Edge Map and Saliency Map Guided Multi-path GAN based Image Colourisation and Representation Learning

Koteswar Rao Jerripothula
Assistant Professor, IIITD

Shradha Agarwal
MT20123

August 24, 2021

1 Abstract

There has been quite a lot of development in the field of self-supervised learning. One of the pretext tasks is image colourisation. We propose a GAN-based colourisation model which outputs a colourful image given a gray scale image. And, by doing so, the model also captures features which are generic in nature and has shown an outstanding performance for the downstream task of image classification. Traditional methods have shown issues like colour bleeding and semantic confusion. To alleviate these shortcomings, edge map and saliency map generation acts as an auxiliary task of the generator. Also, we have used ResNet-18 to make the model capable of running on very less data. Flickr dataset is used for training purpose. We received decent coloured outputs. Moreover, the model performed exceedingly well on the task of image colourisation with an accuracy of 56.47%.

keywords - *Colourisation, self-supervised learning, edge map, saliency map*

2 Introduction

Self-supervised Learning or SSL in short refers to building feature representations from unsupervised data. The SSL based models learn by utilising some characteristics or attributes of the input data as the supervisory signal. This is known as pretext task. The learned representations from SSL models are used to solve a given supervised problem like image classification, image segmentation and object detection. These tasks are known as downstream tasks. Several researches have shown how the need of very huge number of labelled training samples gets alleviated by using SSL. Now, with the help of learned representations from the pretext tasks, there is a requirement of only few human-labelled annotations.

Image colourisation is one of the pretext tasks which we have studied and have proposed an SSL edge and saliency guided GAN based image colourisation network. It is the task of adding colours to any given grayscale image and to make it more pleasing to the human eye. In image colourisation, the colours may vary across images thus make this a sophisticated task. Pres-

ence of an environment or object also determines the colour of the image.

The proposed self-supervised model is an outstanding feature representation learner and gave an accuracy of 56.47% for the task of image classification. This is achieved by training on only 8000 images from CIFAR-10 dataset. Whereas, training the downstream model from scratch gave an accuracy of 55.12% when trained on 8000 images from CIFAR-10 dataset. It can be observed that the proposed model could beat the model which was trained from scratch. The learned representations are generic enough to be used by the downstream tasks such as image classification. (Figure 1) depicts the entire workflow.

Our contributions include a) a proposal of a self-supervised edge and saliency map guided colourisation GAN model, b) a study of the encoder performance part of the generator as a feature extractor for the downstream task of image classification, and c) two ideas proposal for pre-text task in SSL.

