Journal of Information Technology Management

Print ISSN: 2008-5893

Online ISSN: 2423-5059

Mapping Grayscale Images to Colour Space Using Deep Learning

Anu Saini*

*Corresponding author, Assistant Professor, Ph.D., Department of Computer Science and Engineering, G. B. Pant Govt. Engineering College, New Delhi, India. E-mail: anuanu16@gmail.com

Jyoti Tripathi

Assistant Professor, Department of Computer Science and Engineering, G.B. Pant Govt. Engineering College, New Delhi, India. E-mail: loginjyoti@gmail.com

Abstract

People are used to exploring grayscale images in their family albums but it is difficult to grasp the reality without colours. Luckily, with advancements in Machine Learning it has been possible to solve problems previously thought impossible. The authors aim to automatically colourize grayscale images using a subset of Machine Learning called Deep Learning. The system will be trained on an image dataset and given an input grayscale image the model will be able to assign aesthetically believable colours. A grayscale photograph has been provided; our approach solves the problem of visualizing a reasonable colour version of the grayscale picture. This issue is undoubtedly under controlled; therefore earlier methods to this problem have either counted majorly on user interaction or it leads to in unsaturated colourizations. The authors put forward a completely automatic approach that will try to produce realistic and vibrant colourizations as much as possible. The proposed system has been applied as a feed-forward in a Convolutional Neural Network and has been trained on over twenty thousand colour images currently.

Keywords: Convolutional Neural Networks (CNN), Convolution, RGB, CIELAB (Lab), Deep Neural Networks, Feature vector, Prediction, Sampling

DOI: https://10.22059/jitm.2022.85649 Manuscript Type: Research Paper University of Tehran, Faculty of Management

Received: December 12, 2020 Accepted: March 10, 2021

Introduction

Automatic colourization of grayscale pictures has always been an issue of research. Instead of simply being interesting from features and machine learning perspective, this technique has wide range which varying from restoration of videos to enhancement of images for better understandability of data.

During past years, CNNs have been recognized as the concrete standard for reducing the error rates to less than four percent and image classification problems in the “ImageNet challenge”. CNNs are indebted for majority of their success to their capability to learn and recognize colours shapes and patterns inside pictures and relate them through corresponding object classes. These features make CNNs a good choice for colourizing images since patterns, object classes and shapes generally complement with choice of colours.

In computer vision and graphics, mainly two approaches to grayscale image colourization exist: data-driven automatic colourization and user-guided edit propagation. A user is required to draw coloured marks over input grayscale image (Levin and et al. 2004). A colourized image is then generated which matches the scribbles of user, while maintaining image priors. Impressive results can be achieved through these methods but the amount of user interaction required can be huge, as each distinctly coloured region needs to be marked by the user explicitly. Since the system depend on on user marks colour distinction, even the region containing high colour distinction, such as green vegetation and plants, need to be explicitly stated.

In order to counter the limitations in above approach, more data-driven colourization methods have been discovered in consonance with the demand. These data driven methods colourize a grayscale picture in anyone of 2 methods: firstly by provision of matching given image to another prototype colour photograph in the data set. Secondly, non-parametrically “stealing” colours from the exemplar image, an approach which relates to the concept of Image Analogies, or by learning specific “parametric mappings” from grayscale image to colour image from large scale database. DNN and are fully automatic in operation (Zhang and et al. 2016), (Iizuka and et al. 2016). Even though this makes image colourization less costly, the results are often incorrect. Basically, the colour of an object, say for example a cloth, is often inherently confusing– it may be green, red or blue. Currently available automatic methods mainly choose a single colourization, and user is not allowed to mention their choice for an acceptable version.

Literature Review

User-guided colourization: Earlier work on colourization mainly focused on local controls, like user marks for interactive experience (Huang and et al. 2005) , (Levin and et al. 2004). In order to reduce efforts of user, the methods which were developed later focused on designing better similarity and utilizing long range connections. Learning machinery, like feature discrimination (Xu and et al. 2013), local linear embeddings (Chen and et al. 2012), boosting (Li and et al.

2008), and in this era, NN (or simply neural nets), has been expected to study similarity between pixels automatically given user marks and input image set. Nowadays popular methods of expressive global control have emerged, such as colour palette (Chang et al., 2015) and variation of colour intensity (Li and et al. 2015, Wang and et al. 2010), furthermore to the local controls. Concurrently, a system for translation of sketches to real images has also been developed by (Sangkloy and et al. 2017), with comprehensive support for user colour marks.

Automatic colourization: Semi-automatic methods (Chia and et al., 2011, Liu and et al., 2008, Welsh and et al., 2002, Irony and et al., 2005, Gupta and et al., 2012) employ an exemplar method that transfers colour statistics from multiple images or a single reference image (Liu and et al., 2014) to the grayscale picture with use of methods (Hertzmann and et al., 2001) and colour transfer. When the inserted image and the corresponding reference share alike content, these methods perform remarkably well. However, the process of finding reference pictures is a task which takes time, and it can be even more challenging for complex scenes or rare objects, even if we use semi-automatic retrieval methods. Additionally, tedious manual efforts are involved in some algorithms on specifying corresponding regions between different images.

In recent times, some methods have been proposed which are fully automatic (Deshpande and et al., 2015), (Cheng and et al., 2015), (Iizuka and et al., 2016), (Zhang and et al., 2016), (Larsson and et al., 2016), (Isola and et al. 2017). These recent methods train CNNs on image collections at a large scale to straightly correlate grayscale pictures to colourized ones as output. Combination of low and high-level cues can be learned by networks to perform the task of colourization, and the produced results were realistic, as examined in human decisions (Zhang and et al. 2016). However, production of a single plausible result is the ultimate aim of these approaches, albeit colourization is essentially an ill presented problem, with multi-modal uncertainty.

