Investigation into the use of Convolutional Neural Networks for the Purpose of Image Exposure Correction

u1921415

15th June 2022

Contents

1   Abstract

This report examines the implementation of a Convolutional Neural Network for image exposure correction. A Generative Adversarial Network (GAN) style network architecture is used and its loss functions analysed.

2   Introduction

Image exposure correction is a typical activity for photographers to adjust the light levels present in their photos. The exposure is set by the quantity of light let in during image capture. Adjustments are often made using image processing software such as adobe lightroom after the images are taken by the photographer. The image processing is a manual process that the photographer must perform for every image that they capture. This report explores whether a Generative Adversarial Network (GAN) could be used to generate well exposed images from a given under or over exposed image.

3   Network Architecture

The network used in this report was an adaptation of the pix2pix model which is proven to create accurate image transforms (Isola et al., 2016).

3.1   Overview

To achieve image translation from an image of poor exposure to one of more professionally exposed quality, a Generative Adversarial Network architecture was used. A GAN is a type of Convolutional Neural Network architecture with a unique construction. One of the ways in which this implementation varies from others is that the network receives an initial image and a target image. The aim of the network is to correctly hone its weights to consistently generate images or perform image transforms based on training cycles of those input/target paired images.

The 'generative' part of the GAN's structure is a generator model, whose role is to produce an output image from a given input image. On the other side of the GAN structure is the discriminator model which is trained to determine whether an image is real or fake. Here real can be considered to be straight from the dataset of target images, and fake being anything that has too high a loss value post evaluation. This architecture allows the discriminator to critique the generated images. The criticism is used to influence the generator model which reacts by updating its weights to attempt to create more lifelike images after every epoch.

3.2   Technical Implementation

To create new images from an input image, the generator takes advantage of a U-net style, encoder-decoder design. U-net is an architecture that effectively down samples and then up samples an image. It is a fully connected, convolutional neural net that takes advantage of stride convolution and max pooling. It also includes skip connections that make localized image information globally available in the symmetric model, increasing the predictive ability of the generator (Gayathri and Aarthy, 2021). The downscaling or encoding section attempts to gather context from the input whilst the decoder works to identify areas of interest during its upscaling (Ali et al., 2022).

In this implementation, an image of size 256x256 was shrunk down to a 1x1 feature map before being blown up again back into original size. This was performed by six layers down and six up which convolve the image. Stride was set at 2 to downscale and upscale the image. For activation functions, both ReLU and leaky ReLU are used. Using these activation functions over maxout activation was preferred since they have been found to train the model more quickly and work well with high resolution modelling (Radford et al., 2015).

For the discriminator, a PatchGAN was used. A PatchGAN is a comparably small array of values that represent whether a corresponding area or patch of an image is real or fake (Demir and Ünal, 2018). This relationship between patch and image can be visualised better in figure 1. By utilising this technique,

arbitrarily large images can be processed. Other benefits such as fewer parameters and faster runtime can be realised using this method (Isola et al., 2016).
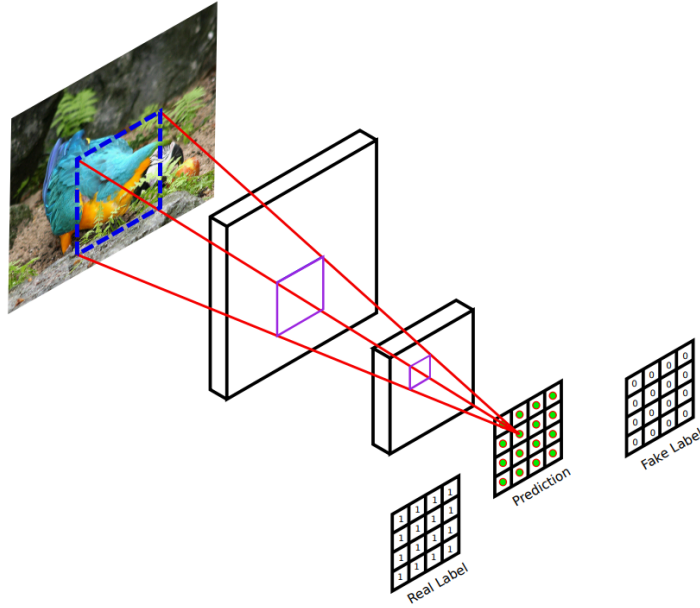

Figure 1: Demonstration of PatchGAN generation (Demir and Ünal, 2018)

4   Input Manipulations

To increase the variety of training data, augmentations were applied to the images. During training, a horizontal flip was applied at a probability of 0.5. Another transformation was the addition of colour jitter at a probability of 0.2 to increase the range of colours the model experiences. To aid in training, a normalisation transform was applied giving every training image a level of blur. The ground truth images however only received the same normalised blur but neither the colours nor the orientation augmentations since they are the target for the model to work towards.