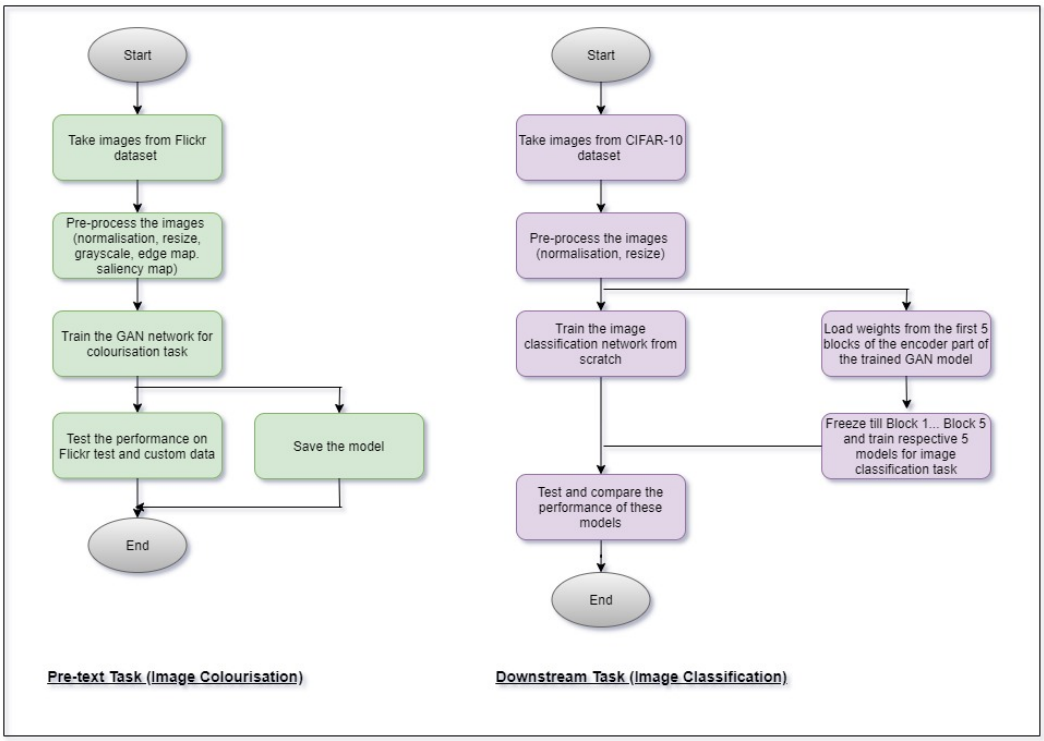

Figure 1: Flowchart for the pre-text and downstream tasks

3 Related Work

There has been a lot of research done in the field of self-supervised learning to learn useful feature representations from unsupervised data. The task used to learn these representations is named as pre-text task and the model which uses these represen-

tations to solve any given problem is the downstream task. Several researches done in this field are discussed below.

Doerch et al. [1] worked on learning visual representations by locating one patch given a reference patch. They experimented on ImageNet and Pascal VOC datasets. They built two AlexNet style architecture networks with combined weights and merged them with fully connected layers. The softmax layer at the end allocates a probability to the eight possible locations of the patch. They have used varied data augmentation techniques to help prevent trivial solutions. Few shortcuts include boundaries and patterns between the patches and colour aberration. To resolve these trivial solutions, they jittered up to seven pixels, kept a gap between the patches and performed green and magenta shift towards gray. One of the challenges here includes choosing the uniform patches which could fit into any of the eight possible locations. Nevertheless, this is one of the pioneer works and did push many ideas in SSL domain.

Pathak et al. [4] proposed image painting to obtain the visual representations by using context encoders to fill the created missing region in the image. For this task, they used AlexNet architecture and fully connected layers for encoder and decoder respectively. The authors worked on StreetView, ImageNet and Pascal VOC datasets and have resized the images to 227 X 227 with 25% missing region in each image. They used reconstruction loss for image context capturing and adversarial loss to select the suitable mode. Both the losses together gave much sharper results. Whereas, using only the reconstruction loss resulted in blurry outputs. They could achieve an accuracy of 56.5%. One of the reasons for low accuracy is the domain gap, i.e., the downstream tasks have 0% missing regions whereas the pretext task has 25% missing regions.

Zhang et al. [5] studied colourisation as a pretext task for learning the visual representations. They proposed a model which takes the L channel of the image and predicts one of the 313 colour blocks of the AB colour channels. By performing the colourisation task, the model has to learn to identify the objects in the given image and thus learns the visual representations. For this task, they used ImageNet dataset. Authors observed that any given image, most of the space is occupied by the background. Thus, the model learns the background colours faster and face difficulty in learning the foreground colours. To solve this issue, they introduced multinomial cross entropy with an added term for colour re-balancing. The authors achieved a score of 32.3 on perceptual realism test.

Noroozi et al. [3] proposed jigsaw puzzle to learn the visual features by predicting the index corresponding to the correct permutation of the given jumbled up input image. The authors stored only the best 64 permutations based on Hamming distance and have used input size of 225 X 225. A random permutation is selected, the image is jumbled up and passed to the network which in turn, predicts the index of the permutation to which the image belongs to. The authors used colour jitter to control chromatic aberration and kept 21 pixels partition is kept to avoid edge continuity issue.

Ledig et al. [7] proposed SRGAN, Self-Resolution GAN to generate high-

resolution (4x) images from very low-resolution images. For this task, they used ImageNet dataset and perceptual loss function. The generator creates high-resolution images while the discriminator discriminates between the real image and the generated image. The loss function for the generator includes L2 loss and content loss. The discriminator loss includes binary cross-entropy. The semantic representations learnt from SRGAN can be used for varied downstream tasks.