Deep-Semantic image manipulation: According to (Zhou and et al., 2014) DNN (or simply neural networks) are exceptionally good at extraction of meanings from images, from mid-level concepts to high-level information such as scene categories and objects. Various image processing tasks for which neural networks were deployed, such as enhancement of images (Yan and et al., 2016), oversimplification of sketches (Simo and et al., 2016), in painting, de-noising, image blending and style transfer.

Problem Description

Muse upon the grayscale images in Figure 1. At the first glimpse, predicting the colours seems to be a formidable task, as much of the details have been lost. If inspection is done closely, one can however notice that in many images, its surface texture and the semantics of the scenery provide substantial clues for numerous parts in each picture: the grass has inherently green colour, the ladybug is mostly of red colour, and the sky presumably has to be blue. Naturally, these types of semantic guesses are not applicable everywhere. In this project, our goal is to produce a believable colourized image rather than recovery of the actual ground truth colour. Thus, our task translates to become more feasible: to prototype required dependencies among the textures and recognizable semantics of grayscale image then its colour version so that visually compelling results can be produced.

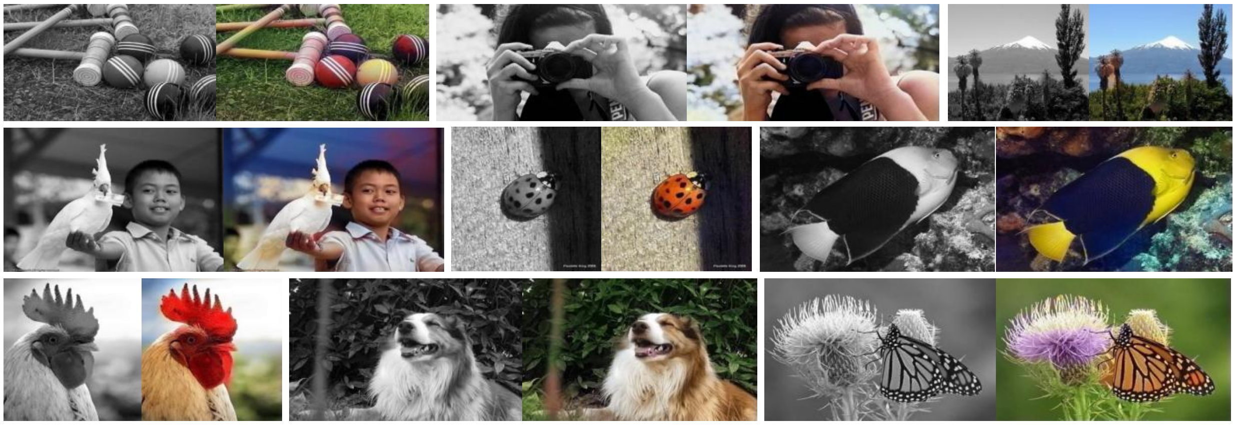


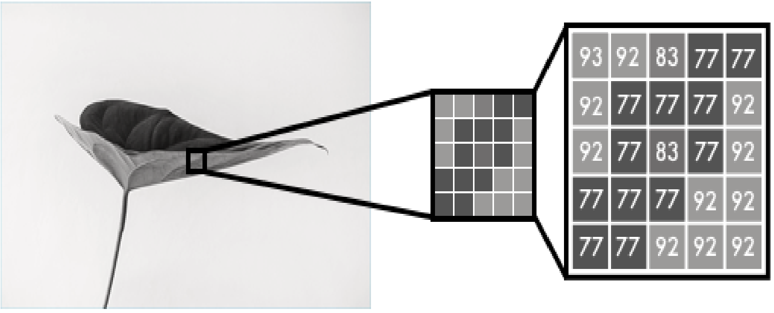
Figure 1. Colourizations related to the problem (ECCV, 2016)

On being given the luminescence channel L, our algorithm works on to forecast the corresponding a and b colour channels in the CIE Lab colour. We grasp large-scale data to solve this problem. One of the best aspects of colour prediction is that training data is virtually free, we can choose any photo as a training assignment, only by taking L channel of the image as input and its corresponding a and b channels to be used as a supervisory. We would use a loss deemed fit for the problem of colourization. For appropriate modelling of multimodal problem, a distribution of possible combination of colours is predicted. The loss is reweighted at time of training to express those colours which are rare. This strategy ensures that full diversity of large-scale database is exploited. In the end, a final colorization is produced by means of annealed mean of the distribution. The final result consists of perceptually realistic and vibrant colorizations as compared to prior approaches.

Approach

We have trained a CNN for mapping a grayscale input image to a quantized colour value distribution by using the network architecture for our model. Each of the convolution (conv) layer in the architecture corresponds to a block of 2 or 3 recurrent conv and ReLU layer. All the variations in resolution have been achieved through spatial up sampling and down sampling in between convolution blocks.

Core logic: The value of every pixel linked to its brightness. These values range from 0–255 (see Figure 2).



93	92	83	77	77
92	77	77	77	92
92	77	83	77	92
77	77	77	92	92
77	77	92	92	92

Figure 2. Representation of Black and White images

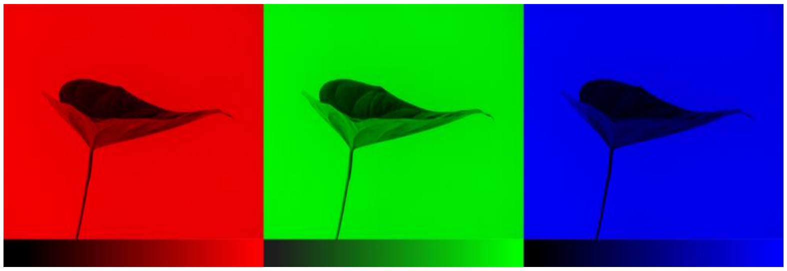
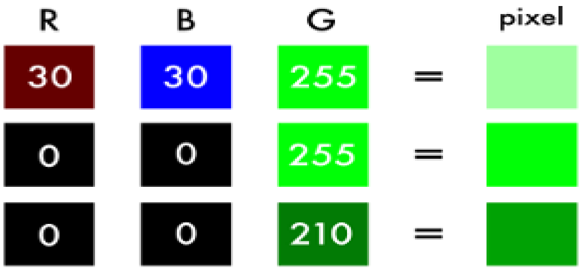