(a) Only normalisation applied. (b) Image with added colour jitter. (c) Image with horizontal flip applied.

Figure 2: Examples of image augmentations on training images.

5   Loss Functions

The nature and architecture of convolutional neural network implemented alongside this report required the use of two loss functions, one for the generator and one for the discriminator. For the discriminator, a Binary Cross Entropy (BCE) loss function is used however, for the generator an L1 loss function is chosen.

5.1   Binary Cross Entropy

Binary cross entropy is an especially useful loss function in the field of classification. This makes it ideal for training the discriminator since it will help to classify an image into the categories of real and fake or class 1 or 0 respectively (Ketkar, 2017). The function works by taking in a probability that an image is real from the generator and a target state. The target state in this case is 1 representing the class real since the generator is trying to output real images and the discriminator should feedback to what extent it believes the image it creates really is. The result of Binary Cross Entropy loss is the sum of probabilities showing how different the input images are compared to what is expected i.e. A real image (Michelucci, 2019).

5.2   L1 Loss

L1 loss or the absolute error loss is the absolute difference between the actual value and a prediction. This loss is calculated for every image in the dataset compared to the image the generator has made. Adding the L1 loss function as a penalty to the generator forces the generator to produce results closer to the ground truth images. In this implementation, L1 loss is chosen over L2 loss since L1 loss has been found to produce less blurry generated images (Sutanto and Kang, 2021).

6   Training and Testing Process Followed

Included in the training dataset were 17,675 input images with varying exposure and corresponding ground truth images, where there were 5 images of varying exposure to every ground truth image. The images were rendered using Adobe Camera Raw to emulate various exposure issues. The images use a relative exposure from -1.5 for very underexposed images through to +1.5 for ones that are artificially overexposed (Afifi et al., 2020), with the full range being -1.5,-1,0,1 and 1.5. Also available were 750 validation photographs following the same format as the training set. For each epoch, the model consumed all input and ground truth images from the training set. Inference was also performed on a validation image every epoch to visualise the performance of the model as it learned.

The model was trained from scratch for 150 epochs to produce the results available in the model evaluation. A batch size of 64 was used to take advantage of GPU parallelism to increase the speed of training.

7   Model Evaluation Metric/s used

For model evaluation, the loss function for the generator is plotted in figure 3a over 80 epochs of training. The graph shows a clear reduction in loss function output over the course of training therefore we can conclude the model is learning efficiently and an improvement should be expected in the ability of the generator to make higher quality images. In figure 3b, the loss over the course of one batch forms a downward slope from an initially high loss to a steady loss which backs up the conclusion that the model is training effectively.

(a) Generator loss over a number of epochs
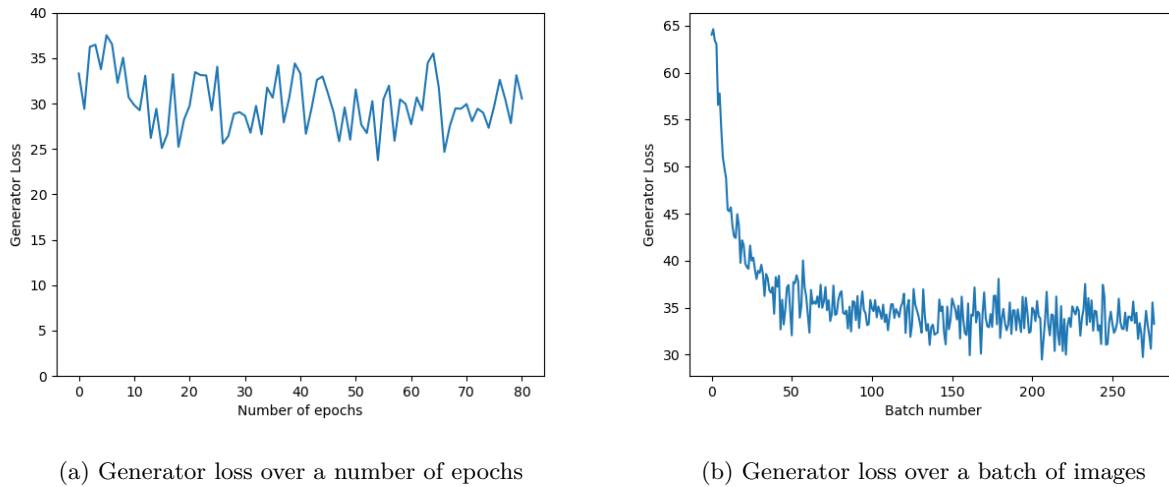
(b) Generator loss over a batch of images

