Style Transfer

Computer Vision

Assignment 2

Varun Edachali (2022101029)

February 2025

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

1 Loss Functions

I implemented the loss functions as stated in the document, largely relying on simple tensor operations and the mse-loss function provided by `torch.functional`.

2 Gradient Descent

2.1 Varying Weights of Style and Content Loss

2.1.1 Very High Style Weight, Low Content Weight

$$style\_weight = 1000\ ;\ content\_weight = 0.1$$

I initially attempted this because the paper reported keeping a similar ratio.

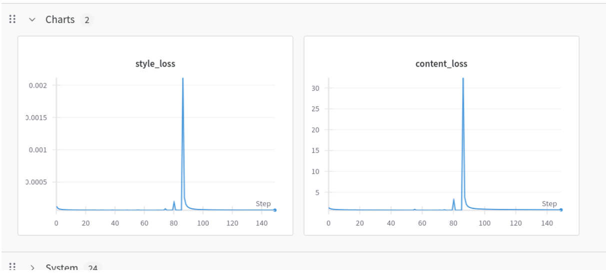

Figure 1: Very High Style, Low Content

The model learns well and seems to escape a local minima at a point. The style and content losses both decrease in a stable manner.

2.1.2 Both High

$$style\_weight = 100\ ;\ content\_weight = 100$$

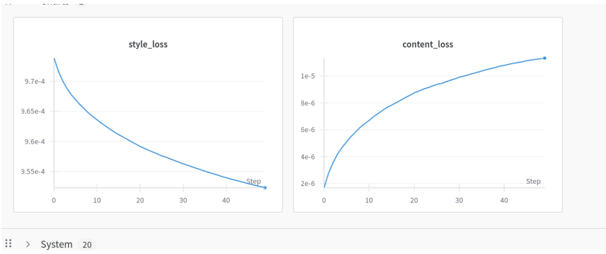

Figure 2: Both High

The content loss begins to increase.

2.1.3 Both Low

$$content\_weight = 0.1 \; ; \; style\_weight = 0.1$$

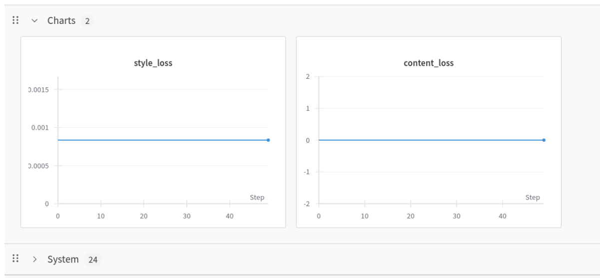

Figure 3: Both Low

The model refuses to learn in this case: compounded with the previous result, we can conclude that it will be difficult to make the model work with both weights being the same / similar.

2.1.4 Style Lower than Content

$$content\_weight = 0.1 \; ; \; style\_weight = 0.01$$

Again, the model refuses to learn. Perhaps the style weight needs to be kept at a significant value.

Figure 4: Style Lower

2.1.5 Style Much Lower than Content

$$content\_weight = 0.1 \; ; \; style\_weight = 0.001$$

On analysing this result along with the preceding ones, we can conclude that we need a high style loss weight, and it should be much higher than that of the content loss weight. This is likely due to the fact that the initialisation of the generated image is that of the content source itself.

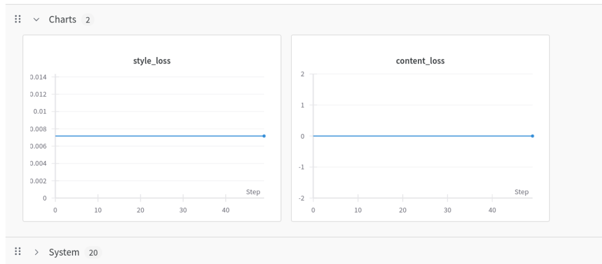

Figure 5: Style Much Lower

2.2 Adam vs L-BFGS

Maintained a content weight of 0.1, style weight of 1000, for 150 steps. Adam was given a learning rate of 0.1.

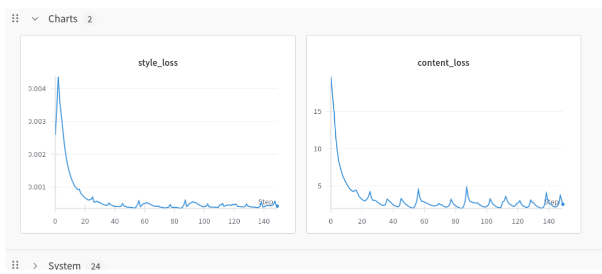

Figure 6: Adam (step weight 1000)

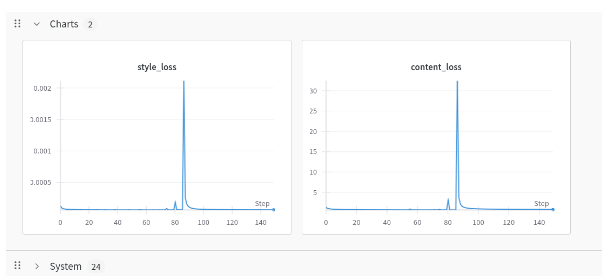

Figure 7: L-BFGS (step weight 1000)

We observe that the learning of Adam is less stable and does not converge to nearly as small a value as L-BFGS over the same number of epochs despite a decently large learning rate.

