Using Pix2Pix Neural Network to colourise black and white photographs

Samuel Badman
17025835

University of the West of England

January 14, 2022

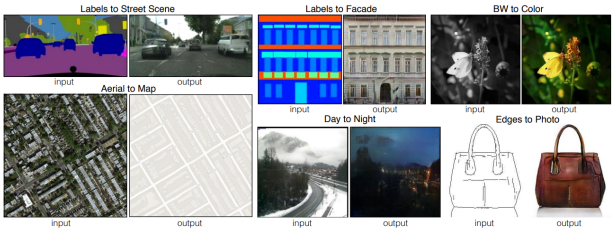

Figure 1: *Example image-to-image domain translations performed using conditional DCGANs (Isola et al., 2018).*

Neural networks can be used to solve image-to-image translation problems such as adding colour to a black and white image. The Pix2Pix implementation of a conditional deep convolutional generative adversarial network was used to colourise the works of Ansel Adams and William Henry Jackson. The accuracy of the results is analysed, and suggestions are made to improve the results of the network.

1 Introduction

Ansel Adams (The Ansel Adams Gallery, 2021) and William Henry Jackson (Britannica, 2021) were early 20th century photographers of parts of western North America including Yellowstone and Yosemite, Sierra Nevada and the Owens Valley. Their work was captured in black and white and can be colourised using artificial intelligence techniques. A Pix2Pix implementation of a conditional deep convolutional generative adversarial network will be used to translate black and white images to coloured images (Figure 1). The network will be trained on paired black and white and coloured images of the photographed locations.

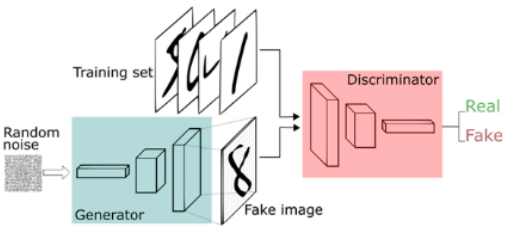

Figure 2: *A GAN is composed of two network models. The generator model generates fake images from random noise and the discriminator model takes input from a training data set and the generator output and predicts whether it is real or fake (Khan, 2021).*

2 Related Work

2.1 General Adversarial Networks

Generative adversarial networks (GANs) (Goodfellow et al., 2014) are a deep learning algorithm implemented to solve generative modelling problems. Generative modelling is a form of unsupervised learning that learns patterns in input data to generate new outputs. Gans achieve this by employing supervised learning

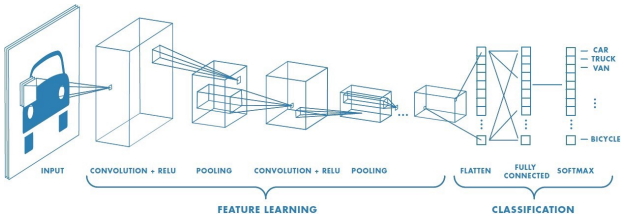

Figure 3: *A CNN is composed of multiple convolution and pooling layers and a final fully connected layer (Saha, 2018).*

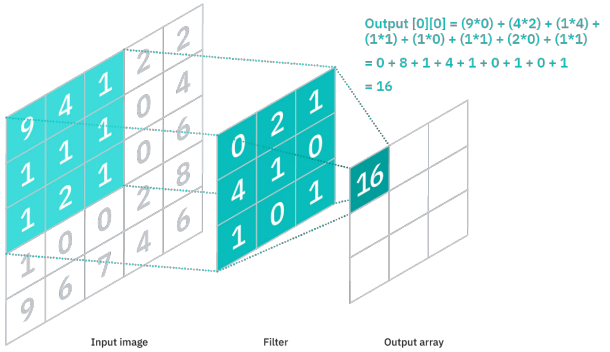

Figure 4: *A filter or kernel is applied to an area of input data and stored in an output feature map. Input data can be a three-dimensional matrix representing a three-channel image (IBM, 2021).*

techniques to train two neural network models (Brownlee, 2019, A Gentle Introduction to Generative Adversarial Networks (GANs)). A generator model is used to create new examples from the problem domain and a discriminator model is used to classify examples as real or fake. Discriminator output is used to update the generator model to improve generated results (Figure 2).

