

COMP4211 PA2 Report

Name: Ruiming Min; SID: 20827430; ITSC: rmin

March 17, 2024

Notice:

Since features of the `LATEX` file, most of the images and tables are not shown under the discirbation. You can find them by index. Also, you can click the index to jump to the corresponding figure or table.

5 Datasets and Data Loaders

5.3 Dataloader Implementation

[Q1]

Number of images in COCO dataset: 3557. Number of images in WikiArt dataset: 7492.

[Q2]

Number of images in PACS train dataset: 1641. Number of images in PACS test dataset: 2723.

6 Style Transfer

6.2 Model

[Q3]

Since after encoding, the image is represented as a 3D tensor and its 2D shape is much smaller than the original image. Therefore, in the decoder, we need the upsampling-re-construct the image-its original size.

[Q4]

For each layer, we have the trainable parameters as follows:

- Conv2d: $(k \times k \times c_{in} + 1) \times c_{out}$
- Pooling and Upsampling: 0

Therefore, the total number of trainable parameters is 2332511.

[Q5]

Since the we need encoder-encode the image and mine the features, but if we only use the encoder, we cannot re-construct the image. Therefore, we need the decoder-re-construct the image.

[Q6]

Since in this model, we do not only need the style match our target style, but also need the content-be preserved. Therefore, we need-use the content loss and style loss-balance the style transfer.

6.4 Training the Style Transfer model

[Q7]

After 55 epochs, total loss is 13021.544, content loss is 8052.103, and style loss is 4969.441.

6.5 Inference

[Q8]

From the figure 1, we can see the style transfer examples. In these comparisons, we can see that the generated images represented the content of the content images very well expect the words in them. For the style, the generated images are very similar-the style images in color and painting techniques. However, if the style is very different from the content, the generated images may not be very good (e.g.: picture 1). Moreover, the generated images have some strange textures in some areas, which may be caused by the imbalance of the style data set, i.e., there are too many paintings in the training set.

7 Classification Task

7.2 Analyzing the Dataset

[Q9]

From the figure 2, and the output of the code (table 1), we can see that the training and testing dataset distributions are very similar expect house and guitar. The ratios of training samples of house and guitar are less than the testing samples. And the number of training samples of each class is a half of the testing samples.

Label	Train Count	Test Count
Giraffe	265	425
Dog	262	407
Elephant	256	370
House	251	386
Person	247	413
Horse	210	382
Guitar	150	340

Table 1: Comparison of training and testing dataset distributions.

[Q10]

Since horse and guitar have less training samples, the model may not be able-learn the features of these two classes very well. Therefore, the model may not be able-classify the horse and guitar very well.

7.3 Model Implementation

[Q11]

For each layer, we have the trainable parameters as follows:

- Dense Layer: $(n_{in} + 1) \times n_{out}$
- Pooling: 0

Therefore, the total number of trainable parameters is 2103303.

7.4 Training the Classification model

[Q12]

After 150 epochs, the cross-entropy loss is 0.856, the detailed loss summary is shown in the table 2.

Table 2: Cross-Entropy Loss Summary Every 10 Epochs

Epoch	Cross-Entropy Loss
10	1.505
20	1.409
30	1.306
40	1.256
50	1.214
60	1.164
70	1.132
80	1.091
90	1.047
100	1.023
110	0.968
120	0.955
130	0.911
140	0.888
150	0.856

7.5 Testing Routine

[Q13]

The accuracy of training dataset is 0.674, and the accuracy of testing dataset is 0.252.

The confusion matrix is shown in the table 3 and table 4.

Also, I visualize the confusion matrix in the figure 3a and figure 3b.

		Confusion Matrix for Train						
		Predicted Labels						
True Labels		dog	elephant	giraffe	guitar	horse	house	person
dog		128	110	8	5	4	3	2
elephant		105	131	3	4	3	2	3
giraffe		7	7	165	37	18	20	9
guitar		14	9	11	108	2	1	2
horse		4	6	22	14	144	4	15
house		11	2	6	1	7	217	5
person		12	0	10	6	8	6	199

Table 3: Confusion matrix for the training dataset.