But, Adam runs much faster and may be able to run for more epochs over the same amount of time.

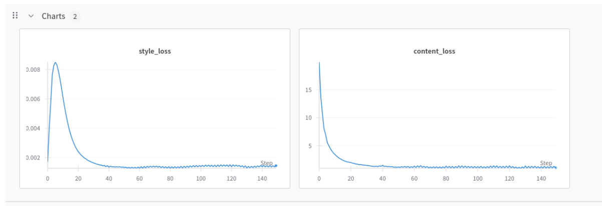

Figure 8: Adam (step weight 100)

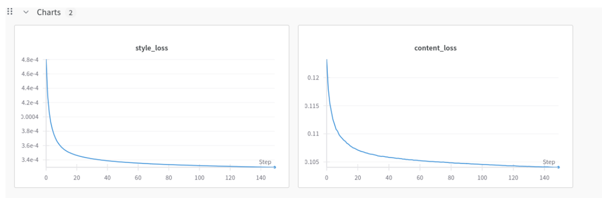

Figure 9: L-BFGS (step weight 100)

Still, the stability of L-BFGS (and the removal of the need to tune a learning rate) suggests it is the better optimizer for the task.

Figure 10: Adam Results (step weight 100)

Figure 11: L-BFGS Results (step weight 100)

However, we should note that although there is no big qualitative difference between the two in the short term (150 epochs), the loss graphs suggest the superiority of L-BFGS for the task at hand.

In addition, this test was carried out assuming the same number of epochs for both optimisers: but Adam works much faster than L-BFGS, so one can argue that the test should be carried out for more epochs for Adam, to match the time taken.

2.3 Observations

In order to study the effects in a structured manner, we vary the content image while maintaing the style source as `bet-you.jpg`. All of these results were trained for 250 steps with a *content_weight* $== 0.1$ and *style_weight* $= 1000$. Some results follow. Some details on the runs for the above are:

Name (20 visualized)	State	Notes	Use	Tag:	Crea	Runtir	Sweep	conten	num_s	optimi:	style_v	conten	style_l
cows.jpgbet-you.jpg	Finished	Add notes	varun-ɛ		4h ago	1h 11m 50	-	0.1	250	lbfgs	1000	0.66445	0.0001192
cat-on-tabl...et-you.jpg	Finished	Add notes	varun-ɛ		6h ago	1h 33m 54	-	0.1	250	lbfgs	1000	0.51087	0.0000892
bear.jpgbet-you.jpg	Finished	Add notes	varun-ɛ		7h ago	1h 14m 41	-	0.1	250	lbfgs	1000	NaN	NaN
town.jpgbet-you.jpg	Finished	Add notes	varun-ɛ		8h ago	1h 33m 17	-	0.1	250	lbfgs	1000	0.69508	0.0001042
white-buil...et-you.jpg	Finished	Add notes	varun-ɛ		10h ago	1h 12m 9s	-	0.1	250	lbfgs	1000	0.58467	0.0000866

Figure 12: details on the run for the observations below

The most fundamental (highly qualitative) observation I made is that images having a *similar contrast and brightness* to the style image tend to lead to decent style transfers. For instance, we can observe the following:

2.3.1 Similar Constrast / Brightness

Figure 13: white building - good results

Figure 14: cows - good results

2.3.2 Somewhat Similar Contrast / Brightness

Here, we observe models that do converge, but quite slowly, and though they seem to give quantiatively decent results, they are not qualitatively appealing.

Figure 15: cats - decent results

Figure 16: town - decent results

2.3.3 Dis-similar Contrast / Brightness

Here we see models that collapse entirely - they fail to appropriately learn the style at all.

Figure 17: bear - bad results

2.4 Miscellaneous Runs

Some more runs are noted below (these also align with the previous hypothesis):

2.4.1 Random Combinations

Figure 18: cat - paula. Decent results.

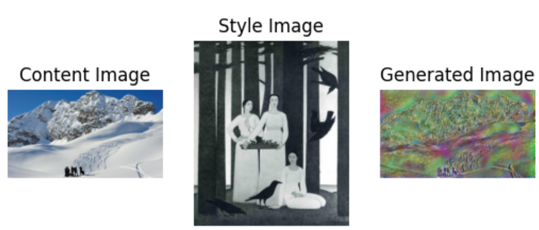

Figure 19: mountains - not detected. Bad results.

2.4.2 My Own Image

Figure 20: my own image