Figure 3. Distribution of colours in leaves

Colour images are composed of 3 layers that are green, red and blue layer. For example consider separating a green leaf into the 3 sections as discussed above. One may think that leaf only possesses green component. But, as can be seen in Figure 3, the spectrum of leaf is existing in all three layers. These layers also determine brightness in addition to colours. For example, in order to achieve the white colour, an equal distribution of all colours is needed. If we add an same quantity of red, blue and green colour becomes brighter (see Figure 4). Therefore, colour and the contrast are encoded in a colour image using three layers:



R	B	G		pixel
30	30	255	=	🟩
0	0	255	=	🟩
0	0	210	=	🟩

Figure 4. Combination of colours to encode colour

The range of colour image in each layer is between 0-255. Image pixels will become black if value is 0. Precisely, the neural network requires the features which link the grayscale image to its coloured counterpart.

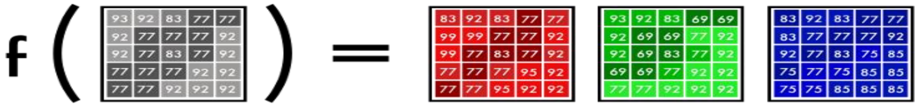


Figure 5. is the neural network, [B&W] is our input, and [R],[G],[B] is our output.

Color Space: Here used an algorithm to convert the color channels of given grayscale image, from RGB to Lab. a stands for colour spectra green–red , b means blue–yellow and L stands for lightness. As shown in Figure 5, a input encrypted picture has 1 layer of grayscale, and became three color layers into 2. It means that the original grayscale picture can be used in concluding prediction. The numbers of channels that need to be predicted are then reduced to just two.

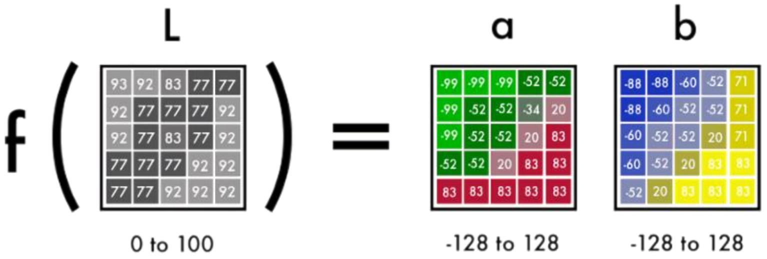


Figure 6. Transformation to Lab color space

Conversion from Grayscale to Colour: Here grayscale image is used as input, to predict 2 colour layers, a and b in CIE. To create the final colour image, the L/grayscale image. The resulting output will be a Lab encoded image (Figure 6).

To convert a single Lightness (L) layer into two layers, used convolutional filters. In case of CNN, every filter is automatically adjusted to achieve desired outcome. Initially we piled up hundreds of filters which were then fine-tuned in 2 layers. Our sole objective is to build a channel that includes CNN and a front end for pre-processing of images.

General Course: The images of pixel channels are read during the course of training in the RGB colour space. These images are then converted to the CIELAB colour space. The grayscale luminance (lightness) L channel is provided as the input to the model. The value of A and B channels is fed to target value.In the testing time, this model takes a 256×256×1 grayscale image. Two arrays of dimension 256×256×1 each are generated, as per the A and B channels of CIELAB colour space. The CIELAB representation of the image to be predicted is facilitated by concatenation of the three channels of CIELAB colour space.

Transfer Learning: Many parts of our model have been initiated from instance of VGG16 which was already trained on the ImageNet data set. Further we go on to assume that a network which has shown expertise in segregating the classes of ImageNet data set would be a right fit for our model as well. It validates our decision for application of transfer learning in the stated fashion.

Activation function: It has been used to serve the purpose of non-linearity which will follow each dense and conv layer (Dahl, 2016).

It has been proven for greater acceleration of training convergence. Additionally, rectified linear unit is easier to compute as compared to other conventional activation functions. That is why it has become a standard for CNNs today. A disadvantage of employing the rectified linear unit for purpose of activation function in a CNN is the ease with which model parameters can be updated so that the active region of function lies necessarily in the zero-gradient section. If this situation arises, the consequent backpropagated gradients will necessarily become zero leaving the neurons in the CNN permanently dormant.

Batch normalization: In our network, we have introduced a batch normalization layer before every rectified linear unit (non-linearity) layer in addition to other layers. During testing, we found that batch normalization improves the rate of training of the systems greatly. Taking an inspiration from Iizuka et al. (2016), taken the combination of CNN along with Inception-ResNet-v2 for this we used dataset of ImageNet which helps in the complete colorization process. The network is logically divided into 4 main parts as can be seen in Figure 7.

Pre-processing: The pixel values of all the image components are adjusted and scaled to retain the values within the given interval of [-1,1].

