Mathematical Statistician and Engineering Applications
ISSN: 2094-0343
2326-9865

A comprehensive Analysis on Image Colorization using Generative Adversarial Networks (GANs)

Hazel Mahajan
Jaypee Institute of Information Technology, Noida,India.
Email: 8500hm@gmail.com

Article Info
Page Number: 9748 - 9755
Publication Issue:
Vol 71 No. 4 (2022)

Article History
Article Received: *15 September 2022*
Revised: *25 October 2022*
Accepted: *14 November 2022*
Publication: *21 December 2022*

Abstract
Using a training set of matched picture pairs, image-to-image translation learns the mapping between an input image and an output image. For many issues, however, matching training data will not be available. We propose a method for learning to translate a picture from a source domain X to a target domain Y in the absence of matched examples. We wish to train a mapping A: X (Y) using an adversarial loss such that the distribution of photos from A(X) is indistinguishable from the distribution Y. Because it is significantly under-constrained, we connect it with an inverse mapping B:Y(X) and apply a cycle consistency loss to push B(A(X))X and A(B(X))X. Qualitative outcomes are presented on several tasks when matched training data is unavailable, such as collection style transfer, obseason transfer, photo enhancement, and so on. Quantitative comparisons to previous approaches reveal that our approach is superior.

Image colorization is the process of adding colours to a grayscale image in order to make it more aesthetically appealing and perceptually important. These are considered sophisticated assignments that typically require prior knowledge of image content as well as human adjustments to achieve artifact-free quality. Furthermore, because objects might have varied hues, there are multiple techniques for assigning colours to pixels in a picture, meaning that there is no one answer to this problem.

Deep learning has recently acquired popularity among academics in the fields of computer vision and image processing. Convolutional neural networks (CNNs) are a common technology that has been well-studied and effectively used to a variety of applications such as image identification, picture reconstruction, image production, and so on.

SUMMARY

Image colorization is the process of adding colours to a grayscale image to make it more aesthetically appealing and perceptually important. These are considered sophisticated jobs because they typically require prior knowledge of image content as well as human adjustments to achieve artefact-free quality. Furthermore, because objects might have varied hues, there are multiple techniques for assigning colours to pixels in a picture, meaning that no one solution to this problem exists.

Nowadays, image colorization is mainly done by hand in Photoshop. Several organisations employ picture colorization services to assign colours to grayscale old images. There is also for colorization reasons in the documentation photograph. Using Photoshop for this purpose, on the other hand, requires more effort and time. One answer to this challenge is machine learning.

Deep learning has lately piqued the interest of academics in computer vision and image processing. Convolutional neural networks (CNNs) have been extensively researched and effectively used for a wide range of tasks such as image identification, reconstruction, and creation.

SCOPE OF IMAGE COLORIZATION

Within computer vision, image colorization is a common issue. Image colorization's ultimate goal is to convert a grayscale image into a visually plausible and perceptually meaningful colour image.
It's worth noting that image colorization is a poorly defined problem. The goal, according to the definition, is to get "visually believable and realistic image colorization," which is constrained by the problem's multi-modal nature — different colorizations are conceivable for a grey-scale image.
Colour is nearly often linked to very personal incidents. Our capacity to distinguish millions of colour hues has provided us with several advantages throughout evolution; after all, we can safely distinguish food from harmful or toxic substances or damaged things owing to our colour vision. As a result, many image processing programmes in which the human eye will make the ultimate interpretation are particularly built to improve colour reproduction for humans.
Colour cameras are in high demand. This significant interest can be explained by emerging applications in which pictures are no longer employed exclusively for computer-based analysis but must also be of acceptable quality to the human sight.
Furthermore, the inclusion of colour information to an image expands and diversifies the options for image processing and analysis.

IMAGE PROCESSING CONCEPTS

An image is defined as a two-dimensional function (x,y), where x and y are spatial coordinates, and the amplitude of F at any pair of coordinates (x,y) is referred to as the image's intensity at that location. We call it a digital picture when the x,y, and amplitude values of F are finite.
In other terms, a picture may be described as a two-dimensional array with rows and columns.
A digital image is made up of a finite number of elements, each of which has a specific value at a specific position. Picture elements, image elements, and pixels are all names for these components. A pixel is commonly used to represent the elements of a digital image.
Types of an image

BINARY IMAGE– The binary picture, as the name implies, has only two pixel elements: 0 and 1, where 0 represents black and 1 represents white. Monochrome is another name for this picture.

