Indian Institute of Technology Patna

CS299 - Innovation Lab

Endterm Report

AutoTone - Auto colourization of Manga

Team:	*Roll No.*
Mayank Vaidya	1601CS22
Mukuntha N S	1601CS27

Contents

ABSTRACT

In this project, we aim to propose a method to colourize manga art automatically, using a deep learning model. Manga is a style of Japanese comic books and graphic novels, typically aimed at adults as well as children. Manga art differs from typical Western comics in the level of detail and storytelling. Manga unlike comics, are rarely coloured. This is owing to their high release frequencies, and the high cost of labour, skill and time involved in the manga colourization process. Since the manga industry is a billion dollar industry, a perfected system for colourization has great market potential.

OBJECTIVE

Our aim was to build a model to automatically colourize manga line-art, and thus propose a method to solve a major industry problem. We aimed to implement and put to use the latest in colourization techniques for this task.

CURRENT SOLUTIONS

The problem of general image colourization is an active research area. Most work in the area though, has relied on significant user input, or has produced desaturated images. Recently, Zhang et al. (2016) and Larsson et al. (2016) proposed fully automatic colourization approaches that produces vibrant and realistic colourizations, given gray-scale images as the input. Our problem is more complex, given how our input to the system consists only of line drawings, which would not contain the ‘lightness’ channel for each pixel.

There have also been other attempts at the colourization problem, such as Stephen Koo(2016), which had the use of the more recent Generative Adversarial Network (GAN) model (Goodfellow et al 2014).. The manga colourization problem is far less explored, but attempts have been made. PaintsChainer is one product which attempts to solve this, but needs significant user-input and produces desaturated output images.

NOVELTY

The manga colourization problem is very new, and is yet to be solved properly. Current models rely on high levels of user interaction, which we aim to remove, and make the system fully automatic. We also aim to solve the problem of desaturation, with a new classification based approach, with a modified loss function from recent work in general image colourization from Larsson et al. (2016). Our proposed method can be scaled to any input size.

METHODOLOGY

Introduction

Our approach to solving the problem is heavily based on the methods proposed by Larsson et al. (2016). Since manga art usually appears according to multimodal colour distributions, we train our model to predict per-pixel colour histograms. Since the colourization problem depends on the model's semantic understanding of the image being coloured, a deep Convolutional Neural Network (CNN) based system was used. Since this is an image-to-image prediction problem, we trained a conditional Generative Adversarial Network (cGAN) with an additional loss function based on the predicted colour histograms. In our Generator, we have made use of the popular U-Net architecture, which has been shown to best regular CNNs and also brings with it speedy computation. Due to its residual connections, the size of the network is reduced drastically, thus saving memory.

Colour Spaces and Histograms

Since our input images consist of line-art, they only contain black/white information per pixel and lack the 'Lightness' channel (L) that grayscale images would contain. We treat the colour information separate, considering the Hue/chroma colour space, as this has been shown to be reasonably successful in colourization problems. Both HSV and HSL are popular colour spaces, but face instabilities. To retain L as a channel, and to avoid instabilities, we use the HCL colour space, where Chroma (C) takes the place of Saturation (S). Conversion to HSV is given by:

$$V = L + \frac{C}{2}$$

$$S = \frac{C}{V}$$

We treat the Lightness prediction as a regression problem. Our non-adversarial loss function for the Lightness prediction was the L2 regularization loss:

$$Loss_L = \|L_{pred} - L_{real}\|^2$$

We treat the Hue & Chroma prediction task as a histogram estimation task (similar to multiclass classification) rather than as regression problem. The problem, when posed as a regression problem, causes desaturation in the output images, owing to its averaging-out effect on all applicable colours for a particular object. We split the Hue and Chroma into equally sized bins for the histogram. Experimentally, the bin size was set to 32. The model outputs marginal probability distributions (or histograms, $f(x)$) for the hue and chroma channels per pixel. For histogram predictions, the last layer of neural network f is always a softmax. Our non-adversarial loss function for the Hue and Chroma has been taken from Larsson et al. (2016) and is given by,