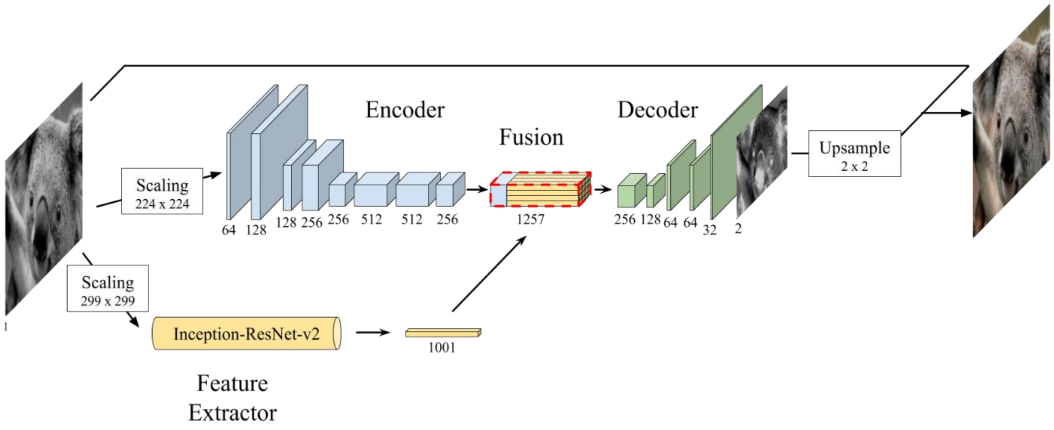


Figure 7. An overview of the model architecture (CoRR, 2017)

Encoder: H × W grayscale images are processed by the encoder. The Encoder uses 8 conv layers with 3 × 3 kernels.

Feature Extractor: Some sophisticated features convey vital information with respect to the image which can be utilized in colorization process. The picture is scaled to 299×299. For fulfilling dimension requirements of Inception, the image is stacked thrice with itself resulting in a three-channel output. The resulting output image is fed to the network and output of last layer is extracted before the softmax function. A 1001×1×1 embedding is obtained at the end.

Fusion: Feature vector is used as input. This feature vector is replicated HW/82 times and is attached to the feature volume along the depth axis. The feature vector is mirrored and is concatenated many times to ensure the uniform distribution of semantic information conveyed by the feature vector in all spatial regions. Finally, 256 convolutional kernels of size 1×1 are applied.

Decoder: It accepts H/8×W/8×256 volume and follows it with application of a multiple of conv and up-sampling layers to get a ultimate layer with measurement of H × W × 2.

Method

A deep learning network is trained for the purpose of predicting colours of an image, we are being given grayscale variant of the image. In the subsequent section, the objective of our network has been described. Then we have described two variants of the model (1) the “Local Hints Network” (LHN) below, which is based on utilization of “sparse user points”, and (2) the “Global Hints Network” (GHN) in subsequent section, based on utilization of “global statistics”. In later Section, we have defined architecture of our network.

“Red layers” are also used in the LHN for (1) accommodation of “user points” ‘Ul’ and (2) prediction of a colour distribution ‘bZ.’ The GHN makes use of “green layers” of the input, to transform the global input ‘Uд’, and the result is added to the “colourization network”. A conv layer is represented by each single box, where vertical dimensions point to “feature map spatial resolution”, and the number of channels is shown by horizontal dimensions. Upsampling and downsampling operations are used to vary the resolution. When the resolution is decreased in the main network, the number of feature channel seem to double up. In upsampling, conv layers short cut connections are also integrated.

Learning to Colourize: Accepts a greyscale image ‘X ∈ $R^{H\times W\ \times 1}$’ as an input, accompanied by an “user tensor” U. Luminance L or lightness is described by the grayscale image in the CIELAB colour space. System output is given as ‘bY ∈ $R^{H\times W\ \times 2}$’, which gives the estimation of the a and b colour channels. Network is trained for minimization of the objective function which is given in Equation 1, over ‘D’, it describes a data set of user inputs, desired output colouriz~ations and grayscale images. The closeness between ground truth and network output is determined by the loss function ‘L’.

$$\theta^* = \arg min_{\theta}\, E_{X,U,Y\sim D}[L(F(X,U;\theta),Y)] \qquad (1)$$

Network has been trained on two variants, one with “global user hints”, Uд and other with “local user hints”, Ul. While training the network, a “peek", or projection, of the ground truth Y is given to produce hints using the functions ‘Pl’ and ‘P∂’ as shown in Equation 2.

$$Ul = Pl(Y), U\partial = P\partial(Y) \tag{2}$$

The minimization problem of LHN and GHN is given by Equation 3 and 4. Since the functions Pl and P∂ are used to synthetically process user inputs, thus the data set requires colour and rayscale ge only.

$$\theta_{l^*} = \text{arg } min_{\theta l}\, E_{X,Y\sim D}\, [L\big(Fl\big(X, Ul;^{\theta}\ l\big), Y\big)] \tag{3}$$

$$\theta_{\partial^*} = \text{arg } min\, E_{X,Y\sim D}\, [L\big(F\partial\big(X, U\partial;^{\theta}\ \partial\big), Y\big)] \tag{4}$$

Loss Function. A loss function ‘L’ guides in learning and gauges performance of network, its selection needs adequate amount of thought. An ‘ℓ2’ loss is used by [4].

Prior works in this field have highlighted the fact that the problem is inherently “multi-modal” in nature and even this loss is not immune to the challenges posed by it. Because of this reason a classification loss is applied in consonance with a “fixed inference step”. A large imbalance in statistics of the natural images is also a matter of dispute, as there are more pixels in unsaturated areas of the colour spread. It can often result in unsaturated and uninteresting colourizations. A “class-rebalancing” stage has been used by [1], so that portions of the gamut which are more colourful can be oversampled during the course of training. The end results are vibrant colourizations which can easily fool humans, at the cost of overly assertive colourized images.

Previously [Isola et al.] have used an ‘ℓ1’ regression loss function combined with a “Generative Adversarial Network” (GAN) element in the pix2pix framework. Many exciting, higher frequency patterns can be generated by using GAN along with ‘ℓ1’ loss.

In LHN, it was found to be more convenient to use a “conservative colourization” with a permission to the user to add those colours which are desired, instead of initiating with a more rousing “artefact-prone” setting and expecting user to fix errors.