Zhang et al. [8] studied split-brain autoencoders which predicts a part of the colour channels given the other parts of the colour channels. The loss functions include L2 loss when seen from regression perspective and multi-class classification loss when seen from classification perspective of predicting to which colour bin the output belongs to. The authors used ImageNet, NYUD and Places dataset. The varied colour spaces studied are RGBD and LAB. In RGBD space, one part of the network is given RGB channels and it predicts for the D channel. Whereas, the other part gives D channel as the input and generated RGB channels. In LAB space, one part of the network is given L channel and it predicts AB channel and vice versa for the other part. Now, the outputs from both the parts are concatenated to generate the original input image. Here, the entire input image is used for feature extraction.

Gidaris et al. [9] proposed prediction of rotation angle as the pretext task. This is a classification task where they have four classes namely, 0, 180, 90 and 270 degrees. They used ImageNet and PASCAL datasets. Alexnet architecture is used and the loss function used is the log loss. The authors observed that in order to predict the rotation angle, the model needs to learn what objects are present in the image and its orientation. Moreover, it has no low-level artifacts like image resizing. Thus, this task is useful for varied downstream tasks.

Noroozi et al.[12] proposed jigsaw++ for learning feature representations. Here, any random tile is taken which replaces one to two tiles in the puzzle [4]. This random tile is the noise. The model here needs to first find where the noisy tile is and then it needs to the jigsaw puzzle. This makes the task more challenging than [3] which worked a good feature representation for the varied downstream tasks.

Chen et al. [13] proposed SSGAN, self-supervised GAN to solve a common issue of forgetting in GANs. The authors observed that with time the generator tends to forget the prior class distribution even when the tasks are similar. To overcome this, they introduced an auxiliary task of predicting the angle of rotation in the discriminator. It is a four-class classification problem with classes being 0, 90, 180 and 270 degrees. Now, the discriminator predicts real or fake and does the rotation prediction. It improved the GAN performance and help alleviate the issue of forgetting in GANs. Also, the authors used discriminator as a feature learner for the tasks of classification, segmentation and object detection.

The pre-text task of colourisation in itself has shown tremendous improvement. Several researches have been done in this field. In earlier researches, non-parametric methods were used wherein user provides a gray image and some coloured reference image

and the model generates the coloured output.

Cao et al. [6] proposed conditional GAN with condition information and multi-layer noise to enhance diversity and to maintain realism. The generator is a fully convolutional generator. On Turing test, a score of 80 was achieved which depicts that their colours were convincing. Manjunatha et.al [10] proposed the task of colourisation by manipulating the model with different captions. For this, the authors used representation wise linear transformation on the output of each convolution layer. Nazeri et al [11] used U-Net [2] architecture based DCGAN which has alternating cost function and predicts A and B channels. Jheng et al [14] used MaskRCNN to extract object wise features which were then fused with the global features of the images. The loss function used is L1 loss and the PSNR score achieved is 27.54.

Overall, several techniques have been proposed in the field of self-supervised learning which includes split brain auto-encoder, colourisation and jigsaw puzzle, among many other techniques discussed above. These approaches work well for the downstream tasks like object detection, semantic segmentation and image classification. Also, a lot of development has taken place in the field of colourisation task.

The rest of the paper discusses ideas proposal, methodology, baselines, analysis, performance on image classification, conclusion and references.

4 Ideas Proposal

4.1 Simple image generation given a complex image

It can be hypothesized that the last few layers of any given image classification model contain the object itself. This is because the model is to associate that object a class. In other words, the image classification model must be working on extracting the object from the image before classifying it. Using this intuition, we can develop a self-supervised model which does exactly this task, i.e., finding the key object from a given image. Thus, easing the process of image classification which will now require only few labelled training samples. Few key terms include complex image which is an image having an object placed in non-uniform background like a bird with trees in the background; Simple image which is an image having an object with a clear background like a bird in front of a white background. The aim is to generate simple image given a complex image using self-supervised learning. For the dataset creation, we have crawled 300 images with simple background and 1000 background images. Now, these simple images are concatenated with the background images to form complex images. The created complex images act as an input to the generator and the output of the generator will be a simple image. The model type is GAN in the category of image-to-image translation. The loss function includes GAN loss along with L1 loss of the generated simple image and its corresponding real simple image.

