SICGAN - Single Image Colourisation using Generative Adversarial Networks

1 Problem Definition

This project will investigate if the colour information of an image, represented in the LAB colour space, can be predicted from lightness and local geometric features. Specifically, the model predicts the A and B channels of the LAB image. To learn local features, this model will use a patch based approach where it randomly samples smaller patches of the image to apply some data augmentations to generate a training set. The input data tensor is denoted as $\mathbf{U} \in \mathbb{R}^{p \times p_h \times p_w \times 5_F}$, where P is the number of patches, p_h and p_w are the height and width of each patch respectively. F is a $p_h \times p_w \times 5$ dimensional array for each pixel in the patch including the L channel, the edge magnitude calculated using the Sobel operator on the L channel, the edge density calculated using the Laplacian operator on the L channel, and normalised local coordinates x and y within the patch. The ground truth colour output tensor is denoted by $\mathbf{V} \in \mathbb{R}^{p \times p_h \times p_w \times 2}$, containing the A and B channels to be merged back with the original L channel. A Generative Adversarial Network (GAN) is used to model this task, consisting of a Generator G that maps inputs to predicted lightness values $\hat{\mathbf{V}} = G(\mathbf{U})$, and a Discriminator D, training to distinguish between real and fake generated patches. This research is important because if successful, this would show that a model can learn to infer meaningful and realistic colour distributions from localised structuresand very limited global context. This has useful applications in fields such as image colourisation for restoring historical photographs, or enhancing medical imaging. Colourisation is not determinate, the same edge and lightness structure could have multiple possible colourings. So, success is not well represented by quantitative measures and rather qualitative measures such as visual coherence are better. Solving this problem in a coherent and plausible way is an interesting deep learning task.

2 Method

Each input patch (the method to create this is explained in the results section) P is converted from RGB to LAB space. The features in the array F for each patch are calculated as follows. Lightness: $f_1(x, y) = L(x, y)$, Sobel edge magnitude: $f_2(x, y) = \sqrt{(\partial_x L)^2 + (\partial_y L)^2}$, Edge density: $f_3(x, y) = \nabla^2 L(x, y)$, Local normalised coordinates: $f_4(x, y) = \frac{x}{p}$, $f_5(x, y) = \frac{y}{p}$.

So each patch becomes a tensor $\mathbf{u} \in \mathbb{R}^{p_h \times p_w \times 5}$ and the label is $\mathbf{v} \in \mathbb{R}^{p_h \times p_w \times 2}$, where $\mathbf{v}(x, y) = [A(x, y), B(x, y)]$. The generator G maps a 5-channel input tensor $\mathbf{u} \in \mathbb{R}^{p_h \times p_w \times 5}$ to a 2-channel output $G(\mathbf{u}) \in \mathbb{R}^{p_h \times p_w \times 2}$ representing the A and B channels:

$$G : (L, E_{Sobel}, E_{Lap}, x_{norm}, y_{norm}) \rightarrow (A, B)$$

The generator consists of a stack of convolutional layers with ReLU activations. The discriminator D attempts to distinguish between real and generated A, B patches. It accepts a 2-channel image $\mathbf{v} \in \mathbb{R}^{p_h \times p_w \times 2}$ and outputs a scaler $D(\mathbf{v}) \in \mathbb{R}$ that indicates the likelihood of an input being a real patch:

$$D : (A, B) \rightarrow \mathbb{R}$$

The model is fully convolutional with downsampling via adaptive average pooling and a final dense layer for classification. Instance noise helps regularise D during training, this helps mitigate mode collapse and overfitting. The GAN is trained in a minimax game and we use label smoothing and noisy targets to further stabilise training. The discriminator is optimised with the following loss:

$$\mathcal{L}_D = \mathbb{E}_Y[BCE(D(Y + \epsilon), 0.9 + \delta] + \mathbb{E}_X[BCE(D(G(X) + \epsilon), \delta')]$$

where $e \sim N(0, \sigma^2)$ is instance noise, and δ, δ' are small random mutations for label smoothing, and BCE is the binary cross entropy loss function. Conversely, the generator is trained to maximise the probability of its outputs being classified as real:

$$\mathcal{L}_G = \mathbb{E}_X[BCE(D(G(X)), 1 + \delta'']$$

where δ'' is a small mutation added to the labels.

