IMAGE COLOURISATION WITH CONDITIONAL GAN

Andreas Lau Hansen (s194235)
Andreas Høst (s194238)
Lukas Wanzeck (s194369)
Malthe Dohm Andersen (s194257)
Yucheng Fu (s194241)

Technical University of Denmark

ABSTRACT

Generative adversarial networks (GANs) have demonstrated their efficiency in colourising black-and-white images. We present two methods to implement this. The first is a conditional GAN (cGAN) using a generator consisting of the encoder-decoder network U-Net and acts as the baseline. The baseline model is used to investigate the difference between using L1 and L2 regularisation in the loss function. Furthermore, we build upon the baseline with the better regularisation term and analyse different network architectures for the generator, those of interest are the backbones VGG19 and Xception. The outputs from the models are compared qualitatively by visual inspection as well as quantitatively through the metrics peak signal-to-noise ratio and colourfulness. Among these models, the model with the Xception backbone and L1 regularisation gives the most promising results for the image colourisation task.

***Index Terms*—** Image colourisation, cGAN, VGG-19, Xception, U-net, PSNR, Colourfulness

1. INTRODUCTION

When looking at old black-and-white images, a part of the appreciation for the photos is lost since they do not accurately convey the visual information due to the lack of colour. However, if these old images were colourised, it would bridge the gap between the past and the present, making the photos more lively. Some of the earliest methods were *hand colouring*, where it was up to an artist to decide which colours fit, *scribble based* methods, where some patches were drawn on the image with some colours and nearby regions are coloured accordingly or *example based* methods, where one image was used as inspiration. Some of the earlier *learning based* methods were feedforward neural networks and later also convolutional neural networks [1].

Some of the problems with previous approaches for image colourisation were that they are reliant on user inputs, such as choosing appropriate colours of each object in the image or achieve dissatisfying results. However, the recent advances in the field of deep learning addresses these problems, and deep learning is now the preferred approach for colourising images.

Image colourisation is an ill-posed problem by nature, as a unique RGB-value does not exist for each greyscale value [2, ch. 8]. Therefore, the goal of colourisation is not to recreate colours accurately, but rather to fool the human viewer into believing the authenticity of the colours. In this project we focus on investigating the backbones Xception and VGG-19 in comparison to a baseline U-Net structure as well as which type of regularisation to use. These intermediate results will be used to find an optimal conditional generative adversarial network (cGAN) architecture that solves the image colourisation problem.

Code available on Github: `https://github.com/AndreasLH/Image-Colourization`

2. DATA

The validation split of the Places365 dataset is used as dataset for this project. The dataset consists of 36500 images of size 256×256 from 365 location categories, varying from categories like *Rainforest* to *Cafeteria*.

The dataset is preprocessed to remove the few black-and-white images that are present. This is accomplished by converting the RGB images to HSV colour space, and if the hue component of the first pixel in an image is 0, it is a black-and-white image and then subsequently removed from the dataset. After preprocessing, there are $33,034$ images, split into $30,000$ for training, $2,509$ for validation, and 525 for testing. The distribution of the 365 classes in the dataset is somewhat even with around 90 images per category, as visualised in the figure below:

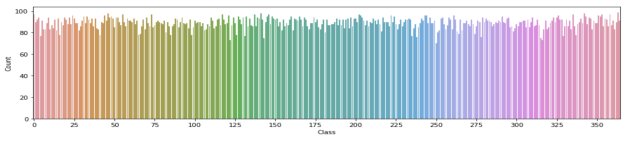

Fig. 1. Distribution of the 365 classes in the dataset.

3. METHODS

3.1. L*a*b* Colour Space

The L*a*b* colour space is an alternative to the RGB colour space to describe coloured images. Whereas the RGB colour space has three colour channels to describe the colour of one pixel, the L*a*b* colour space has only two channels, namely a* and b*. a* describes the red/green values ranging from -127 to 127, in the OpenCV image processing library [3], where movement in the +a* direction represents a shift towards red. Similarly, b* also ranges from -127 to 127 where movement in the +b* direction represents a shift towards yellow. The remaining L* channel is the luminescence channel describing the lightness of the pixel with a range from 0 to 100. Here, 0 is black and 100 is white. So when displaying only the L* channel one sees the black-and-white image.