Supervised learning is used to train a model by giving the model input variables with expected output class labels and updating the model to improve the accuracy of its output compared to the expected output (Brownlee, 2019, A Gentle Introduction to Generative Adversarial Networks (GANs))). This requires a training data set with validation examples that are used to check the model's output accuracy and update the model. An example of a supervised learning problem is classification where a model is trained to classify new inputs that it has not seen before.

Unsupervised learning is used to train a model by giving the model input variables without specifying an expected output. As a result, the model is not updated as in a supervised learning environment as there is no expected output to base changes on (Brownlee, 2019, A Gentle Introduction to Generative Adversarial Networks (GANs)). Alternatively, the unsupervised model recognises patterns and similarities in the input data to base its output on.

An improved implementation of GANs is deep convolutional generative adversarial networks (DCGANs) (Radford et al., 2016) that improve GAN results using convolutional neural network (CNNs) implementations for the generator and discriminator models (Brownlee, 2019, A Gentle Introduction to Generative Adversarial Networks (GANs)).

2.2 Convolutional Neural Networks

CNNs are composed of convolutional, pooling and fully connected layers. The first layer is always a convolutional layer followed by additional convolutional and pooling layers before the final layer that is a fully connected layer (Figure 3) (IBM, 2021). Including more layers in the CNN allows it to identify more features from the input and improves the accuracy of its output. CNNs have improved performance over other neural networks and are commonly used for supervised learning problems.

A convolutional layer is composed of input data, a kernel, and a feature map (Figure 4). The kernel is a two-dimensional array of weights that is moved across the input data and multiplied with receptive areas of the input data. The results of the multiplication are stored in the feature map which then has a rectified linear unit (ReLU) function (Brownlee, 2019, A Gentle Introduction to the Rectified Linear Unit (Relu)) applied to introduce non-linearity to the model (IBM, 2021). The applied kernel detects features in the input data and stores the result in the feature map which can then have additional convolutional layers applied hierarchically to further refine feature detection.

Pooling layers reduces the number of parameters in the input data by moving a kernel across the input data and applying a function to the values in the receptive area. The applied function uses either a max pooling type, that selects the maximum value for each input, or an average pooling type, that calculates the average value. Results are stored in an output array to be used in further layers in the network (IBM, 2021). This layer type reduces complexity of the input data, improving network performance and reducing the risk of overfitting.

Fully connected layers map parts of the previous layer to parts of the output layer (IBM, 2021). This is necessary as previous layers can be partially connected with input data being sized differently to output data. Fully connected layers apply a SoftMax activation function (Wood, 2021) to produce a classification probability between 0 and 1 (IBM, 2021).

2.3 Generator And Discriminator Models

The generator model generates new examples of the problem domain once it has been trained with a data set. Training the model builds a representation of the problem domain with compressed data containing key features from the problem domain. This is called latent

space (Tiu, 2020). N dimensionally compressed data in the latent space can be sampled at various locations with an N dimensional vector (Brownlee, 2021, A Gentle Introduction to Generative Adversarial Networks (GANs) to generate new outputs. Locations in the latent space that are near to each other contain similar features.

Generated, fake examples along with real examples from a training data set are given to the discriminator model which outputs a predicted real or fake label for the input. The discriminator model is trained alongside the generator model and is updated to increase the accuracy of predicted real or fake labels when an input is incorrectly classified. The generator model is updated based on how well it managed to fool the discriminator model (Brownlee, 2021, A Gentle Introduction to Generative Adversarial Networks (GANs)). The discriminator model is discarded once training is complete as it is only used to provide supervised learning to train the generator model.

2.4 Conditional Deep Convolutional Generative Adversarial Networks

DCGANs have been extended by introducing an additional input variable during training, making the network a conditional DCGAN as it has been conditioned on an additional input. Both the generator and discriminator models are trained with the conditioning variable allowing the models to be trained for a specific example from the problem domain (Brownlee, 2019, A Gentle Introduction to Generative Adversarial Networks (GANs)). Conditioning enables the DCGAN to be used for tasks such as image-to-image translation (Isola et al., 2018) where an input image is translated from one domain to another.