Local Hints Network: The LHN takes input as “sparse user points”. The input and simulation of user points can be described as below.

System Input. “User points” can be represented using parameters such as ‘Xab ∈ $R^{H\times W \times 2}$’, is a ‘sparse tensor’ with a and b values defined solely by user and ‘Bab ∈ $B^{H\times W \times 1}$’, represents a “binary mask” which differentiates user provided points. Unspecified points are differentiated from user provided ones, with (a,b) = 0 by the binary mask.

Simulation of User Interaction. Collection of training data is a major challenge in deep learning networks. Although data is easily accessible for “automatic colourization” purpose, since grayscale and colour components can be easily extracted from any colour image, but it is a quite daunting task to acquire user interaction data. As user interaction behaviour is strongly dependent on the performance of system, it is difficult to assess the cause and effect trail between the two. Training with the help of artificial user interactions can help us cope this problem. If we simply employ random sampling of data, we can also avoid the problem of inadequate space coverage which will result in an efficient system.

Global Hints Network: Easier adaptation of the “end to end learning framework” according to type of input given by user, is an added advantage.

Global statistics, as provided by the user can also be used, these require “global histograms”. These histograms can be effectively computed using by way resizing of colour ‘Y’ up to one fourth resolution, then each pixel is quantized and averaged in a spatial manner. One way to compute saturation is by conversion of “ground truth image” to HSV colour space, followed by spatial averaging over channel ‘S’.

Network Architecture: The main colourization network is used by LHN and GHN both. Here we have described layers specific for LHN, namely the “sparse user input” processing unit and the colour spread predicting unit. Similarly, the layers used for GHN only are described.

Main colourization branch. The ‘F’ branch is considered as the main branch, it is based on the “U-net architecture” which is considered versatile for ‘conditional generation’ tasks [2]. Whole network comprises many conv blocks, each consisting of two-three conv-ReLU pairs. For recovery of spatial data symmetric shortcut connections are also integrated. It facilitates the ease of access to low level data for the subsequent layers; e.g. the ab gamut is constrained by the value of luminance. We use downsampling and upsampling to realize a variation in the spatial resolution. Addition of “BatchNorm” layers after each conv layer has been proven to aid training task.

Local Hints Network. In LHN, integration of sparse user points is done by concatenating them with the input image. The features of main branch are being used to predict colour spread, since it is closely associated with the process of predicting a distinct colourization. A “hyper column” approach is employed, features are extracted from various layers of main network branch in order to concatenate them along with learning a 2-layer classification technique. In order to avoid unnecessary computations, the distribution is predicted with only one fourth resolution, followed by application of bilinear up sampling for full resolution prediction.

In Figure 8, a comparison has been shown between different approaches as illustrated earlier [1]. Column 2 to 5 illustrate how the automatic methods perform. Similarly, Ground truth has been shown in the last column (unknown to the user). The colourization results given by our system in row 1 to 4 are of high quality. As can be seen in rows 5 and 6 the colours have not propagated successfully, it is due to the presence of non-localised effects which are otherwise undesirable.

Global Hints Network. Since there is no spatial data present in the "global inputs", the data is integrated in the intermediate part of the colourization framework. Processing of inputs is done by using conv-ReLU layers where size of each kernel is 1×1 having 512 channels. The "feature map" needs to be repeated in a spatial manner so that it matches the size of 'conv4' feature of the main network branch ('RH/8×W/8×512'), it is then amalgamated using summation approach.

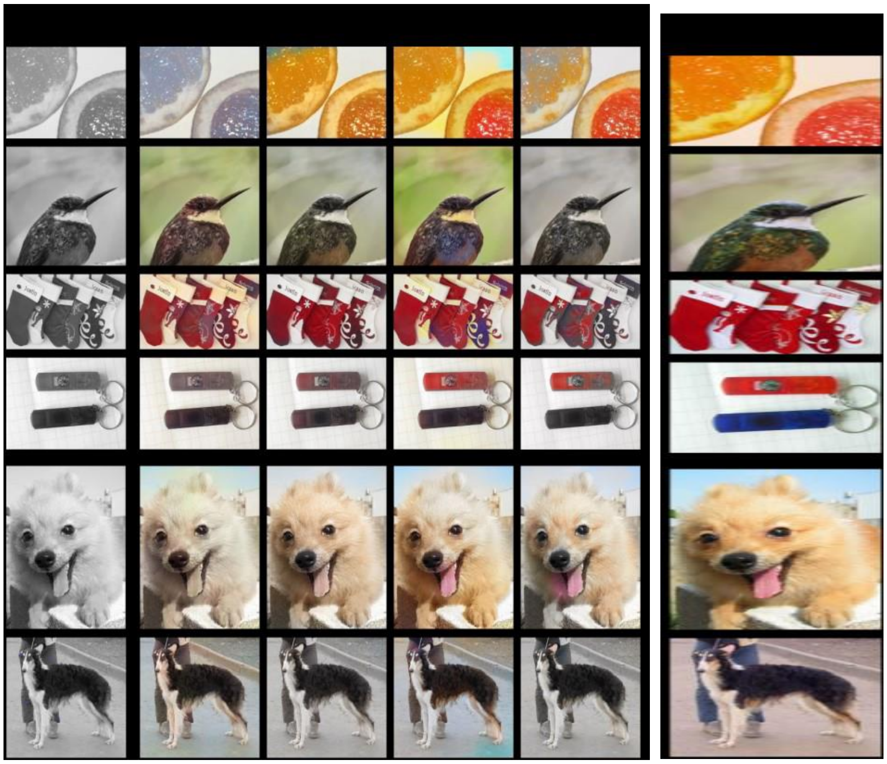


Figure 8. Comparison with other existing models

Results

We implemented the model architecture using Keras with Tensorflow at the backend. We initially validated our model architecture with just 1 train image, it was able to reconstruct the exact input image from the greyscale version since it learned (or memorized the values) but it was not able to colourize any other image since the model was not generalized. After this we created another version to train on multiple images and test the same. The model was trained on 100 epochs with 19 steps per epoch, the batch size we used was 500 since the memory size we used was limited. Keras also allows to rotate and shear input images in each run to avoid the model from over fitting and treat each rotated or sheared image as different. We saved the trained model architecture in a json file along with the final trained weights that can be used to colourize any image without further training the model again from starting. This was again implemented using the Keras inbuilt functions save_weights and to_json.