Fig. 2. Visualisation of the different channels in the L*a*b* colour space. Unedited picture from [4]

The reason the L*a*b* colour space is used is due to reduced computational complexity. In the RGB colour space, one pixel has 256^3 possible values to describe its colour, since each of the three colour channels has 256 possible values. The advantage of using the L*a*b* colour space therefore lies in the fact that it has two colour channels, meaning only 256^2 possible values to predict, which greatly reduces the computational complexity in colourisation.

3.2. GAN loss

A GAN has two key elements: a *Generator* (G), whose job in our case is to colourise greyscale pictures, and the *Discriminator* (D), whose job is to *discriminate* between the generated (colourised) and the real (original colour) images, i.e. determine if they are real or fake. The fact that the generator and discriminator have opposite goals enforces a competition between the two. This is reflected in the loss function for the GAN:

$$\begin{aligned}\mathcal{L}_{cGAN}(G,D) = \min_G \max_D \mathbb{E}_{x,y}[\log D(x,y)] \\ + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z)))]\end{aligned} \tag{1}$$

Here, x is the greyscale image (L channel), which is the same for the real image and the colourised image. y is the real a*b* channels. The colourised a*b* channels $G(x,z)$ is generated by feeding x along with some input noise z to the generator. The colourised a*b* channels $G(x,z)$ are then concatenated with x to a full image. The discriminator is also fed real images, in which x is concatenated with y

As seen in the loss function, the terms with G are minimised, while the terms with D are maximised. Therefore, one cannot interpret the loss in the traditional sense, as it will fluctuate up and down. This means that the exact value of this loss term is not important, what is important is that both the generator and discriminator improve over time.

Additionally, to improve the performance of the generator, a regression term is added:

$$\mathcal{L}_{L^p}(G) = \mathbb{E}_{x,y,z}\left[\|y - G(x,z)\|_p\right] \tag{2}$$

Here, p denotes the p-norm between the generated and the real image. The final loss function can be written as:

$$\begin{aligned}\mathcal{L}(G,D) = \min_G \max_D \mathbb{E}_{x,y}[\log D(x,y)] \\ + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z)))] + \lambda L_{L^p}(G)\end{aligned} \tag{3}$$

where λ is a hyperparameter which weighs the importance of the regression term. The way in which the weights of the generator and discriminator are updated is visualised in the figure below:

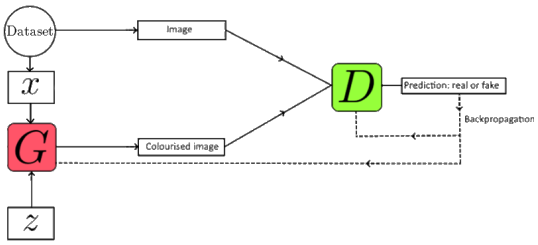

Fig. 3. An overview of the cGAN structure. The greyscale image x is input into G, and noise z is introduced in the form of dropout layers in G. The colourised image is then input into D, and D is also fed a real image from the training set. In both cases, D classifies the images as either real or fake. Based on this, the weights of both G and D are updated accordingly.

3.3. Models

We used a U-net structure for all four of our models' generators, see Figure 4. The generators were based on the U-net architecture, meaning that they have a contracting, downsampling path or encoder followed by an expanding, up-sampling path or decoder. [5]. The part between the encoder and decoder is referred to as the bottleneck. Another essential feature of the U-net architecture is the feature concatenation from each block in the downsampling path to the block in the upsampling path with the same dimensions, a feature called *long skip connections*. This means that there needs to be

as many upsampling blocks as downsampling blocks. This aids the upsampling blocks in reconstructing a more precise image.