2.5 Previous Works

A colourisation method is presented (Joshi et al., 2020) that uses a CNN with the pretrained Inception-ResnetV2 model (Szegedy, 2016) and back propagation to detect patterns in RGB and greyscale values. The CNN is also trained on a custom dataset of historical images at 256x256 resolution producing an accuracy of 75.23%. The accuracy of the output of the model is objectively measured using mean squared error (MSE) and peak signal-to-noise ratio (PSNR) functions (Joshi et al., 2020).

Results show that the presented method was mostly successful at colourising greyscale images. It is suggested that results could be improved in small details of an image by increasing the size and variability of the training data set. Furthermore, alternative objective measuring methods could be used to MSE and PSNR to improve the accuracy of the model accuracy measurement. Finally, the use of a GAN is suggested to produce a better result (Joshi et al., 2020).

Figure 5: *Results obtained from the model presented by (Joshi et al., 2020) used on a validation data set, showing colourisation of greyscale images.*

3 Method

Pix2Pix is an implementation of a conditional DCGAN used for image-to-image translation problems that is trained on paired data (Figure 6). During training, the generator and discriminator networks are conditioned with the same input image (Sharma, 2021) training the networks conditioned with the expected result.

An implementation of pix2pix (Zhu, 2020) was used to colourise the works of Ansel Adams and William Henry Jackson and was trained using a bespoke paired-image training data set. 400 coloured images of the

Figure 6: *An example of paired training data used for an image-to-image translation task where black and white images are colourised. The coloured image (right) is paired with an identical black and white version (left).*

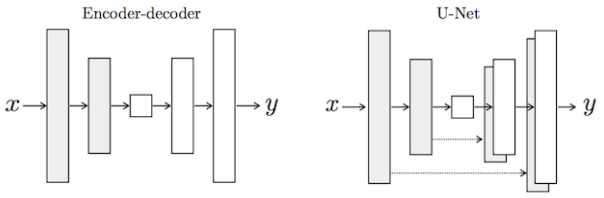

Figure 7: *A diagram representation of encoder-decoder and U-Net network architecture that shows how connections are made between layers of the same size in the U-Net architecture allowing bottleneck layers to be skipped (Brownlee, 2019).*

locations photographed by Adams and Jackson was collected and curated to ensure that the images would produce an appropriately trained model. Images included needed to be taken from a similar viewpoint to the viewpoint used in Adam's and Jackson's works. This meant that aerial photographs could not be used, for example.

Collected images needed to be prepared before they could be used for training. Images were transformed to a size of 256x256 pixels as this was the size used by Joshi et al. (Joshi et al., 2020). Copies of the resized images were converted to black and white and stored in a separate folder to the resized coloured images. Image transformations were applied using batch operations with Photoshop software (Adobe, 2021). Coloured and black and white images were split 70%, 15% and 15% to create training, test and validation data sets respectively.

A script provided in the repository containing the pix2pix implementation (Zhu, 2020) was used to combine the coloured image with its corresponding black and white image. The combined images were used as the paired training data to train the model. The model was trained for 200 epochs with a batch size of 1 to translate images from the black and white domain to the coloured domain.

4 Technical Description

Pix2pix uses U-Net architecture (Figure 7) for the generator model that employs downsampling until a bottleneck layer is reached and then upscaling however, skip-connections are made between layers with the same size (Brownlee, 2019). This architecture allows bottleneck layers to be skipped, sharing low-level information between the input and output, which is useful for image-to-image translation problems.

Pix2pix uses PatchGAN for its discriminator model which is implemented as a CNN and predicts whether an input is real or fake. PatchGAN classifies sections of the input and outputs the probability of that section being real or fake. The probabilities for each section of the image are averaged to provide a single probability output for the entire input image (Brownlee, 2019).

