Automatic Image and Video Colourisation using Deep Learning

Brian Sam Thomas
Department of Computer Engineering
St. John College Of Engineering and Management
Village Vevoor, Manor Road, Palghar (East) 401404
brians95@gmail.com

Rajat Dogra
Department of Computer Engineering
St. John College Of Engineering and Management
Village Vevoor, Manor Road, Palghar (East) 401404
kakarajat1070@gmail.com

Bhaskar Dixit
Department of Computer Engineering
St. John College Of Engineering and Management
Village Vevoor, Manor Road, Palghar (East) 401404
bhaskardixit786@gmail.com

Aditi Raut
Department of Computer Engineering
St. John College Of Engineering and Management
Village Vevoor, Manor Road, Palghar (East) 401404
aditir@sjcet.co.in

***Abstract*— A pivotal area of research among the machine learning and computer vision communities is the Colourisation of monochrome/black and white images. Colourisation is the computer-assisted process of adding colour to a greyscale image/movie. Traditionally, this process required significant user interaction, in the form of placing numerous colour scribbles, looking at related images, and performing segmentation. However, with advancements in technology, automated Colourisation systems have been created. Apart from the aesthetic appeal, such capability has broad practical applications ranging from video restoration to image enhancement for improved comprehensibility. However, these current systems face some major challenges, such as - Colour inconsistency within individual objects, under/over-saturation, and green tones in bright environments. The main purpose here is to eliminate the issues faced by current systems, and at the same time, efficiently colourise Black & White images and videos.**

Keywords*— *Computer Vision, Convolutional Neural Networks, Deep Learning, Image Colourisation, LSTM.

I. INTRODUCTION

Semantic Awareness has been the key aspect in Colourisation so far. However, Contextual awareness in Colourisation is the key to efficient automated Colourisation, especially in videos. Our system will colourise greyscale image frames based on the premise that: *Neighbouring image frames with neighbouring pixels in space-time having similar intensities should have similar colours.* Based on newly-found research and techniques, in this system, we use Long Short-Term Memory (LSTM) Convolutional Neural Networks (CNNs) to colourise greyscale images. This basically means colouring objects in frames based on the colour of those objects in previous frames. This can be referred to as Contextual Awareness in Colourisation.

We design and build a Convolutional Neural Network (CNN)-Long Short-Term Memory (LSTM) that accepts a black-and-white image as an input and generates a colourised version of the image as its output; The colouring information is stored by the LSTM, and provided for colouring the next image (in a sequence of image frames, i.e. a video). This system colours an image solely on *images it has learnt from* in the past, with no direct human intervention.

II. RELATED WORKS

Our Project was inspired by Richard Zhang's - Colourful Image Colourisation [1]. Zhang's system uses AlexNets layers trained on the ImageNet dataset. Our method also learns to classify colours, but with some improvements to the loss function.

Another inspiration is Ryan Dahl's CNN based system for automatically colourizing images [2]. Dahl's system relies on several ImageNet-trained layers from VGG16 [3], integrating them with an auto-encoder like system with residual connections that merge intermediate outputs produced by the encoding portion of the network comprising the VGG16 layers with those produced by the latter decoding portion of the network.

In terms of results, Dahl's system performs extremely well in realistically colourizing foliage, skies, and skin. This technique, however, resulted in many over/under-saturated, sepia-toned images. Dahl used a regression based approach, however, a classification based approach is better suited for the task of colouring images.

A. Abbreviations and Acronyms

LSTM – Long Short-Term Memory is a module used along with a Neural Network in order to retain information after the output is generated. LSTMs basically hold information for as long as required.

CNN – A Convolutional Neural Network is a neural network capable of handling image data. It consists of 3 layers, 1 convolution layer, 1 pooling layer, and 1 fully connected layer. We will look at how each of these work in further depth in the following sections.

III. Approach

We created a pipeline, as shown in the architecture below. In case of videos, the LSTM will be used. If single image, LSTM is omitted. The following figure shows the overall pipeline/architecture of our network.

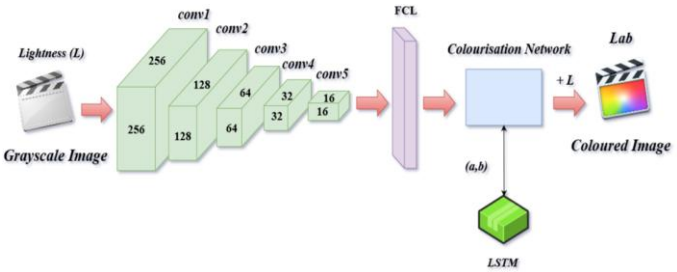

Fig 1. Network Architecture.

A. Network Architecture

As per fig 1, the Input Image, or (Extracted Frames from a) video is passed through CNN Layers. Each *conv layer* consists of a set of convolution & pooling layers. 5 such layers are used. The output of each layer is passed through Batch Normalization [4] before it enters the next.