The output of the U-net is then sent to the discriminator, which was chosen to be a patch discriminator. Instead of classifying the whole image as real/fake, the patch discriminator only discriminates on a patch scale by sampling $N \times N$ patches across the image, like convolution. The discriminator then determines for each patch if the image is real of fake, and the result is averaged across all patches. The patch discriminator implemented in this project is the same as the one proposed in the "Image-to-Image translation with conditional adversarial networks" paper [6].

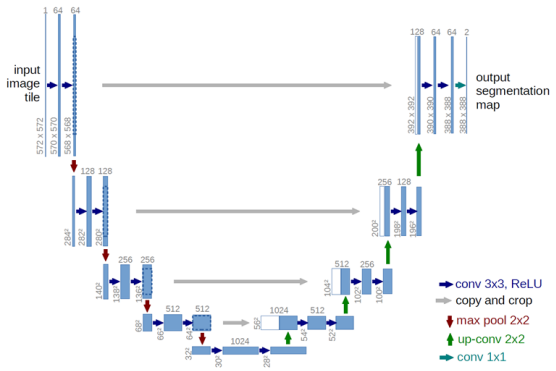

Fig. 4. Unet model structure, note that the dimensions are different from ours. From [5]

The first model was a simple U-net generator with 8 downsampling blocks each containing one convolutional layer of kernel size 4, stride 2 and padding 1, and using transposed convolution layers to upsample. The implementation was taken from this tutorial [7]. The model weights are initialised from a Gaussian distribution with mean $\mu = 0$ and $\sigma = 0.02$ [6]. Since the resulting network was relatively straight forward, we designated it to be our baseline model. The regression term of the model was set as L1-loss. The output of the last layer of the encoder, which is the place where the input has the lowest resolution and the highest amount of channels, is $1 \times 1 \times 1024$, where 1×1 is the resolution and 1024 is the amount of channels.

The second model is identical to the baseline, except the regression term is L2-loss instead (MSE).

For the third model an Xception CNN network, that was pretrained for image classification on the ImageNet dataset, was used as the "backbone" in our U-net structure. This means that we swapped the encoder of the generator with the pretrained network. As the decoder needs to match the encoder, we used `DynamicUnet` from `fastai` to create a fitting decoder. In this process the final pooling layer and fully connected layers are cut off, which only leaves the convolutional layers for feature extraction. The Xception network is characterised by its depthwise separable convolution layers and short skip connections which enables it to achieve high classification accuracy with relatively low size and training time [8]. L1-loss was used for the regression term. The output of the encoder has the dimensionality $8 \times 8 \times 2048$ where 8×8 is the resolution and 2048 is the amount of channels.

For the fourth model a pretrained VGG19 CNN was used as the backbone, while the decoder was constructed using `DynamicUnet` from `fastai`, similarly to how Xception was implemented in model 3. VGG19 is characterised by having many convolutional layers, with small 3×3 convolution filters. L1-loss was used as the regression term. The output of the encoder has the dimensionality $8 \times 8 \times 512$ where 8×8 is the resolution and 512 is the amount of channels.

3.4. Training the models

The models were trained on the DTU HPC cluster on an NVIDIA V100 GPU. All four models were trained with adversarial loss for 100 epochs with a batch size of 16. The binary cross entropy loss function was used for the adversarial loss, and the hyperparameter for the regression term was set to $\lambda = 100$ (see here: for why we chose this [7]). The gradients are updated using the Adam optimiser with learning rate $\eta = 2 \cdot 10^{-4}$ and decay rates $\beta_1 = 0.5$ and $\beta_2 = 0.999$.

Additionally, the two pretrained backbones were finetuned to the colourisation task by training on the entire training set for 20 epochs with L1 loss. The Adam optimiser is used with learning rate $\eta = 10^{-4}$.

The adversarial training for the baseline models and the finetuning of pretrained backbones took around 24 hours. The adversarial training for the two models with VGG19 and Xception as backbone took 3 and 4 days, respectively.