$$L_{\text{hue/chroma}}(\mathbf{x}, \mathbf{y}) = D_{\text{KL}}(\mathbf{y}_{\text{C}} \| f_{\text{C}}(\mathbf{x})) + \lambda_H y_{\text{C}} D_{\text{KL}}(\mathbf{y}_{\text{H}} \| f_{\text{H}}(\mathbf{x}))$$

where D_{KL} is the KL-divergence, $f_c(x)$ is the chroma histogram and $y_h(x)$ is the Hue histogram. $\mathbf{y}_H$ and $\mathbf{y}_C$ are one-hot vectors corresponding to the actual value of the hue and chroma, and y_c is the real value of the sample pixel's chroma. Due to $\mathbf{y}_H$ and $\mathbf{y}_C$ being one-hot vectors, the KL divergence can be replaced by softmax loss function. The value of λ_H was set to 5, roughly the inverse expectation of y_c (as set by Larsson et al.), thus equally weighting hue and chroma.

Similar to Larsson et al., the inference for the colour from the histogram was taken as the expectation of the histogram for the Hue, and the median of the histogram for the Chroma.

Model Architecture

This being an image generation problem and due to the recent success of GANs in generating realistic images, we decided to use a conditional Generative Adversarial Network (cGAN) conditioned on the input line-art images.

Generator

We made use of the popular U-Net architecture in our Generator. The generator takes as input the line-drawing of size w × h × 1. The generator contains 5 convolutional layers. Convolutions were done by a '5 × 5' convolutional filter, using a stride of 2 units. Once dense feature matrix of size w/32 × h/32 × 512 is reached, transpose convolution operations are performed, with residual connections between layers (like in the U-Net architecture) as shown. The output is the concatenated per-pixel histograms for Hue (w × h × 32), the histograms for Chroma (w × h × 32) and the Lightness (w × h × 1.). All outputs of the generator are obtained after a softmax operation.

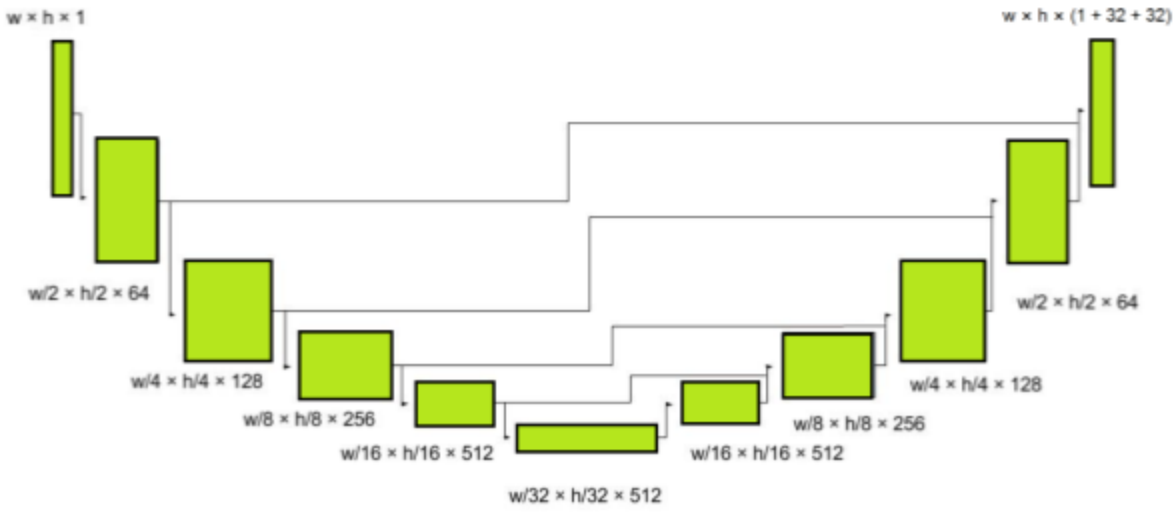

Figure : Generator structure (based on U-Net Architecture)

Since the generator is completely based on Convolutional and Transpose Convolutional layers, the input size of the generator can vary. This enables us to train and test on images of different sizes. The only constraint is that the height and width of the image have to be divisible by 32, which can easily be fixed with zero-padding.