4.2 Frequency domain analysis

To the best of our knowledge, the work done so far in the field of self-supervised learning is in the spatial domain. The frequency domain is also one of the potential areas where SSL can be studied. FFT can be applied to an image which acts as a supervision and a model can be built which takes an image in the spatial domain and generates its corresponding map in the frequency domain.

5 Methodology

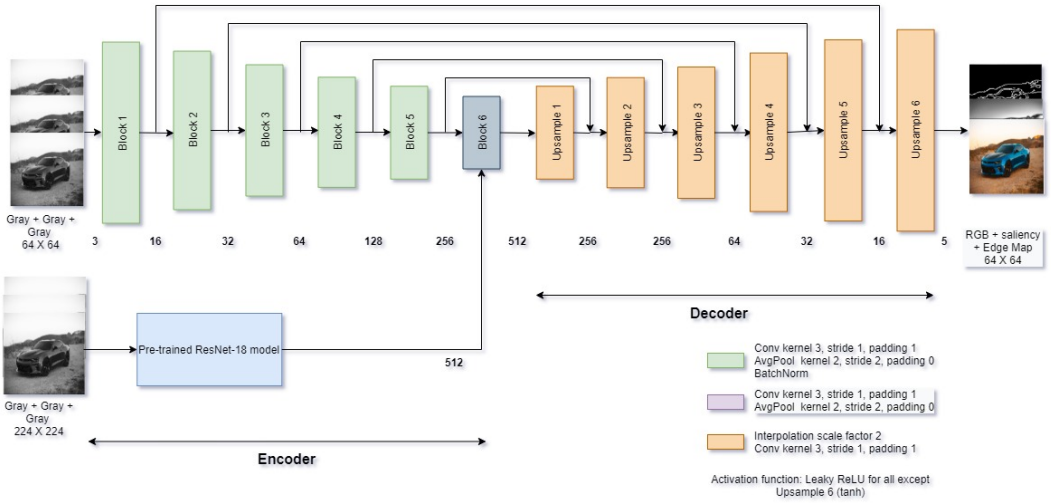

Figure 2: Generator Architecture

Data Pre-processing: The RGB training samples are converted to grayscale and resized to 64 X 64. Grayscale images are stacked together to create a 3-channel input. For creating edge map, Canny edge detection is used. Saliency map is also constructed for the training data. Also, the samples are normalised to a range of -1 to +1. Finally, the data shuffling is done and batches of size 64 are created.

Input: The generator takes three channel inputs of size 64 X 64. The grayscale image created above are the inputs to the generator.

Output: The 5-channeled output of the generator consist of the RGB image in the first three channels, saliency map as the fourth channel and edge map as the fifth channel. The size of the output is of 64 X 64.

Generator architecture:(Figure 2) It is composed of an autoencoder with skip connections and a pre-trained version of the ResNet-18 model. The encoder part consists of six groups of three layers each, namely convolution, average pool and a batch normalisation layer. The features from ResNet are concatenated with fifth batch normalisation layer and acts as an input to the sixth convolution layer. The decoder part of the generator consists of six up-sampling