3.5. Evaluation metrics

There exists many qualitative metrics used for colour evaluation, but there is no optimal metric that measures colour plausibility or correlates with how humans perceive naturalness and detail [1]. We will use three different metrics to evaluate the colourised images.

The peak signal-to-noise-ratio (PSNR) is measured in decibels and evaluates the per pixel, pixel-wise difference between a colourised image and the ground truth. A greater PSNR indicates more similar images.

$$\text{PSNR} = \begin{cases} 100, & \text{if MSE} = 0 \\ 20 \cdot \log_{10}\left(\frac{255}{\sqrt{\text{MSE}}}\right), & \text{else} \end{cases} \tag{4}$$

where MSE is the mean squared error of values across the

RGB channels between the colourised image and ground truth.

The colourfulness metric (CFN) is a metric for the overall colourfulness of an image. It is a linear combination of the mean and standard deviation of the red-green (rg) and yellow-blue (yb) colour spaces. A greater colourfulness score indicates more colourful images [9].

$$\text{Colourfulness} = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2} + 0.3 \cdot \sqrt{\mu_{rg}^2 + \mu_{yb}^2} \quad (5)$$

where

$$rg = R - G, \quad yb = \frac{R+G}{2} - B$$

To accurately evaluate the level of colourfulness of the colourised images as compared to the real images, the absolute difference in colourfulness is computed for each of the $N = 525$ test images, and the mean is presented in the results, i.e.:

$$\text{Abs. diff. CFN} = \frac{1}{N}\sum_{n=1}^{N} |(\text{CFN (G)})_n - (\text{CFN (ref.)})_n|$$

4. RESULTS

4.1. Colourised images

Figure 5 shows five examples of images colourised by each of the models, with the ground truth images on the bottom row. The images are cherry picked by the authors to showcase how the models fare on different locations with varying levels of detail. Additionally, figure 6 shows some examples of failure cases for each of the models.

Fig. 5. Cherry picked colourised images. L1 and L2 are different regression terms for the baseline. 3rd and 4th row are the Xception and VGG19 models, respectively. Bottom row is the ground truth images.

Fig. 6. Cherry picked examples of failure cases, e.g. discolouring, colour overspill, colour artifacts, and change of scenery style.

4.2. Quantitative results

Table 1 shows the results from the metrics mentioned in section 3.5 for each of the models. Ground truth is compared to give a reference to the colourfulness. Theses scores are computed on the test set consisting of 525 images.
The baseline with L1 loss achieves the best PSNR score, while Xception achieves the best score in both colourfulness and absolute difference in colourfulness.

5. DISCUSSION

Looking at Figure 5, it can be seen how the different models perform in the image colourisation task. The picture with the train, in particular, showcases the ill-posed nature of the problem, as the trains all have different colours but most look plausible regardless. If the ground truth had not been shown alongside it would not have been obvious which colour the train is supposed to be. As a matter of fact, for older black-and-white photos one typically does not have access to the ground truth, so a result like the train would be perfectly acceptable.

It is clear that the L1 baseline model is very conservative with the saturation leaving most of the images grey-brown. See especially image no. 4 in Figure 5. This is likely due to the L1 and L2 regression terms acting as regularisation terms

	PSNR	Colourfulness	Abs. diff. colourfulness
Baseline L1	**30.33**, CI: $[30.24, 30.43]$	30.09, CI: $[29.05, 31.15]$	15.59, CI: $[14.32, 16.85]$
Baseline L2	29.99, CI: $[29.90, 30.09]$	37.02, CI: $[35.52, 38.52]$	17.40, CI: $[16.11, 18.68]$
Xception	30.29, CI: $[30.21, 30.39]$	**39.44**, CI: $[37.96, 40.94]$	**13.52**, CI: $[12.47, 14.56]$
VGG19	30.17, CI: $[30.09, 30.26]$	22.87, CI: $[22.49, 23.26]$	19.26, CI: $[17.81, 20.72]$
Ground Truth	100	39.01	0