3 Results

To obtain a test set, we use patch sampling and data augmentation on the original image. Let $P_k \in \mathbb{R}^{p_h \times p_w \times 3}$ be a patch sampled from the image I. We sample n such patches with a minimum distance constraint:

$$\forall i \neq j, ||c_i - c_j||^2 \geq d_{min}$$

where c_i is the origin coordinate for the i-th patch. This encourages sampling to be more uniform to avoid accidental oversampling certain regions in the image. Each patch P_k is augmented by a group of transformations:

$$\mathcal{T} = \{id, flip_x, flip_y, rot_{90}, rot_{180}, rot_{270}\}$$

So each raw patch generates $\{T(P_k)|T \in \mathcal{T}\}$.

The training details of the experiment was to to train the model over 20 epochs, using 5000 patches, a patch size of 128x128, the Adam optimiser with learning rates of 1×10^{-4} for the generator and 4×10^{-4} for the discriminator, and $(\beta_1, \beta_2) = (0.5, 0.999)$.

The results of this were successful, measuring this quantitatively proves difficult as colourisation is a one to many problem, and consequentially the losses are expected to appear high, see Table 1. Figure 1 shows that over time the loss of the generator did improve, however the discriminator did not appear to improve over time. This could be due to insufficient amounts of training data, but as previously mentioned given that colourisation is a one to many problem, the discriminator may struggle quite a bit with this task. However, observing Figure 2 - the generated image for these given losses - as well as Figure 3 - another run of the program with the same settings, shows that the model is producing a visually coherent response, which is a success.

4 Reflection

This research has a larger impact on people outside of researchers. This has applications in the fields of image colourisation. It could be used to help recolour old photographs or even be used as part of tools that could suggest or apply plausible colours to enhance workflows. However, there are ethical considerations. Viewers of recolourised media may not be aware that it is not the original and could mistake it for accurate historical data. As the model is trained on a single image, this means that it may not extrapolate well if a trained model is transferred to predict colour in a different domain. This could result in a misrepresentation of the original cotents of the image. Human oversight would be necessary to ensure results are not misleading and generated colourisations are transparently labelled.

5 Conclusion

This research has found that it is possible to generate a coherent and plausible colourisation of an image based on lightness and localised geometric context. There is some overtuning to specific colour regions in the image. Given more time, this research would have attempted to optimise the training process and upgrade hardware so that it could handle more input data, and see if this improved the overtuning. It would also investigate the geometric parameters of the input tensor to identify how effective they were and if other choices would have proven better. This research is a success, the model clearly learned a visually coherent and plausible solution.

Table 1: Discriminator and Generator Losses at Batch 1 and Batch 101 (/157) for Each Epoch

Epoch	D Loss (Batch 1)	G Loss (Batch 1)	D Loss (Batch 101)	G Loss (Batch 101)
1	0.9677	1.0073	1.3735	0.7032
2	1.3697	0.6303	1.3599	1.1640
3	1.3465	0.7036	1.3182	0.7057
4	1.3401	0.7056	1.3345	0.6782
5	1.3466	0.7062	1.3263	1.1546
6	1.2796	0.7485	1.3243	0.5882
7	1.3567	0.7052	1.3827	0.7827
8	1.3539	0.8261	1.3473	0.7878
9	1.3548	0.7964	1.3323	0.8061
10	1.3190	0.7568	1.3402	0.7689
11	1.2765	0.8591	1.4008	0.8026
12	1.4041	0.8298	1.2873	0.8714
13	1.3923	0.6315	1.2622	0.6247
14	1.3888	0.8126	1.3688	0.8094
15	1.3817	0.8174	1.3841	0.7972
16	1.3783	0.8028	1.3929	0.8598
17	1.3775	0.8354	1.3602	0.7763
18	1.3213	0.6257	1.4734	0.6861
19	1.3908	0.8319	1.3515	0.7404
20	1.3758	0.8048	1.3609	0.8050