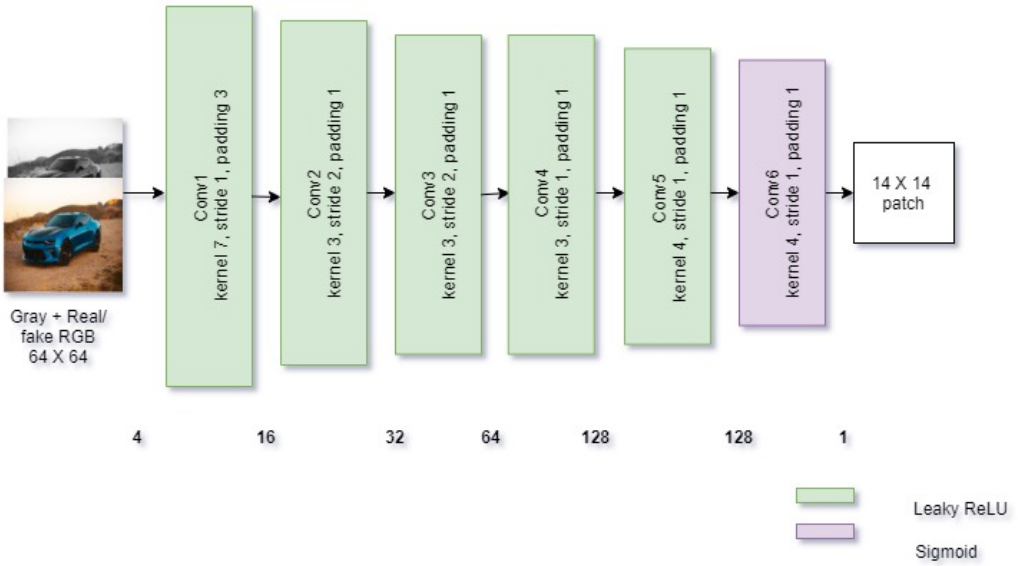

Figure 3: Discriminator Architecture

layers. The corresponding layers in the encoder part are connected via skip connections. LeakyReLU is used as the activation function for all the layers except the last layer wherein tanh activation function is used.

Discriminator architecture:(Figure 3) It is composed of six 2D-convolutional layers. The activation function used in all the layers is LeakyReLU and sigmoid for the output layer. The discriminator output is of 14 X 14 and value of each pixel here denotes the probability of that pixel to be real or fake.

Discriminator Loss: It is the difference of the losses in the fake RGB samples (generated RGB samples) and the real RGB samples.

$\text{Loss\_Discriminator} = E[D(G(x_{gray}, x_{gray})_{RGB})] - E[D(x_{RGB}, x_{gray})]$

Here, x is the input image, G is the generator, D is the discriminator and E is the expectation.

Generator Loss: It comprises of four loss terms: L1 loss of the RGB image, L1 loss of the weighted saliency map, L1 loss of the weighted edge map and the GAN loss. L1 loss of the RGB image is the difference between the generated RGB image and the original RGB image. L1 loss of the weighted saliency map is the difference between the generated saliency map multiplied by the generated RGB image and the real saliency map multiplied by the real RGB image. L1 loss of the weighted edge map is the difference between the generated edge map multiplied by the generated RGB image and the real saliency map multiplied by the real RGB image. GAN loss is calculated as the negative of the mean of output of the discriminator which takes the input of stacked fake RGB image and its corresponding input grayscale image. The weights used are 1, 0.5, 0.5 and 0.05 for L1 loss for the RGB image, GAN loss, L1 loss for the weighted saliency and L1 loss for the weighted edge map respectively.

Loss Generator $= L1\_loss + \lambda_{sal} * saliency\_loss + \lambda_{edge} * edge\_loss + \lambda_G * GAN\_Loss$

Here, λ_G, λ_{sal}, λ_{edge}, GAN_Loss are the weights for generator loss, saliency loss, edge loss and GAN loss respectively.

Xavier initialization is used for initialization purpose. Adam optimiser is utilised with learning rates of the discriminator and generator are 1 X 10^{-4} and 1 X 10^{-3} respectively.

6 Dataset and Baselines

For training purpose, Flickr30k dataset is used. The training samples were 15k. For testing purpose, we have used another 5k samples from Flickr30k images dataset. The test has also been conducted on few random samples taken from Google images. The results are reasonably good across the test set as well as the custom dataset.

6.1 Baseline 1: [15]

Architecture Generator: The network is similar to the Main colourisation Network with seven layers in the encoder part and seven in the decoder part. The activation function used is LeakyReLU except the output layer which uses tanh. For faster convergence, skip connections are used. The loss functions include GAN loss and L1 loss for the generated RGB image.

Architecture Discriminator: It consists of six convolution layers with activation function as LeakyReLU except for the output layer which uses sigmoid activation function. The input to the model is 4-channel consisting of gray image stacked with either real or fake RGB image. The output size is 30 X 30. The loss function is calculated by taking difference between mean of the output of the discriminator with real RGB image and fake RGB image.

Implementation details: The learning rate for the generator is 2e-4 and for the discriminator is 1e-3. Adam optimiser is used and the model is run for 100 epochs.