Figure 8: *Results from the Pix2Pix network used to colourise black and white test image data. The table shows two columns of results with the generated fake image (left) and real original coloured image (right).*

Two losses are calculated during discriminator training called adversarial loss and L1 loss. The adversarial loss is calculated based on how similar an output image is to the input, while the L1 loss is calculated based on the difference between the generated translated image and the expected output (Brownlee, 2019). Adversarial and L1 losses are combined to update the generator model. This causes the generator model to learn to create outputs that fool the discriminator and are close to the expected output.

5 Results With Test Dataset

The results of the network run on test data are shown (Figure 8) demonstrating that the network has learned to colourise the test images. Results are mostly correct with some loss in saturation and minor change in colour hues across the image. The network has been able to colourise features depicting large natural objects such as rocks, trees and water accurately including identifying reflected objects in water surfaces. Images taken in different seasons have also been colourised correctly with snow features being identified and coloured differently to exposed rock in images taken in warmer seasons.

The network has not been able to correctly colourise skies in some instances. Skies that are incorrectly coloured are less saturated than the real image and are being confused with surrounding rock colour in

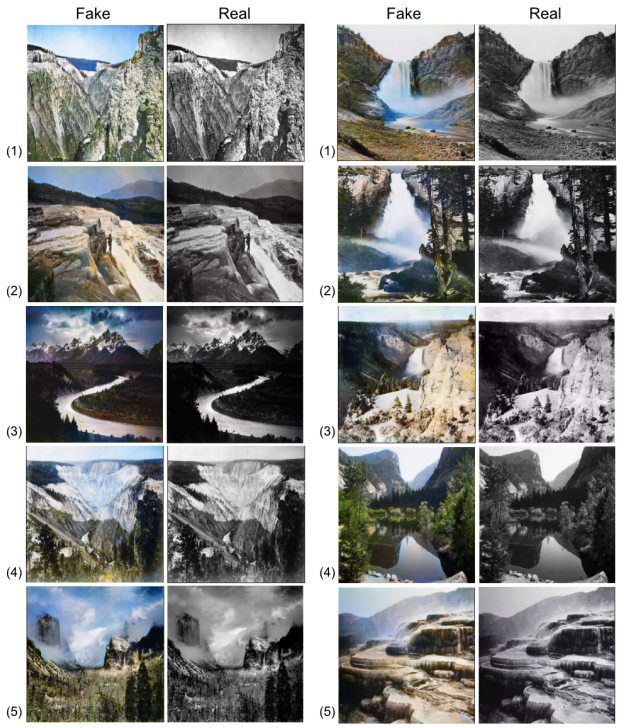

Figure 9: *Results from the Pix2Pix network used to colourise the works of Ansel Adams and William Henry Jackson. The table shows two columns of results with the generated fake image (left) and real original coloured image (right).*

some instances. This can be seen in image 2 and image 10 (Figure 8). There are less accurate results in noisy parts of images where there is a lot of detail that can be seen in image 8 (Figure 8).

These results could be improved by training the network for more epochs. This would allow the losses of the generate and discriminator networks to converge closer together, increasing the accuracy of the generated fake image. Furthermore, a larger training dataset with more images containing features that the network has produced less accurate results for will improve the accuracy of generated results as the network will have seen more examples of feature combinations that have been challenging.

6 Application

Some of the results of colourising the works of Adams and Jackson are shown (Figure 9). The full results are available at appendix A.

7 Conclusion And Suggestions For Future Work

A deep conditional convolutional generative adversarial network using the Pix2Pix implementation to solve an image-to-image translation problem of colourising black and white images. The network was trained with a bespoke data set consisting of paired black and white and coloured images of national parks in western North America. The network was tested with a similarly paired testing data set and applied to colourise the works of Ansel Adams and WIlliam Henry Jackson.

The network can be improved in future work by using a larger training dataset with a greater number of feature examples that will improve the accuracy of generated images. Training the network for a larger number of epochs will also improve results from the network by allowing losses from the generator and discriminator networks to converge further, producing more accurate generated results.

8 Bibliography

Brownlee, J. (2019) A Gentle Introduction to Generative Adversarial Networks (GANs). Machine Learning Mastery[blog]. 19 July. Available from: `https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/` [Accessed 17 December 2021].