Figure 3: Plots of Generator loss

Whilst experimenting with the learning rate of the model, an improvement in performance could be seen from decreasing the learning rate from 0.0002 to 0.00008. By decreasing the learning rate, the weights of the model were changed less drastically over each epoch, preventing it from passing over the local minima. Figure 3b also justifies our choice of learning rate since the model learns at a steady rate over each epoch before settling to a lesser gradient. Future experimentation could be performed with the parameters of the model to decrease the loss over the number of epochs to further improve the model and increase training speed.

8   Analysis and Conclusions

The performance of the model can be seen in the example images below. In general, the model can successfully correct the brightness in the images it evaluates. In figure 4 the model has taken the image from totally overexposed to close to its respective ground truth image. Similarly, when the image is very under exposed like in figure 5, the brightness has been raised whilst maintaining pretty good colour accuracy. Despite this, the model can struggle with artifacts being left in images which depreciates the quality of its output. In figure 6, the model manages to decrease the exposure but the centre of the image experiences artifacts. With additional training time, we should expect the model to reduce these artifacts and thus increase its ability to generalize exposure correction.

(a) Input image   (b) Generated image   (c) Ground truth image

Figure 4: Example images using photo of a car

(a) Input image   (b) Generated image   (c) Ground truth image
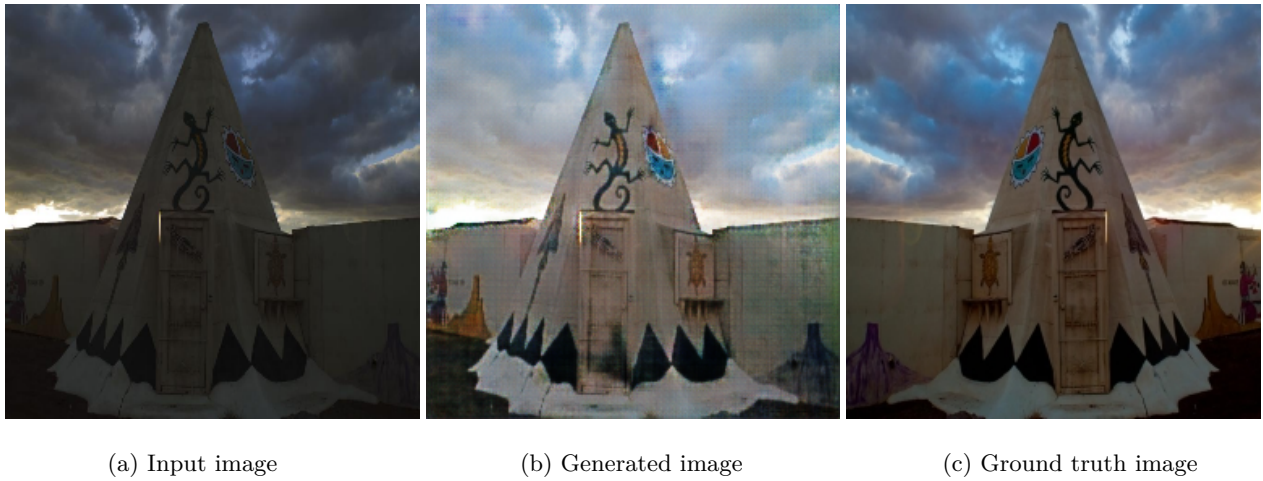
Figure 5: Example images using photo of a tent