BLACK AND WHITE IMAGE– BLACK AND WHITE IMAGE refers to an image that is exclusively black and white.

8-bit COLOUR FORMAT– It is the most well-known picture format. It includes 256 distinct colour tones and is usually referred to as a Grayscale Image. In this format, 0 represents black, 255 represents white, and 127 represents grey.
16-bit COLOUR FORMAT– It is a type of colour picture format. It contains 65,536 distinct colours. It is often referred to as High Color Format. The colour distribution in this format is not the same as in a grayscale image.

Deep learning has recently acquired popularity among academics in the fields of computer vision and image processing. Convolutional neural networks (CNNs) are a common technology that has been well-studied and effectively used to a variety of applications such as image identification, picture reconstruction, image production, and so on. A CNN is made up of several layers of tiny computing units that only process parts of the input picture in a feed-forward method. Each layer is the result of applying numerous image filters to the preceding layer, each of which extracts a different characteristic of the input image. As a result, at different levels of abstraction, each layer may include important information about the input image.

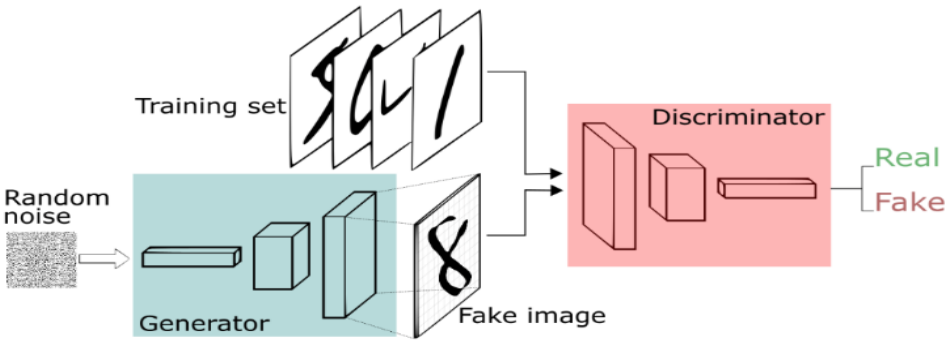
Image Processing Techniques

1. Image Restoration
2. Linear Filtering
3. Independent Component Analysis
4. Pixelation
5. Template Matching
6. Image Generation Technique