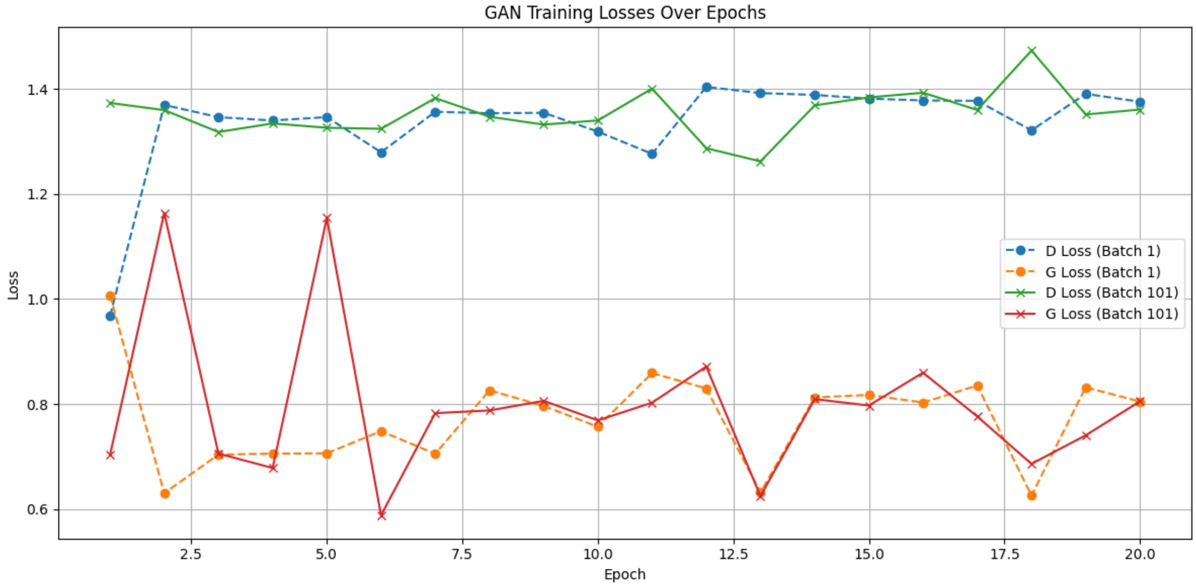

Figure 1: Visualisation of Discriminator and Generator Losses at Batch 1 and Batch 101 (/157) for Each Epoch

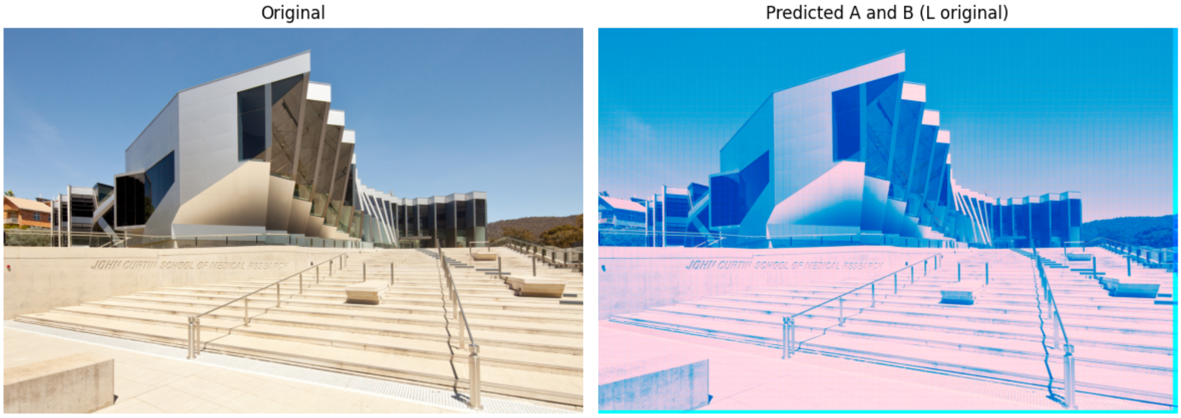

Figure 2: Generated Image 1

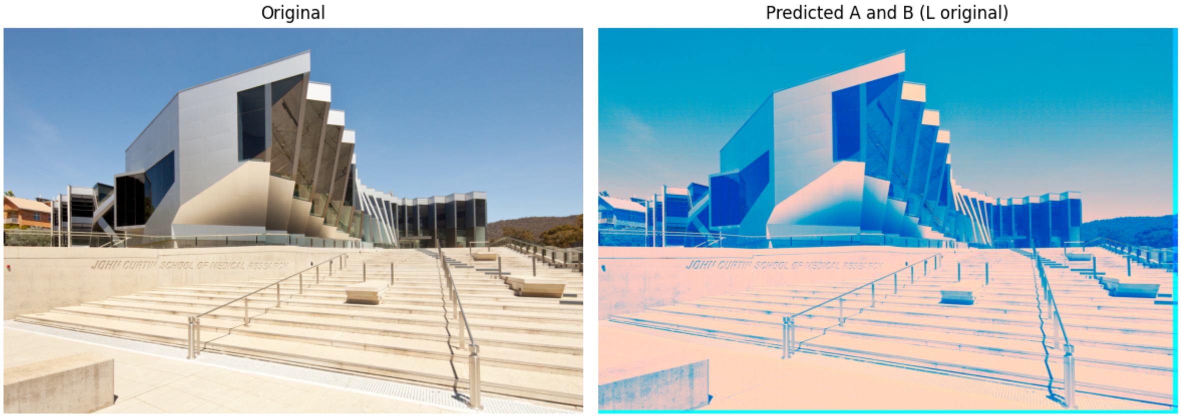

Figure 3: Generated Image 2