Our model predicts two arrays for the a and b values in the Lab colour space. Since the predicted values are normalized, we convert them back to original range of -128 to 128 by scalar multiplication of arrays with 128. While testing the model we create an empty canvas of 256 x 256 x 3 and put the greyscale image as one layer, and the other two predicted arrays as a and b layers. The image is then finally converted back to RGB and saved for evaluation.Some of the best results that we were able to generate using our model are shown in Figure 9 below. The table depicts a set of predicted images and the ground truth images along with associated grayscale images. Our network predicted results are reasonable to an extent if we compare them to ground truth image. There also exists a noticeable amount of noise in some of the results but it is negligible and there can be many factors which may have attributed to this problem.

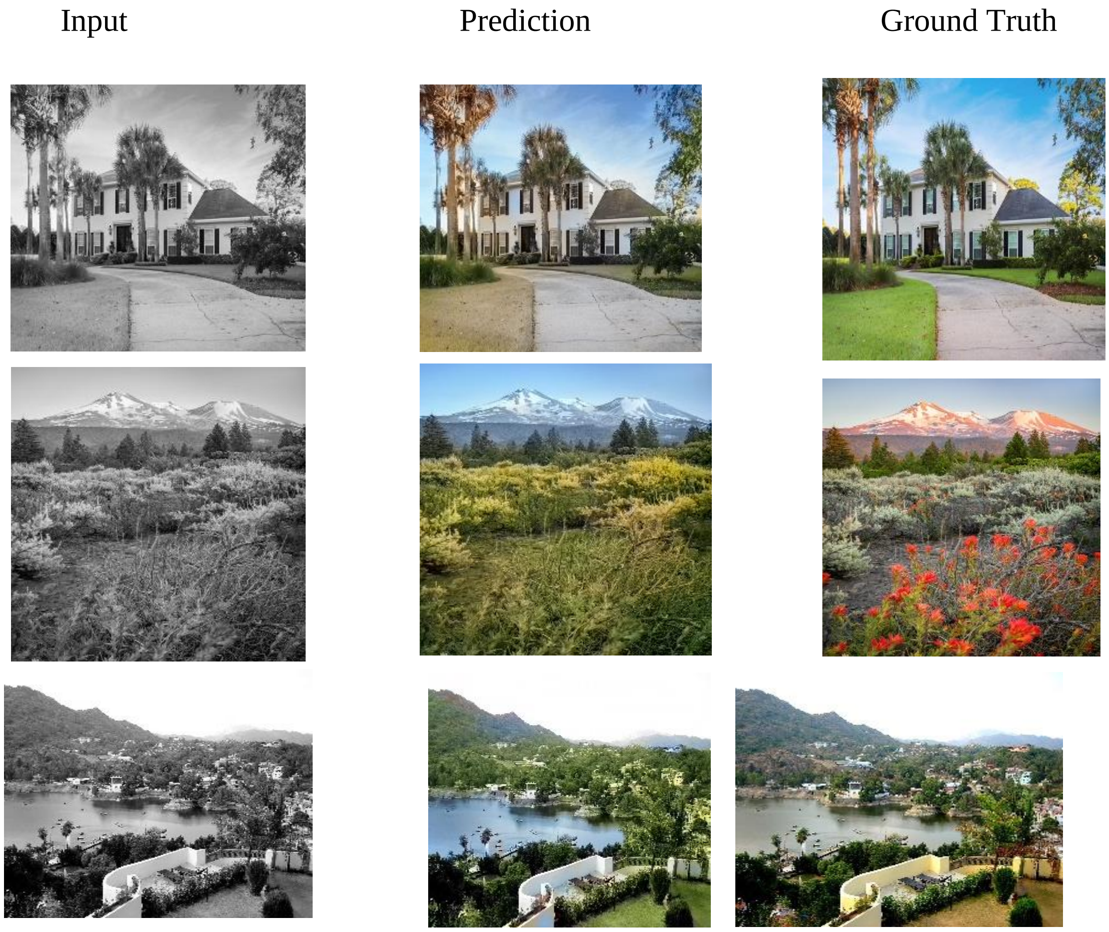

Input Prediction Ground Truth

Figure 9. Best Results generated by using our model

There also exists a noticeable amount of noise in some of the results but it is negligible and there can be many factors which may have attributed to this problem. There are some input cases in which the colourization output may not seem much vibrant to human eyes but is quite acceptable. These cases are shown below in Figure 10.