Confusion Matrix for Test

True Labels	Predicted Labels						
	dog	elephant	giraffe	guitar	horse	house	person
dog	66	58	72	56	58	56	41
elephant	67	78	47	48	53	44	33
giraffe	67	45	109	90	35	57	22
guitar	68	40	35	84	53	19	40
horse	66	48	52	51	100	39	26
house	82	9	65	30	34	142	23
person	122	10	64	30	50	30	106

Table 4: Confusion matrix for the testing dataset.

[Q14]

The figure 4 shows the misclassified images. These figures show that the model may not be able to classify those images with complex backgrounds and multiple objects very well. Also, they may be too vague, concise and unconventional.

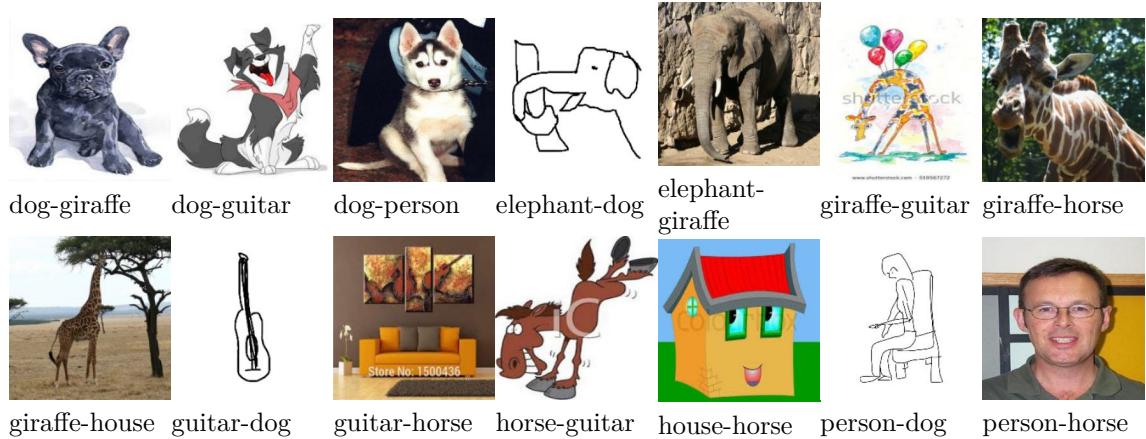


Figure 4: Misclassified Images (the label below is [true label]-[wrong label])

7.6 Data Augmentation by Style Transfer

[Q15]

In figure 5, we can see the example of images with labels and styles.

7.7 Retraining with Augmented Dataset

[Q16]

After 300 epochs, the cross-entropy loss is 0.884.

[Q17]

The accuracy of training dataset is 0.670, and the accuracy of testing dataset is 0.265.

The confusion matrix is shown in the table 5 and table 6.

Also, I visualize the confusion matrix in the figure 6a and figure 6b.

Confusion Matrix for Train							
True Labels	Predicted Labels						
	dog	elephant	giraffe	guitar	horse	house	person
dog	320	61	19	23	16	7	11
elephant	188	170	18	26	14	16	19
giraffe	19	13	318	31	26	20	34
guitar	31	24	44	208	15	11	14
horse	12	11	28	21	281	20	31
house	22	7	12	3	11	368	21
person	20	5	22	14	13	19	351

Table 5: Confusion matrix for the training dataset.

Confusion Matrix for Test							
True Labels	Predicted Labels						
	dog	elephant	giraffe	guitar	horse	house	person
dog	125	17	62	64	54	43	42
elephant	109	64	46	44	46	29	31
giraffe	100	24	136	68	35	38	23
guitar	99	27	41	80	46	26	21
horse	86	37	48	46	105	38	21
house	103	11	50	32	48	120	22
person	127	13	54	38	59	30	92

Table 6: Confusion matrix for the testing dataset.

Comparing the confusion matrix of the original model and the model with augmented dataset, we can see that the model with augmented dataset has a little bit better performance.

[Q18]

This result may be caused by the fact that the augmented dataset has more samples and more styles, which can help the model learn more features and be more robust.