Discriminator

The discriminator classifies images into ‘real’ and ‘fake’ . It is trained on both the generated image and the real image. It returns a number close to 1 if it is convinced that the image is real, and a number close to 0 otherwise. We represented the discriminator as a stack of convolutional layers, along with one fully connected layer at the end.

The dimensions of the input layer was ‘w × h × 4’, w being the width and h being the height of the input image. The input image was concatenated with hue(H), chroma(C) and lightness(L) predictions.

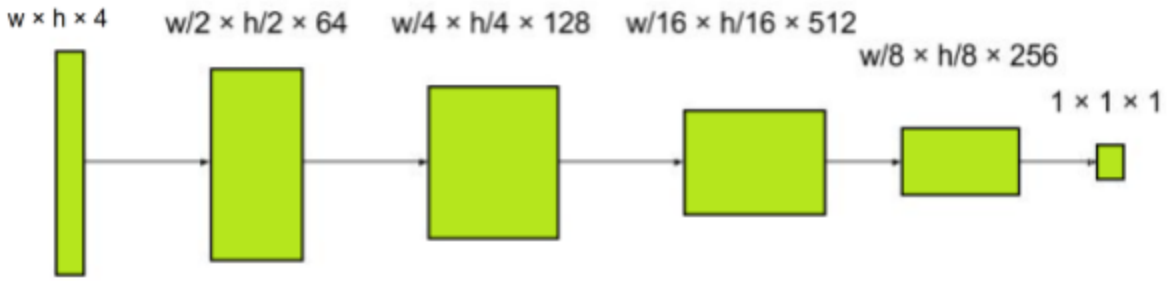

Figure: Discriminator structure

Convolutions were done by a ‘5 × 5’ convolutional filter, using a stride of 2 units. The final output was obtained after a softmax operation.

EXPERIMENTS AND RESULTS

Dataset

Having searched the Internet to the best of our ability, we couldn’t find a dataset of line art and their corresponding coloured images. Hence, we generated the dataset by processing coloured manga art into line art. This was done using the Adaptive Thresholding algorithm provided by OpenCV, and fine tuned to produce real line art-like images. In simple thresholding, the threshold value is global, i.e., it is same for all the pixels in the image. Adaptive thresholding is the method where the threshold value is

calculated for smaller regions. This method performs well in most cases, except when the image is heavily textured where it leaves some unwanted marks.

Around 20,000 images were downloaded using a public API provided by Safebooru, of which 18,000 were used for training. These were then processed as mentioned above and saved for further use.

Experiments

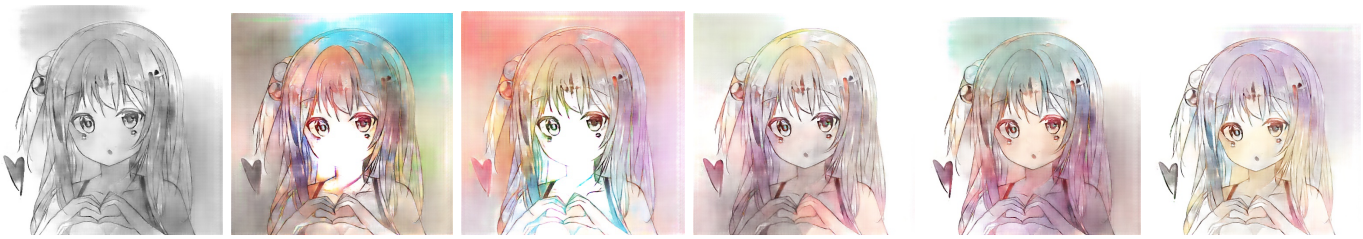

Figure: Colourized outputs after 2, 4, 6, 8, 10, 12 epochs respectively.