Observations – Considering Colab's RAM, GPU,disk space constraints our model could learn colourisation and could generate plausible coloured outputs.(Figure 5). At the end of 100^{th} epoch, the loss was 0.04. (Figure 4).

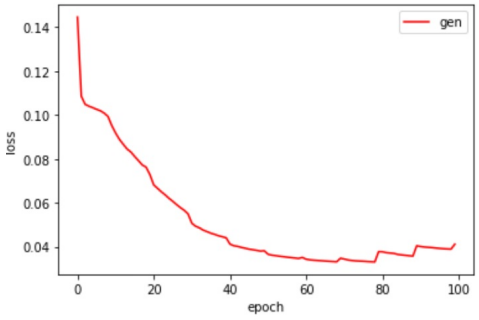

Figure 4: Training loss

Figure 5: Gray(L), Fake(M),Real(R)

6.2 Baseline 2: reference [5]

The model takes gray scale image as the input and outputs the probability distribution of pixels in AB colour channels which is divided into 313 colour bins. For creating the output training data, soft encoding is done.

Architecture:It consists of five groups of three or two conv2D layers. The activation function used is ReLU. Each group is followed by a layer of Batch Normalisation. After the fourth block, up-sampling is done. The last layer uses softmax activation function.

Implementation details: The loss function used is multinomial cross entrpy loss and the optimiser used is NAG. The learning rate is kept at 0.01 and the model is trained for 200 epochs.

Observations: From (Figure 7) shows the ability of the model to produce coloured outputs in very small number of epochs.

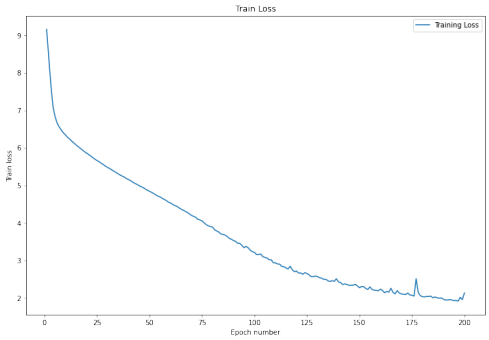

Figure 6: Training loss

Figure 7: Gray(L), Fake(M),Real(R)

7 Analysis

7.1 Result

It can be observed from Figure 8 and Figure 9 that the proposed model did produce beautiful colourful images across varied categories and could solve the issue of colour bleeding. The samples in the first row of Figure 5 and Figure 6 are the best outputs whereas, few failure cases are mentioned in the second row. The model could colour categories like dog, face, water, sky, grass. Whereas, the failure cases can be seen in objects which could have got different valid colours like cars, house and clothes. The PSNR score obtained by the proposed model for the Flickr Test set is 18.98.

Figure 8: Flickr test set

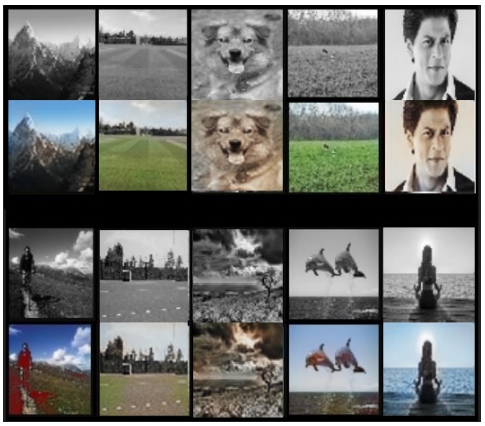

Figure 9: Random Test Data

7.2 Error

Considerable amount of error is present which is obvious considering limited GPU and RAM constraints. The proposed model failed to produce decent colours for the objects which could have multiple possible colours. It does not work on images with high colour variance like rainbow or peacock. This model is trained on Tesla K80 GPU and 12 GB RAM by Google Collaboratory. The performance could have been better if we fine-tuned ResNet-18 which normally all multi-path networks do. Only 15k training samples could be used whereas SCGAN was trained on 1.3L images. This is one of reasons why the outputs are not atpar for few cases. Also, it is challenging to find the right hyper parameters for any GAN based model. Not many hyper parameter combinations could be checked due to limited usage time of the given GPU.