Table 1: Mean PSNR, mean colourfulness and mean absolute difference in colourfulness for the different backbones for $N = 525$ test images. Best result listed in bold. The confidence intervals are on a 95% confidence level.

in the loss function. Since L1-regularisation tends to encourage zero coefficients [10] and since zero in the a*b* colour space is the absence of colour, i.e. greyscale, it is more likely that images with L1 as the regression term would produce grey colours. On the other hand, the L2 model is overconfident and uses colours that are too vivid to look realistic. Additionally, it tends to over-spill a lot on the colours which also reduces the believability. It is especially evident on the image of the white church, which is coloured blue as the sky behind it. As the baseline L2-loss model is the only model that uses L2-loss as the regression term, it explains why it appears to be the only model that suffers from choosing colours that are too vivid.

The Xception backbone is the only model that manages to somewhat accurately colour the people on image no. 5 (Fig 5). This is impressive, since a group of people contain high level of detail. Aside from colouring people, Xception also seems to achieve both the grey tones and realistic look of the baseline L1 model as well as the bright colours seen in the baseline L2 model. While it still has some issues with artefacts and colour over spill (which can be seen in the 1st and 4th picture from the right in Figure 5), they are not as severe as with the baseline L2 model in the examples in Figure 5.

The obvious exception of good colourisation is, however, the model trained with the VGG19 backbone, which is only able to colour in matte green and grey colours. This is not just a plotting issue, as the colourised images have the correct number of colour channels, as well as proper L*a*b* and RGB values. The most plausible reason for VGG19's failure is an implementation error with `DynamicUnet`. Our hypothesis is that the characteristic long skip connections of the U-net structure were not made correctly for VGG19 because of the one-block structure in the code, as opposed to Xception, which has multiple blocks in the code. During training, it was observed that the discriminator improved much faster than the generator, despite the fact that the pretrained generator had been finetuned, as seen in the top left of figure 7. It shows that the loss for the discriminator D of the VGG19 network goes to 0, while the loss for the generator G steadily increases.

It is worth noting that the dimensionality differs between the models in the bottleneck, due to the differences in architecture. Both baselines with L1 and L2 loss has the dimensions $1 \times 1 \times 1024$ in the bottleneck, while VGG19 has the dimensions $8 \times 8 \times 512$ and Xception has the dimensions $8 \times 8 \times 2048$. Larger dimensionality in the botteneck means more information may be preserved when upscaling to produce the a*b* channels, and could also contribute to why Xception generates better looking images.

Some parts of the images are colourised way better than others. This could be explained by the heavily data driven approach taken in this project, as the data set is primarily comprised of simple scenes with outdoor scenery, which explains why the models are good at landscapes (e.g. the mountain range in figure 5) and other simple scenes with less intricate details. Figure 6 showcases how the models perform poorly on various images. For example, even Xception performs badly on colourising the people in the second picture, with a lot of miscolouring on the outfits. Poor performance on people and complicated indoor scenes makes sense given how few people there are in the data set. Therefore, skin is a pain point where the results typically end up looking zombie-like whenever it is colourised in grey. The worst behaviour is that sometimes weird colour artefacts show up on the colourised images, which especially the L2-model had a tendency of doing, as seen in e.g. picture row 2 col. 4 of Figure 6. Exactly what causes these artefacts is unknown to us but it could, highly unlikely, stem from various watermarks present on the training photos. See for example the image of the flowers; it contains a watermark and has been colourised really poorly. However, this is most likely due to the complexity of the object and not the watermark.

In table 1, one can observe that Xception had the highest score in colourfulness, as well as the lowest absolute difference in colourfulness, and this also corresponds to what can be observed in figure 5. However, the 95% confidence intervals for Xception in these two metrics overlap with the corresponding confidence intervals for L2 and L1 respectively, so this difference is not significant on a 5% significance level with $N = 525$ test images. The baseline with L2 loss has the second highest colourfulness score, much more colourful than L1, which is reflected in the more matte colours of L1, as seen in figure 5. Even though L1 produces more matte colours, it outperforms L2 in the mean absolute difference in colourful-

ness, meaning that L1 deviates less in colourfulness from the true images compared to L2. However, keep in mind that a more passive strategy is rewarded since the test set generally consist of fairly colourless images.