For our experiments, we used the Tensorflow library from Google, and OpenCV for most image processing tasks. The generator was trained to optimize the weighted sum of the adversarial loss and the non-adversarial losses (for Lightness(L), Hue(H) & Chroma(C)) as discussed in the 'Colour Spaces and Histograms' section. A weighing of 1 : 100 was used for the adversarial and the non-adversarial losses. The discriminator was trained on only the adversarial loss. The Adam Optimizer was used for training.

The dataset was split into batches of size 4 for training. For every batch, the discriminator and then the generator are trained separately. Since the generator often lagged behind the discriminator in training, the discriminator was trained only every alternate epoch, that too only if the average loss was above 0.2. The images were trained on, for a total of 12 epochs, which took about 30 hours on a NVIDIA Tesla K80 GPU.

Some of the colourized outputs, after every even epoch was completed, is shown in the figure above. Initially, for the first two-three epochs, the colour outputs were not visible at all, but this improved as soon as the shading outputs improved over training.

Results and Analysis

Figure: (Clockwise from top left): Original coloured image; Processed line art; Final coloured output; Shaded output (L)

Some of the results obtained by our method are shown in the figures in this section. Since the problem has been posed as a multi-class classification problem, any of the several applicable classes (colours) for the output have been selected by the model. (For example, the colour of the hair in the input image above is Blue, but the model has chosen an equally applicable colour Yellow for the output).

Figure: (from left to right) Original coloured image; Processed line art; Shaded output (L); Final coloured output

Although one of our original aims was to solve the problem of desaturation, it is still starkly visible that this problem has not been solved. The shaded outputs obtained from the Lightness channel (L) show that shading is overall realistic. The colours in the outputs are often ‘patchy’ and ‘leaky’, carrying a water-washed effect. The generator often also gets confused between colours when colouring the same object, giving rise to multi-coloured hair, eyes, etc. Heterochromia (differently coloured eyes) is often observed, since several downloaded images contained the tag ‘heterochromia’, it being popular in manga art.

Figure: These images show multi-coloured hair and the generator's confusion. Also, heterochromia is observed in the image on the left. This would've manifested as a highly desaturated image with previous methods.

CONCLUSION AND SCOPE FOR FUTURE WORK

We have presented a novel manga colourization technique, involving a conditional Generative Adversarial Network with a U-Net based Generator, which can scale to any input size. The U-Net architecture is also highly memory efficient due to their residual connections, and is faster than traditional stacked CNN models. Our outputs are unique, and partially solve the desaturation problem. But we also face the problem of having patchy, differently coloured outputs. Several images also have a water-washed effect, with leaky colours.

For future work, it might help to build two separate models for shading (Lightness prediction), and colourization (Hue & Chroma prediction). Also, experimentation has only been done with the HCL colour space. The LAB colour space has also shown to be extremely useful in colourization problems, and might yield better results. Overall, we hope this work contributes to future attempts at solving the line art colourization problem.

Bibliography

Larsson, Gustav, et al. “Learning Representations for Automatic Colorization.” *Computer Vision – ECCV 2016 Lecture Notes in Computer Science*, 2016, pp. 577–593., doi:10.1007/978-3-319-46493-0_35.

Ronneberger, Olaf, et al. “U-Net: Convolutional Networks for Biomedical Image Segmentation.” *Lecture Notes in Computer Science Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 2015, pp. 234–241., doi:10.1007/978-3-319-24574-4_28.

Zhang, Richard, et al. “Colorful Image Colorization.” *Computer Vision – ECCV 2016 Lecture Notes in Computer Science*, 2016, pp. 649–666., doi:10.1007/978-3-319-46487-9_40.

Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.