7.3 Visualisation

From Figure 7, a decrement in overall generator loss can be observed as the number of epochs increased. And, somewhere near to epoch 50, the graph has become stagnated. A possible reason for this is less data. The normal phenomenon of tug-of-war can be observed for discriminator and that the generator tries to retain it in balance. In this case, it can be seen that with the increase in the number of epochs, the discriminator loss is increasing while the generator loss is decreasing. It attributes to a comparatively strong generator while the discriminator struggled to discriminate between real and fake samples. In other words, the generator is able to capture the True data distribution fairly well.

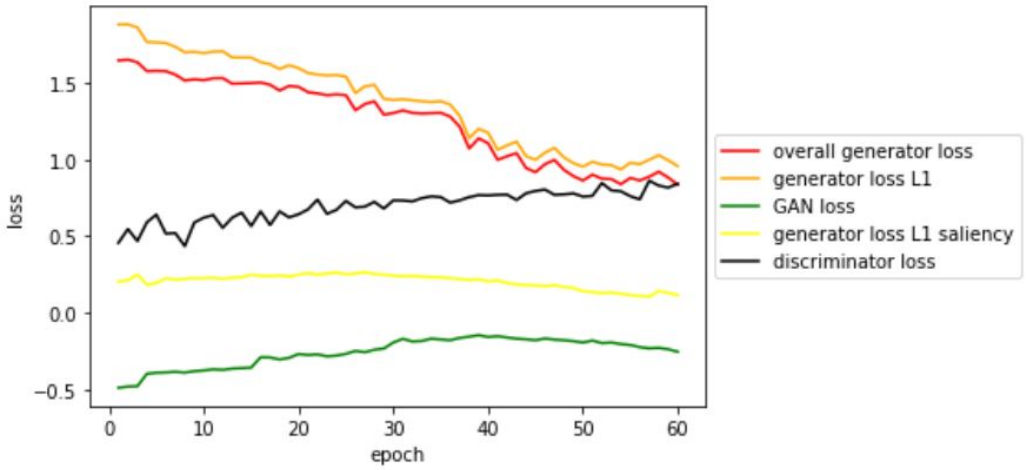

Figure 10: Loss curves for the proposed model

8 Downstream Task - Image classification

The proposed model is used as a feature extractor for the downstream task of image classification. The first five set of layers of the encoder part of the generator is used for fine-tuning. These set of layers are followed by alternating three fully connected layers and two dropout layers. Refer (Figure 11) for the model architecture. Several cases have been studied which includes only the first block are frozen, first two blocks are frozen, first three blocks are frozen, first four blocks are frozen and the last case is keeping all the five blocks frozen. These models have been trained for image classification using 8000 samples from the CIFAR-10 dataset.

Also, another model is trained having the same layers as that of first five blocks of the encoder part of the generator of colourisation model followed by alternating three fully-connected layers and two dropout layers. It is trained on 8000 samples from the CIFAR-10 dataset. The varied accuracy and loss plots can be seen in the figures 12 to 23.

From the Table 1, it can be observed that the best performance is achieved when the block 1 weights were kept frozen and the rest for the model was trained. The test accuracy obtained in this case is 56.47% whereas the validation accuracy is 55.94%. Moreover, as the number of frozen blocks is increased from colourisation model, the accuracy decreases. This indicates that the model slowly becomes specific to the task assigned as we go deeper in network. The model which was trained from scratch performed better than the cases when weights till block 4 and till block 5 were frozen but could not perform better than the models with frozen weights till block 1 and block 2. The test accuracy obtained is 55.12% and the validation accuracy is 52.13%. Thus, we can say that for the image classification task, the self-supervised model could learn meaningful generic representations which work better than its supervised version.

The analysis of other downstream tasks such as object detection and semantic segmentation is left for future studies.