(a) Content Image



(b) Style Image



(c) Generated Image



(d) Content Image



(e) Style Image



(f) Generated Image



(g) Content Image



(h) Style Image



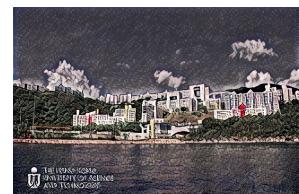
(i) Generated Image



(j) Content Image



(k) Style Image



(l) Generated Image



(m) Content Image

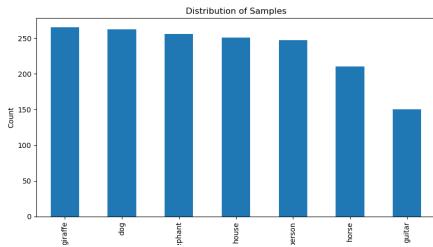


(n) Style Image

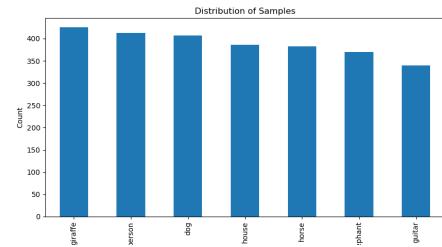


(o) Generated Image

Figure 1: Example of Style Transfer

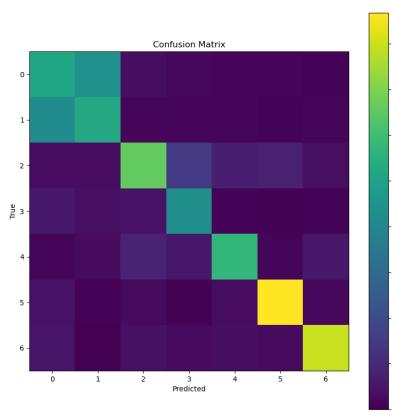


(a) Train Image

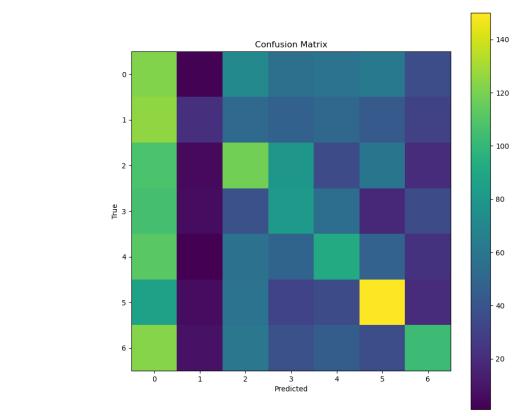


(b) Test Image

Figure 2: Class Distribution



(a) Train Confusion Matrix



(b) Test Confusion Matrix

Figure 3: Confusion Matrix

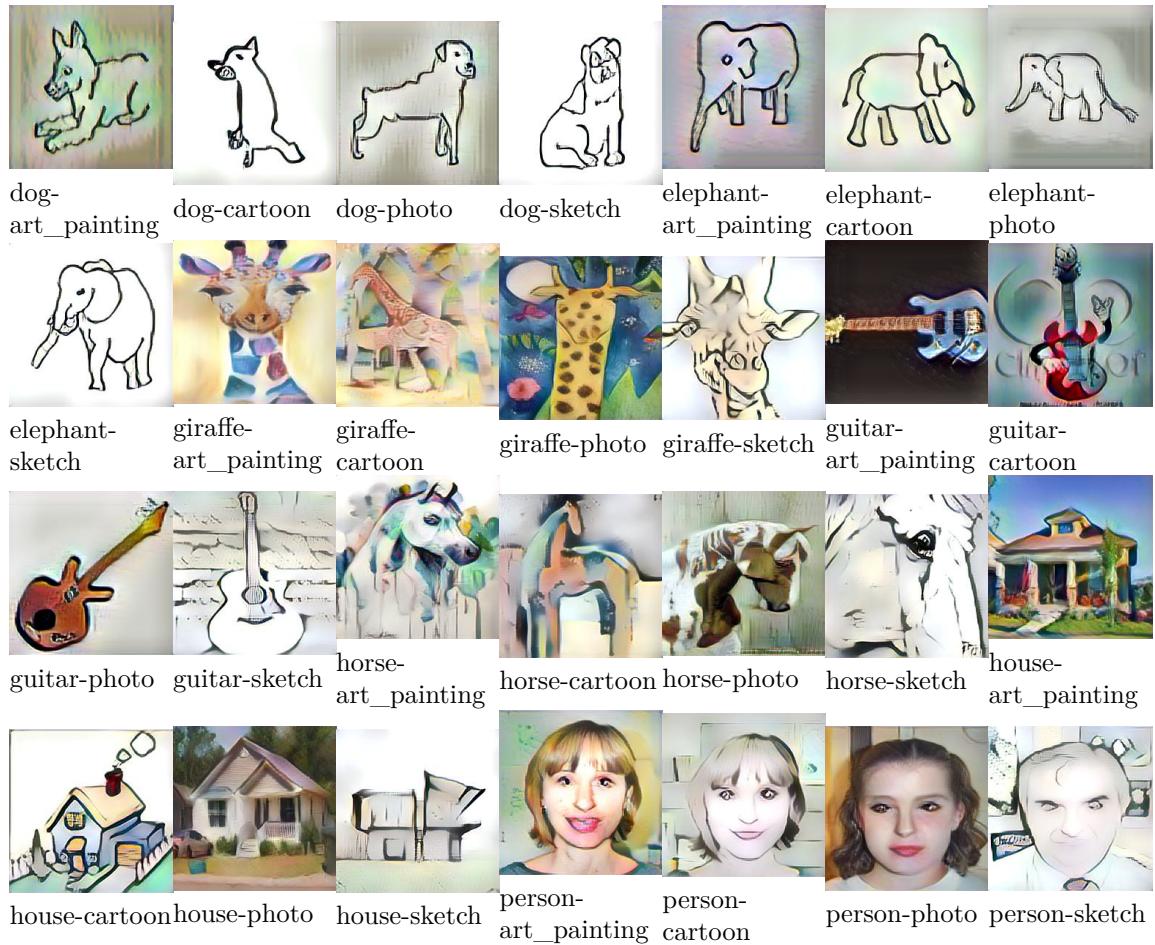


Figure 5: Example of images with labels and styles.

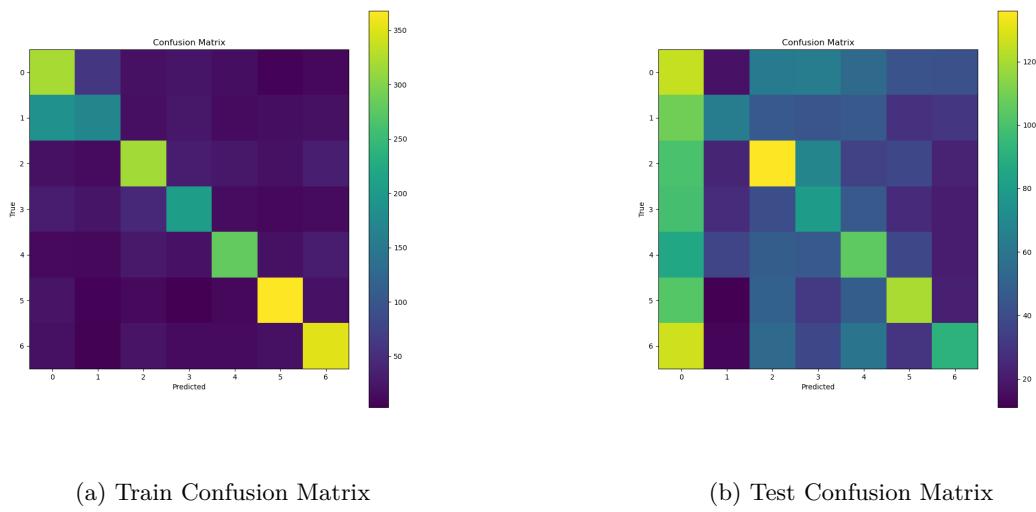


Figure 6: Confusion Matrix