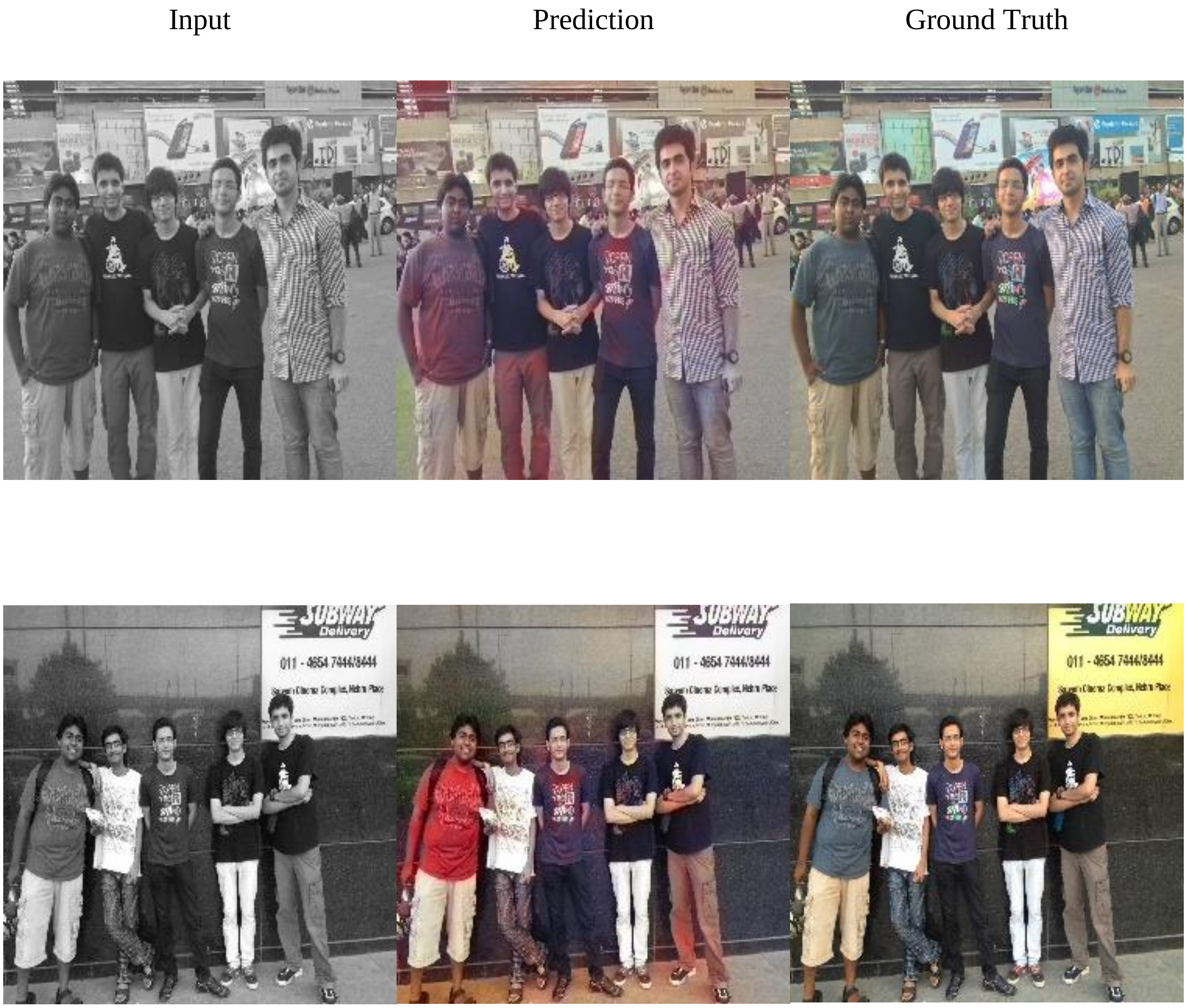


Figure 10. Some faulty Results generated by using our model

We have also applied our algorithm on some legacy photographs to test the robustness of our model. Since the grayscale images which were used as input on this model were synthetically generated from ground truth images, we needed a real-world grayscale input image. This opportunity can only be provided by historical photographs. Some of the results generated by our model on legacy images are shown in Figure 11 below.

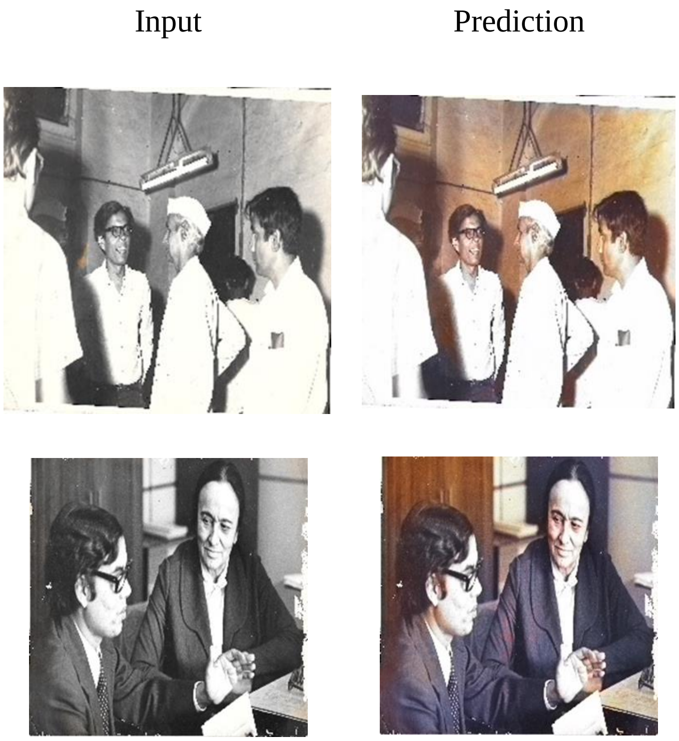


Figure 11. Some purely predicted results in absence of ground truth generated by using our model

We can clearly notice that our algorithm is capable of producing colourizations on legacy images which are of acceptable quality, even though the low-level statistics of historical photographs are not similar to those photographs which have been used in training.

Conclusion

Our model is able to decently colourize images that involve scenery and outdoor settings, but it fails on images that include multiple objects and give a brownish effect in the overall image. This brownish outcome is expected and is common in colourizing models when the model is not able to generalize on different input images. This could be perhaps resolved with a larger dataset and tweaking with other hyper parameters. Overall, we gained newfound appreciation for the challenge of producing realistic colourization of greyscale images and enjoyed experimenting and tweaking the approaches of previous papers.