Table 1. Performance obtained under different cases			
Weights frozen till	Train Accuracy	Validation Accuracy	Test Accuracy
Block 1	**62.98**	**55.94**	**56.47**
Block 2	61.24	53.37	55.42
(Trained from scratch)	56.03	52.13	55.12
Block 3	55.99	50.75	51.33
Block 4	54.31	48.69	48.11
Block 5	54.27	48.44	47.56

Dataset: Cifar-10 dataset; Number of samples used for training the models: 8000; Number of samples used for testing the models: 1600

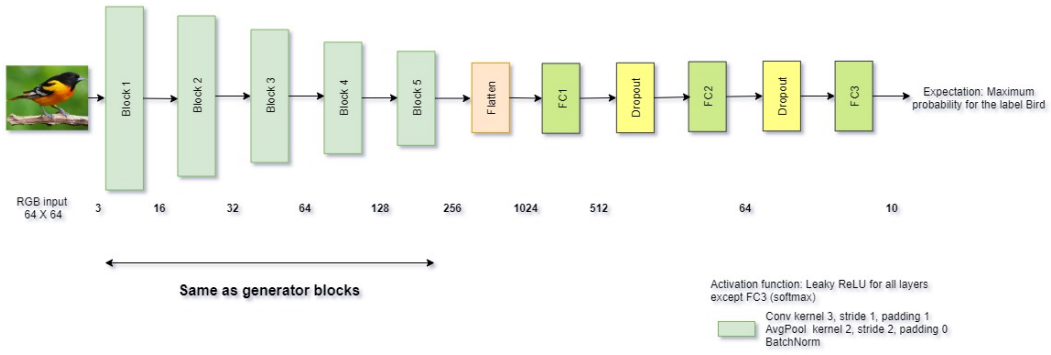

Figure 11: Image classification Architecture

9 Conclusion

In this paper, we proposed a self-supervised multi-path GAN based colourisation model which generates beautiful coloured images. A wide variety of images containing landscapes, sky, ground, dog, skin could be coloured using the proposed model. The colourisation model proved to be an outstanding representation learner. It could beat its supervised counterpart (model trained from scratch) by giving the test accuracy of 56.47% and validation accuracy of 55.94% for the downstream task of image classification. Moreover, two news ideas of self-supervised learning are proposed namely, simple image generation given a complex image, and frequency domain analysis.

10 References

[1] C. Doersch, A. Gupta, and A. A. Efros. "Unsupervised Visual Representation Learning by Context Prediction". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1422–1430. DOI: 10.1109/ICCV.2015.167.

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].

[3] M. Noroozi and P. Favaro. "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles". In: *ECCV*. 2016.

[4] D. Pathak et al. "Context Encoders: Feature Learning by Inpainting". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2536–2544. DOI: 10.1109/CVPR.2016.278.

[5] Richard Zhang, Phillip Isola, and Alexei A. Efros. *Colorful Image Colorization*. 2016. arXiv: 1603.08511 [cs.CV].

[6] Yun Cao et al. *Unsupervised Diverse Colorization via Generative Adversarial Networks*. 2017. arXiv: 1702.06674 [cs.CV].

[7] C. Ledig et al. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 105–114. DOI: 10.1109/CVPR.2017.19.

[8] R. Zhang, P. Isola, and A. A. Efros. "Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 645–654. DOI: 10.1109/CVPR.2017.76.

[9] S. Gidaris, P. Singh, and N. Komodakis. "Unsupervised Representation Learning by Predicting Image Rotations". In: *ICLR*. 2018.

[10] Varun Manjunatha et al. *Learning to Color from Language*. 2018. arXiv: 1804.06026 [cs.CV].

[11] Kamyar Nazeri, Eric Ng, and Mehran Ebrahimi. "Image Colorization Using Generative Adversarial Networks". In: *Lecture Notes in Computer Science* (2018), pp. 85–94. ISSN: 1611-3349.

[12] M. Noroozi et al. "Boosting Self-Supervised Learning via Knowledge Transfer". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9359–9367. DOI: 10.1109/CVPR.2018.00975.

[13] Ting Chen et al. "Self-Supervised GANs via Auxiliary Rotation Loss". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[14] Jheng-Wei Su, Hung-Kuo Chu, and Jia-Bin Huang. *Instance-aware Image Colorization*. 2020. arXiv: 2005.10825 [cs.CV].

[15] Yuzhi Zhao et al. “SCGAN: Saliency Map-guided Colorization with Generative Adversarial Network”. In: *IEEE Transactions on Circuits and Systems for Video Technology* (2020), pp. 1–1. ISSN: 1558-2205.

Name: Shradha Agarwal
Enrollment number: MT20123
Course: M.Tech. (CSE)