INTRODUCTION TO GANS

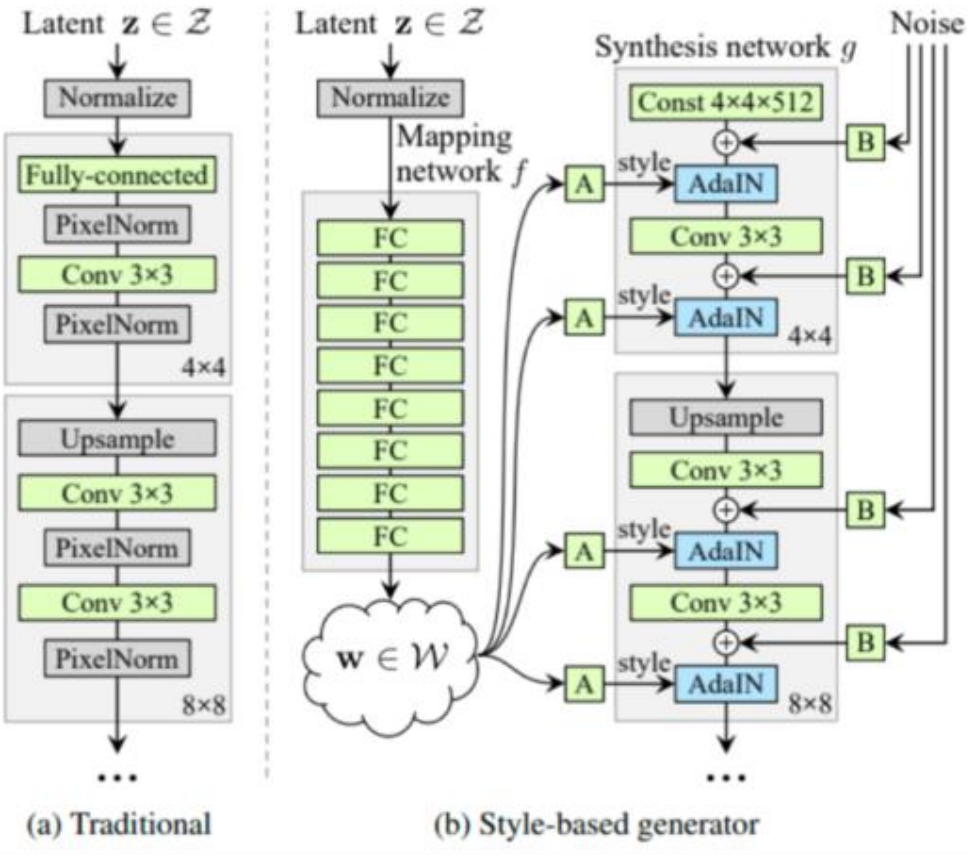
GAN is a generative algorithm introduced by Ian Goodfellow and other researchers in 2014. GAN tries to mimic the distribution of input dataset and accordingly generate new samples from the learnt distribution. GAN is analogous to a game between two players, where one player tries to produce fake notes and the other player tries to distinguish between real and fake notes. GAN is said to be trained well enough when it one player generates fake notes that are good enough to fool the other player to believe that generated notes are real.

ARCHITECTURE -Two smaller networks, namely generator and discriminator constitute generative adversarial network. The task of the generator as mentioned earlier is to produce an output that is indistinguishable from real data. The task of the discriminator, on the other hand is to classify whether a sample came from real data or is fake i.e generated by generator. Architecture of both generator and discriminator is generally a multilayer perceptron model.



WORKING- Generator tries to fool discriminator by generating data instances close to input dataset. Thus, **G** tries to maximise the probability of **D** making a mistake. Discriminator tries to

identify counterfeit data instances produced by the generator. This framework corresponds to a minimax two player game (One for the real images and another for the fake).



BACKPROPAGATION THROUGH TIME

To identify the mistake, we utilise forward-propagation in neural networks to obtain the output of your model and then evaluate whether it is accurate or inaccurate.

Backpropagation is the process of traversing your neural network backwards in order to retrieve the partial derivatives of the error with respect to the weights and subtracting this value from the weights.

Such derivatives are used in gradient descent, a technique for iteratively minimising a given function. The weights are then changed upwards or downwards based on which lowers the error. A neural network learns in this manner throughout the training phase.

So, with backpropagation, we simply try to change our model's weights during training. The diagram below depicts the concepts of forward and backpropagation in a feed-forward neural network:

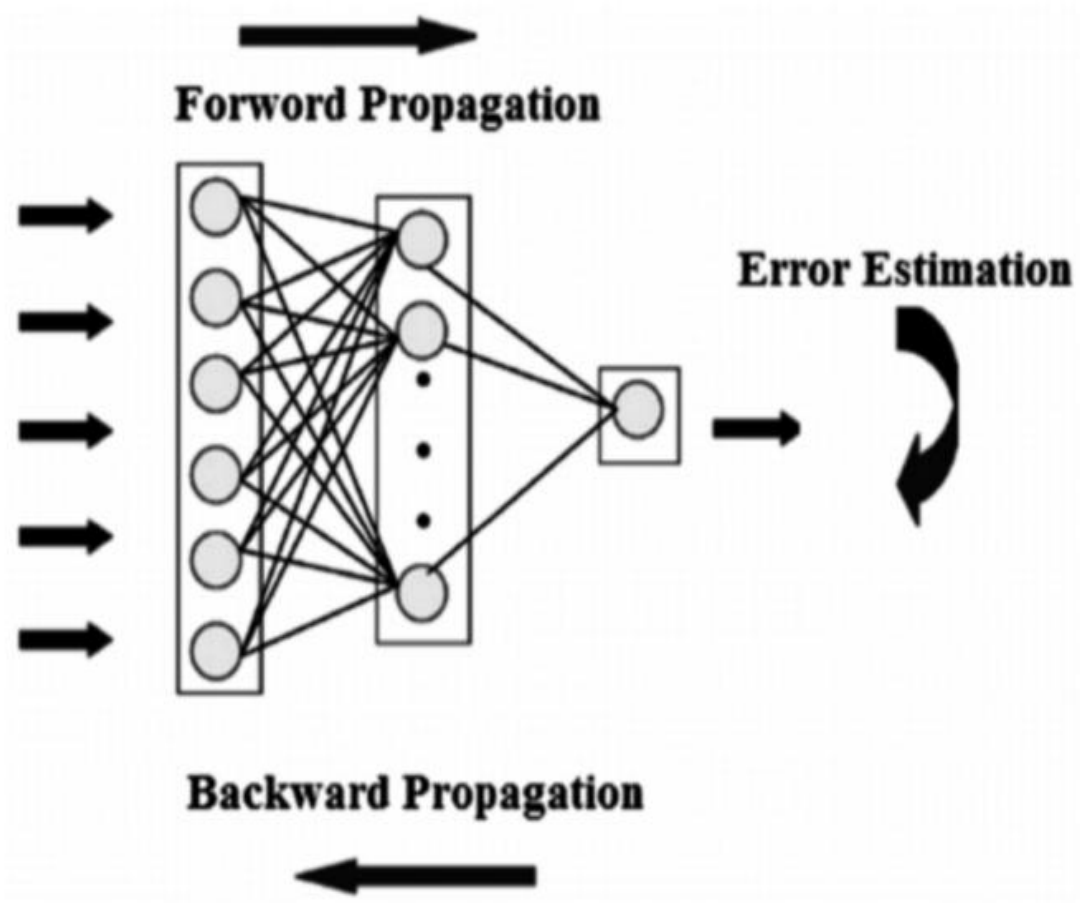
BPTT is used to do backpropagation on an unrolled RNN. Unrolling is a network visualisation and conceptual tool that helps you comprehend what's going on. Backpropagation is usually handled automatically when constructing a recurrent neural network in standard programming frameworks.

A RNN can be thought of as a series of neural networks that are trained one after the other via backpropagation.

An unrolled RNN is seen in the image below. After the equal sign, the RNN is unrolled on the left. Because the different time steps are visible and information is transmitted from one time step to

When performing BPTT, the idea of unrolling is essential since the inaccuracy of a given time step is dependent on the preceding time step.

BPTT propagates the mistake from the last to the first time step. while unrolling all timesteps. This allows you to calculate the error for each timestep, which allows you to update the weights. It should be noted that BPTT can be computationally costly when there are a large number of timesteps.



DATASET USED

Content

Monet2Photo dataset consists of 1193 Monet Paintings & 7038 Natural Photos with each split into train and test subsets.

This dataset was obtained from UC Berkeley's official directory of CycleGAN Datasets. For more details on the dataset refer to the related CycleGAN publication. Work based on the dataset should cite:

```
@Inproceedings {CycleGAN2017,
 title= {Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks},
 author= {Zhu, Jun-Yan, and Park, Taesung and Isola, Phillip and Efros, Alexei A},
 book title= {Computer Vision (ICCV), 2017 IEEE International Conference on},
 year= {2017}
}
```

TRAINING STRATEGIES

For our model implementation, we used the open source Python libraries PyTorch and Keras. We used the free Google Colaboratory GPU to train the model. The size of our batch is 50 images.

BATCH NORMALISATION

Because of a process known as mode collapse, the GAN is thought to be difficult to train. The generator succeeds in convincing the discriminator that the same generated output is true during the model collapse phase. As a result, the generator consistently produces similar outputs, and the generation lacks variation. This is an undesirable situation during the GANs' training phase. To avoid the aforementioned occurrence, we employ batch normalisation, which has been shown to lower the likelihood of mode collapse. However, batch normalisation is avoided in the discriminator and generator's first layer and the generator's last layer, as advised by.