Goodfellow, I J and Pouget-Abadie, J and Mirza, M and Zu, B and Warde-Farley, D and Ozair, S and Courville, A and Bengio, Y. (2014) Generative Adversarial Networks. Université de Montréal. Available from: `https://arxiv.org/abs/1406.2661` [Accessed 17 December 2021].

Radford, A and Metz, L and Chintala, S. (2016) Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. ICLR. Available from: `https://arxiv.org/abs/1511.06434` [Accessed 17 December 2021].

IBM (2021) Convolutional Neural Networks. Available from: `https://www.ibm.com/topics/convolutional-neural-networks` [Accessed 17 December 2021].

Brownlee, J. (2019) A Gentle Introduction to the Rectified Linear Unit (Relu). Machine Learning Mastery[blog]. 19 July. Available from: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/ [Accessed 19 December 2021].

Wood, T. (2021) Softmax Function. DeepAI[blog]. Available from: `https://deepai.org/machine-learning-glossary-and-terms/softmax-layer` [Accessed 19 December 2021].

Tiu, E. (2020) Understanding Latent Space in Machine Learning. Available from: https://towardsdatascience.com/understanding-latent-space-in-machine-learning-de5a7c687d8d [Accessed 19 December 2021].

Khan, A. (2021) GANs – Theory and Introduction in PyTorch. Available from:

https://medium.com/geekculture/introduction-to-the-gan-in-pytorch-bba920347b01 [Accessed 19 December 2021].

Saha, S. (2018) A Comprehensive Guide to Convolutional Neural Networks. Available from: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 [Accessed 19 December 2021].

Isola, P and Zhu, J-Y and Zhou, T and Efros, A A. (2018) Image-to-Image Translation with Conditional Adversarial Networks. Berkeley AI Research Laboratory. Available from: https://arxiv.org/pdf/1611.07004.pdf [Accessed 19 December 2021].

Zhu, J-Y. (2020) pix2pix. Available from: https://github.com/phillipi/pix2pix [Accessed 19 December 2021].

Sharma, A. (2021) Px2Pix:Image-to-Image Translation in PyTorch TensorFlow. Available from: https://learnopencv.com/paired-image-to-image-translation-pix2pix/#example1 [Accessed 19 December 2021].

Brownlee, J. (2019) A Gentle Introduction to Pix2Pix Generative Adversarial Network. Machine Learning Mastery[blog]. 19 July. Available from: https://machinelearningmastery.com/a-gentle-introduction-to-pix2pix-generative-adversarial-network/#: :text=Pix2Pix%20is%20a%20Generative%20Adversarial,presented%20at%20CVPR%20in%202017. [Accessed 19 December 2021].

Joshi, M R and Nkenyereye, L and Joshi, G P and Islam, R and Abdullah-Al-Wadud, M and Shrestha, S. (2020) Auto-Colorization of Historical Images Using Deep Convolutional Neural Networks. MDPI. Available from: https://www.mdpi.com/2227-7390/8/12/2258/htm [Accessed 19 December 2021].

Szegedy, C and Ioffe, S and Vanhoucke, V and Alemi, A. (2016) Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. Available from: https://arxiv.org/pdf/1602.07261.pdf [Accessed 19 December 2021].

Adobe. (2021) Photoshop (Creative Cloud) [computer program]. Available from: https://www.adobe.com/uk/products/photoshop.html [Accessed 23 December 2021].

Brownlee, J. (2019) A Gentle Introduction to Pix2Pix Generative Adversarial Network. Machine Learning Mastery[blog]. 29 July. Available from: https://machinelearningmastery.com/a-gentle-introduction-to-pix2pix-generative-adversarial-network/#: :text=Pix2Pix%20is%20a%20Generative%20Adversarial,presented%20at%20CVPR%20in%202017 [Accessed 23 December 2021].

The Ansel Adams Gallery (2021) The Ansel Adams Gallery. Available from: https://www.anseladams.com/ [Accessed 30 December 2021].

Britannica (2021) William Henry Jackson. Available from: https://www.britannica.com/biography/William-Henry-Jackson [Accessed 30 December 2021].

9 Appendix

Appendix A - Repository containing code, training datasets and results: https://github.com/MuelSB/AIFCT