Future Work

Much more can be done in this task of assigning colours to pixels, ours is a simple CNN model that performed well beyond our expectations for some set of input images. If we had more resources and time there are many ways in which we could improve our model. More time and computing power would let us train on a larger dataset and and work with images with a larger size than 256 x 256 that we have limited as of now. Another area we wanted to explore was working with different loss functions, initially we decided to test on different loss functions and compare results but as the model took many hours to train it was not feasible and we settled on the mean squared error loss function. Futher we can extend this approach to colourize old gray-scale videos which have colour information lost. One simple way would be to extract all the frames from a video as individual images and colourize them one by one, then combine the coloured images into one.

Conflict of interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

Chang, H., Fried, O., Liu, Y., DiVerdi, S., & Finkelstein, A. (2015). Palette-based photo recoloring. ACM Transactions on Graphics (TOG), 34(4), 139.

Chen, X., Zou, D., Zhao, Q., & Tan, P. (2012). Manifold preserving edit propagation. ACM Transactions on Graphics (TOG) 31(6), 132.

Cheng, Z., Yang, Q., & Sheng, B. (2015). Deep Colorization. IEEE International Conference on Computer Vision (ICCV), 415–423.

Chia, A.Y.S., Zhuo, S., Gupta, R.K., Tai, Y.W., Cho, S.Y., Tan, P., & Lin, S. (2011). Semantic colorization with internet images. ACM Transactions on Graphics (TOG). ACM, 30, 156.

Dahl, R. (2016). Automatic colourization. Retrieved from http://tinyclouds.org/colourize

Deshpande, A., Rock, J., & Forsyth, D. (2015). Learning Large-Scale Automatic Image Colourization. IEEE International Conference on Computer Vision (ICCV), 567–575.

Gupta, R.K., Chia, A.Y.S., Rajan, D., Ng, E.S., & Zhiyong, H. (2012). Image colorization using similar images. In Proceedings of the 20th ACM international conference on Multimedia (ACMM '12). Nara, Japan: ACM, 369–378.

Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., & Salesin. D.H. (2001). Image analogies. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques. Los Angeles, CA: ACM.

Huang, Y.C., Tung, Y.S., Chen, J.C., Wang, S.W., & Wu, J.L. (2005). An adaptive edge detection based colorization algorithm and its applications. Proceedings of the 13th annual ACM international conference on Multimedia. Singapore: ACM, 351-354.

Iizuka, S., Simo-Serra, E., & Ishikawa, H. (2016). Let there be Colour: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colourization with Simultaneous Classification. ACM Transactions on Graphics (TOG), 35(4), Article 110.

Irony, R., Cohen-Or, D., & Lischinski, D. (2005). Colorization by example. Eurographics Symp. on Rendering. Citeseer,2.

Isola, P., Zhu, J.Y., Zhou, T., & Efros, A.A. (2017). Image-to-image translation with conditional adversarial networks. CVPR.

Larsson, G., Maire, M., & Shakhnarovich, G. (2016). Learning Representations for Automatic Colourization. Computer Vision – ECCV 2016, 577-593.

Levin, A., Lischinski, D., & Weiss, Y. (2004). Colorization using Optimization. ACM Transactions on Graphics.

Li, X., Zhao, H., Nie, G., and Huang, H. (2015). Image recoloring using geodesic distance based color harmonization. Computational Visual Media 1(2), 143– 155.

Li, Y., Adelson, E., & Agarwala, A. (2008). ScribbleBoost: Adding Classification to Edge-Aware Interpolation of Local Image and Video Adjustments. Computer Graphics Forum. Wiley Online Library, 27, 1255–1264.

Liu, X., Wan, L., Qu, Y., Wong, T.T., Lin, S., Leung, C.S., & Heng, P.A. (2008). Intrinsic colorization. ACM Transactions on Graphics (TOG). ACM, 27, 152.

Liu, Y., Cohen, M., Uyttendaele, M., & Rusinkiewicz, S. (2014). AutoStyle: automatic style transfer from image collections to users' images. Computer Graphics Forum. Wiley Online Library, 33, 21–31.

Sangkloy, P., Lu, J., Fang, C., Yu, F., & Hays, J. (2017). Scribbler: Controlling Deep Image Synthesis with Sketch and Color. CVPR (2017).

Simo-Serra, E., Iizuka, S., Sasaki, K., & Ishikawa, H. (2016). Learning to simplify: fully convolutional networks for rough sketch cleanup. ACM Transactions on Graphics (TOG). ACM, 35(4), 121.

Wang, B., Yu, Y., Wong, T.T., Chen, C., & Xu, Y.Q. (2010). Data-driven image color theme enhancement. ACM Transactions on Graphics (TOG). ACM, 29, 146.

Welsh, T., Ashikhmin, M., & Mueller, K. (2002). Transferring color to greyscale images. ACM Transactions on Graphics (TOG). ACM, 21(3), 277–280.

Yan, Z., Zhang, H., Wang, B., Paris, S., & Yu, Y. (2016). Automatic photo adjustment using deep neural networks. ACM Transactions on Graphics (TOG). ACM, 35(2), 11.

Zhang, R., Isola, P., & Efros, A.A. (2016). Colourful Image Colourization. Computer Vision – ECCV 2016, 649-666.

Zhang, R., Zhu, J.U., Isola, P., Geng, X., Lin, A.S., Yu, T., & Efros, A.A. (2017). Real-Time User-Guided Image Colourization with Learned Deep Priors. SIGGRAPH.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., & Oliva, A. (2014). Learning deep features for scene recognition using places database. In Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS '14). Montreal, QC: MIT Press, 487-495.

Bibliographic information of this paper for citing:

Saini, Anu, & Tripathi, Jyoti (2022). Mapping Grayscale Images to Colour Space Using Deep Learning. *Journal of Information Technology Management*, Special Issue. 52-68.

Copyright © 2022, Anu Saini and Jyoti Tripathi

cc BY NC