It can also be observed that the PSNR scores are generally close to each other, but this does not reflect what can be observed from the colourised images in section 4.1, especially considering the performance of VGG19, which outperforms L2 in PSNR score. According to Wang et al. [11], PSNR is appealing because it is simple to calculate, but it is not very well matched to perceived visual quality. PSNR is therefore not a very suitable metric to evaluate the colourised images. Furthermore, the 95% confidence intervals for the PSNR scores almost all overlap, which prevents us from discerning a difference between the scores. The baseline L2 loss is the only one which falls below the rest with non-overlapping confidence intervals.

As Xception performs best in the absolute difference metric, as well as producing the most natural looking colours as seen in figure 5, this metric seems to be the one which corresponds best with our qualitative analysis of the colourised images.

While the model with the Xception backbone performed well, it clearly has space for improvement. To further improve the model, we hypothesise that including more conditional data might help improve all models. In particular, we had considered introducing the true label of each picture (e.g. *ocean*) as conditional information alongside the greyscale image. However, we did not manage to do it in the span of this project and as such propose it as future work.

Another area that could be improved is the selection of the hyper-parameters, in particular λ, which determines the weight of the regression loss in comparison to the adversarial loss. Due to computing time limitations we did not get to optimise hyper-parameters, and instead went with the default values from the tutorial [7]. It is possible that a better choice of parameters could improve the models.

6. CONCLUSION

In this project, four different GANs have been trained for the image colourisation task. The model using Xception as the backbone for the generator and L1-loss for the regression term created the most believable colourisation results. The L1 baseline seems to produce more natural images than L2 baseline, as the L2 baseline is more colourful, but often has colour spillover which is not desired. The model using VGG19 as backbone performs poorly and only manages to colour images with the colour green.

The Xception models also achieves the best scores in colourfulness and absolute difference in colourfulness. Although compared to the second best models which are the L2 and L1 baselines respectively, this difference is not significant on a 5% significance level with $N = 525$ test images. The L1 baseline achieves the highest PSNR score, but all models have scores that are close to each other even though the coloured images are vastly different. Based on this and the qualitive evaluation, the absolute difference in colourfulness seems to be a better metric for colourisation than PSNR and colourfulness alone.

Overall, the Xception models performs by far the best, especially in the qualitative analysis. Our hypothesis is that the VGG-19 model possibly performs poorly due to the fact that it has no long skip connections.

7. REFERENCES

[1] Ivana Žeger, Sonja Grgic, Josip Vuković, and Gordan Šišul, "Grayscale image colorization methods: Overview and evaluation," *IEEE Access*, vol. 9, pp. 113326–113346, 2021.

[2] Rasmus Reinhold Paulsen and Thomas B. Moeslund, *Introduction to Medical Image Analysis*, Springer, 2020.

[3] OpenCV, "Opencv color conversion," https://docs.opencv.org/3.4.3/de/d25/imgproc_color_conversions.html.

[4] fineartamerica, "unedited parrot picture," https://images.fineartamerica.com/images-medium-large-5/1990s-macaw-parrot-jungle-miami-florida-animal-images.jpg.

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015.

[6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, vol. abs/1611.07004, 2016.

[7] Moein Shariatnia, "Colorizing black white images with u-net and conditional gan tutorial," https://towardsdatascience.com/colorizing-black-white-images-with-u-net-and-conditional-gan-a-tutorial-81b2df111cd8.

[8] François Chollet, "Xception: Deep learning with depthwise separable convolutions," *CoRR*, vol. abs/1610.02357, 2016.

[9] David Hasler and Sabine Suesstrunk, "Measuring colourfulness in natural images," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 5007, pp. 87–95, 06 2003.