ALL CONVOLUTIONAL NET

Instead of spatial pooling, strided convolutions are used. Instead of depending on fixed downsampling and upsampling, strided convolutions allow the model to learn its own upsampling and downsampling. Using simple convolution layers, this strategy has been shown to increase training performance and assist the network in learning crucial invariances.

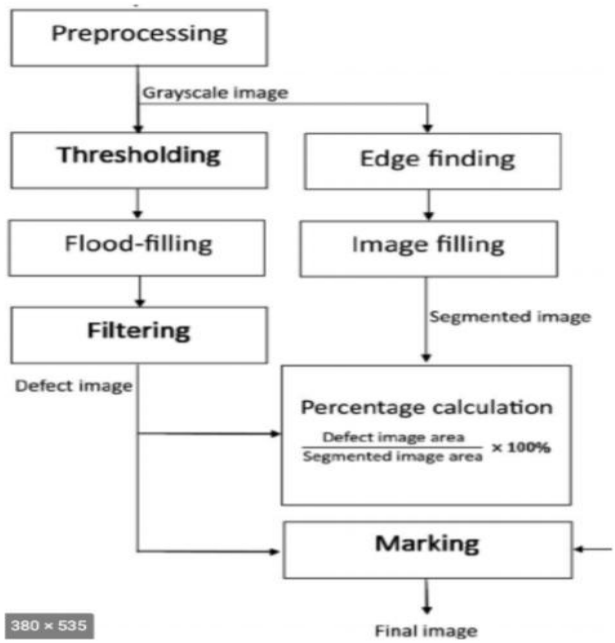
LEAKY RELU ACTIVATION FUNCTION

LeakyReLU activation function gives better performance then normal ReLU, hence we have used it wherever activation function is used.

MODE COLLAPSE AVOIDANCE STRATEGY

In our example, we discovered that the generator generated photos with the same colour pattern as well as colour grids during some of the training. As previously stated, this was the result of 'mode collapse.' We adopted an innovative strategy proposed by our friend Kush Jajal, in which we train the generator to avoid utilising the same colours, in addition to batch normalisation. This is a form of reverse training for the generator, since we intuitively avoid making the same colour mistakes every time.

PROCESS FOLLOWED



CONCLUSION

Overall, our recent work on Cycle GANs has allowed us to enter into the complex and interesting domain of deep learning deployment. The challenges of moving from research to production taught us how to adopt a more industry-oriented approach to machine learning research while also allowing us to examine the leading edge of GANs.

On the research side of this project, testing different numbers of residual layers in the generator and their effects on model performance are some areas for further investigation, and code optimization with something like NvidiaTensorRT or PyTorchOnnx could produce interesting benchmarks on the deployment side. We are really satisfied with this study and what we learned from it, and we are eager to continue exploring the boundaries of deep learning.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[2] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2018. [Online]. Available: https://doi.org/10.1109/cvpr.2018.00917

[3] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *2017 IEEE Conference on*

Computer Vision and Pattern Recognition (CVPR). IEEE, jul 2017. [Online]. Available: https://doi.org/10.1109/cvpr.2017.18

[4] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," 2016.

[5] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to- image translation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2018. [Online]. Available:

[6] C. Esteban, S. L. Hyland, and G. Ra ̈tsch, "Real-valued (medical) time series generation with recurrent conditional gans," 2017.

[7] K. G. Hartmann, R. T. Schirrmeister, and T. Ball, "Eeg-gan: Generative adversarial networks for electroencephalographic (eeg) brain signals,"2018.

[8] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "MAD-GAN:Multivariate anomaly detection for time series data with generative ad- versarial networks," inHazel Mahajan 18102263, ShreySoni 18102286 49 *Artificial Neural Networks and Machine Learning ICANN 2019: Text and Time Series*. Springer International

[9] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial net- works. In CVPR. IEEE, 2017

[10] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional gen- erative adversarial networks. In ICLR, 2016.

[11] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image syn- thesis. In ICML, 2016.

[12] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenen- baum. Learning a probabilistic latent space of ob- ject shapes via 3d generative-adversarial modelling. In NIPS, 2016.