: 0.00, F1 Score: 0.00
Class Class 9: Precision: 0.59, Recall: 0.88, F1 Score: 0.70
Class Class 10: Precision: 0.35, Recall: 0.03, F1 Score: 0.05
Class Class 11: Precision: 0.41, Recall: 0.18, F1 Score: 0.25
Class Class 12: Precision: 0.01, Recall: 0.03, F1 Score: 0.02
Class Class 13: Precision: 0.00, Recall: 0.00, F1 Score: 0.00
Class Class 14: Precision: 0.25, Recall: 0.44, F1 Score: 0.32
Class Class 15: Precision: 0.11, Recall: 0.56, F1 Score: 0.18
Class Class 16: Precision: 0.00, Recall: 0.00, F1 Score: 0.00
Class Class 17: Precision: 0.00, Recall: 0.00, F1 Score: 0.00
```