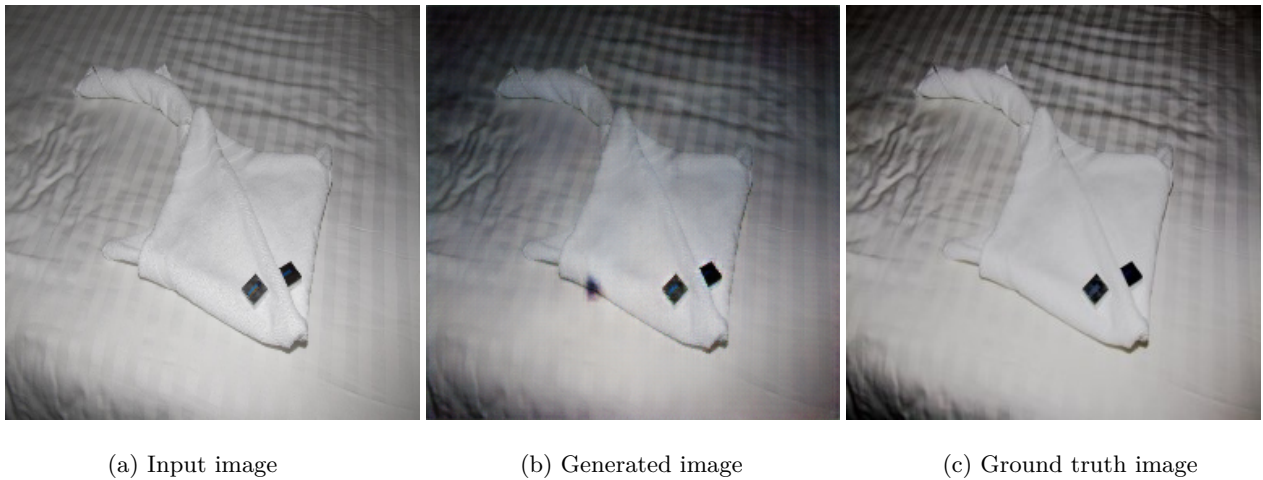(a) Input image   (b) Generated image   (c) Ground truth image

Figure 6: Example images using photo of a towel

9   Description of possible alternative approaches

The problem of poor image exposure can be approached in a few other ways. One method is to use a more basic version of image processing. By creating a 3x3 convolution kernel that takes an average of the pixels adjacent to the central pixel, it would be possible to brighten or darken an image depending on its exposure in post processing. Another method would be to continue to process the images manually using photo editing software. However, both alternative approaches require a manual application to the image to correct its exposure. Only the convolutional neural network approach with the aid of a Generative Adversarial Network architecture would be able to correct image exposure without the input from the photographer making the implementation documented in this report a very exciting step in image processing.

Word Count: 1631 words

References

Afifi, M., Derpanis, K.G., Ommer, B., Brown, M.S., 2020. Learning multi-scale photo exposure correction. https://doi.org/10.48550/ARXIV.2003.11596

Ali, O., Ali, H., Shah, S.A.A., Shahzad, A., 2022. Implementation of a modified u-net for medical image segmentation on edge devices. IEEE Transactions on Circuits and Systems II: Express Briefs 1–1. https://doi.org/10.1109/tcsii.2022.3181132

Demir, U., Ünal, G.B., 2018. Patch-based image inpainting with generative adversarial networks. CoRR abs/1803.07422.

Gayathri, S., Aarthy, D.K., 2021. Computer-aided detection of malignant mass in mammogram using u-net architecture, in: Advances in Intelligent Systems and Computing. Springer Singapore, pp. 179–185. https://doi.org/10.1007/978-981-16-2674-6_14

Isola, P., Zhu, J., Zhou, T., Efros, A.A., 2016. Image-to-image translation with conditional adversarial networks. CoRR abs/1611.07004.

Ketkar, N., 2017. Feed forward neural networks, in: Deep Learning with Python: A Hands-on Introduction. Apress, Berkeley, CA, pp. 17–33. https://doi.org/10.1007/978-1-4842-2766-4_3

Michelucci, U., 2019. Cost functions and style transfer, in: Advanced Applied Deep Learning : Convolutional Neural Networks and Object Detection. Apress, Berkeley, CA, pp. 161–193. https://doi.org/10.1007/978-1-4842-4976-5_5

Radford, A., Metz, L., Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. https://doi.org/10.48550/ARXIV.1511.06434

Sutanto, A.R., Kang, D.-K., 2021. A novel diminish smooth l1 loss model with generative adversarial network, in: Singh, M., Kang, D.-K., Lee, J.-H., Tiwary, U.S., Singh, D., Chung, W.-Y. (Eds.), Intelligent Human Computer Interaction. Springer International Publishing, Cham, pp. 361–368.