[10] Terence Parr, "The difference between l1 and l2 regularization," https://explained.ai/regularization/L1vsL2.html.

[11] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

A. LOSS GRAPHS

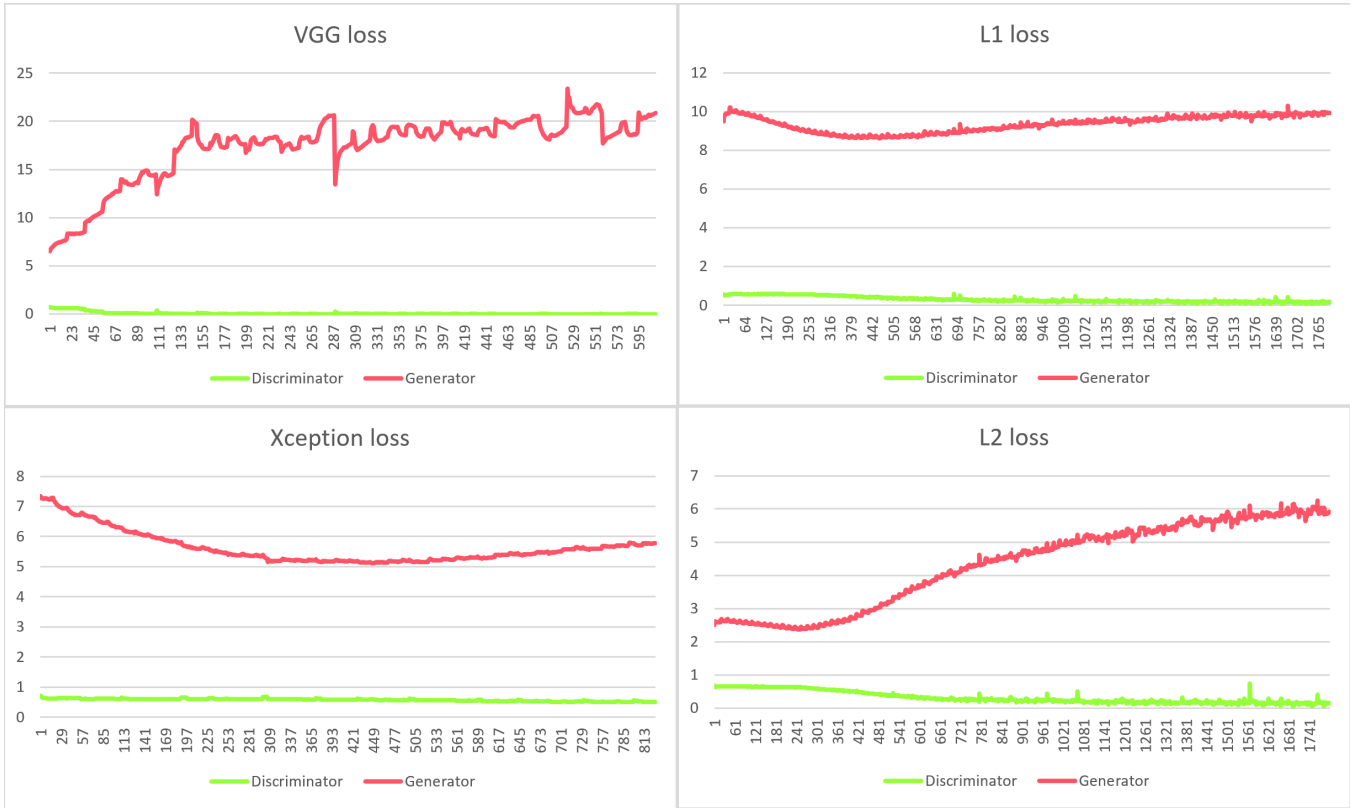

Fig. 7. Loss graphs for each model. the x-axis is more frequent than the number of epochs for vgg and xception data was logged every 300 iteration and for the baselines for every 100 iterations. One epoch is roughly 1875 iterations. All models were trained for 100 epochs.