The fifth layer's output is directly given to the fully connected layer for classification. In this layer, based on the *softmax* function, the different objects are identified/classified. The Lab colour space data is generated. In case of videos, this data is stored in the LSTM upon each frame's entry into the colourisation network. If the input is a single image, this data is directly used to colour the image, and the coloured image is received as an output.

B. Activation Function

We use the rectified linear unit (ReLU) as the activation function that follows each of our convolutional layers. Mathematically it is defined as -

$$f(x) = max(0; x)$$

ReLU has been known to greatly accelerate training convergence [5]. It is used as a standard activation function for convolutional neural networks due to the simplicity of its computation when compared to other activation functions.

One *theoretical* downside of using the ReLU is that the model parameters can be updated in such a way that the function's gradient is always in the *zero-gradient* region. In theory, once a ReLU activation ends up in this state, it is unlikely to recover. In practice however, this has not created any issues.

C. Batch Normalization

Ioffe et al introduced batch normalization as a means of dramatically reducing training convergence time and improving accuracy [4]. For our networks, we place a batch normalization layer before every CNN layer apart from the first layer, and after the last layer before the output. This move improves training times of the system.

D. Transfer Learning Approach

We initialized parts of the model with the [5] Inception-v4 (a.k.a. -Inception-ResNet) instance that has been pre-trained on the ImageNet [13] dataset. A network that has demonstrated excellence with the vast number of classes present in the ImageNet dataset would serve well as the foundational basis of our network. The reason this works well is because Image subject matter implies colour palette, almost always.

E. Objective Function

Given an input L (Lightness) channel $\mathbf{X} \in \mathbb{R}^{HxWx1}$ (where H and W stand for Height and Width respectively), our objective is to learn the mapping of $\hat{\mathbf{Y}} = \mathbf{F(X)}$ with respect to the two associated colour spaces, i.e. a, b channels $\mathbf{Y} \in \mathbb{R}^{HxWx2}$. We perform this task in CIE Lab colour space. Because distances in this space model perceptual distance, a natural objective function, as used in [1] is the Euclidean loss L2 between predicted and ground truth colors, as follows:

$$\mathbf{L_2(\hat{Y},Y)} = \frac{1}{2}\sum_{h,w}\left\|Y_{h,w} - \hat{Y}_{h,w}\right\|_2^2 \qquad \mathbf{(1)}$$

(Here, we represent the prediction with a ^ symbol, and the ground truth without.)

However, if an object takes on a set of distinct a,b values, the optimal solution to the Euclidean loss will be the mean of the set. In color prediction, this averaging effect favours greyish, de-saturated results. So we too [1], treat the problem as multinomial classification. We quantize the a,b output space into bins with grid size 10 and keep the Q = 310 values which are in-gamut.

For a given input **X**, we learn a mapping $\hat{\mathbf{Z}} = \mathbf{G(X)}$ to a probability distribution over possible colours as - $\hat{\mathbf{Z}} \in [\mathbf{0}, \mathbf{1}]^{HxWxQ}$, where Q is the number of quantized a,b values.

To compare the predicted $\hat{\boldsymbol{Z}}$ against ground truth $\boldsymbol{Z}$, we define the function $\boldsymbol{Z} = \boldsymbol{H}_{\boldsymbol{gt}}^{-\mathbf{1}}(\boldsymbol{Y})$ which converts the ground truth colour Y to a vector Z, using soft-encoding schemes.

We then use a cross-entropy Loss function as follows –

$$\boldsymbol{L}_{\boldsymbol{cl}}\left(\hat{\boldsymbol{Z}}, \boldsymbol{Z}\right) = -\sum_{\boldsymbol{h},\boldsymbol{w}} \boldsymbol{v}\left(\boldsymbol{Z}_{\boldsymbol{h},\boldsymbol{w}}\right) \sum_{\boldsymbol{q}} \boldsymbol{Z}_{\boldsymbol{h},\boldsymbol{w},\boldsymbol{q}} \boldsymbol{log}\left(\hat{\boldsymbol{Z}}_{\boldsymbol{h},\boldsymbol{w},\boldsymbol{q}}\right) \quad \textbf{(2)}$$

Where, $\boldsymbol{v}$ is a weighting term that can be used to rebalance the loss based on colour-class rarity.

Finally, we map, as in [1], the probability distribution $\hat{\boldsymbol{Z}}$ to colour values $\boldsymbol{Y}$ with the function $\boldsymbol{H}(\boldsymbol{Z})$.

IV. Preprocessing And Results

Both, the image, and video requires pre-processing. At every layer, batch normalization is used to pre-process the image data for the next layer. However, at the beginning i.e. when the image is provided, we only do certain operations like changing the angle by a small degree, and flipping the image. We perform these operations since the contents in an image do not have to be at the same position/angle in every case.

A. Frame Extraction & Stitching

We use the OpenCV (Open Source Computer Vision Library) python library to extract frames from video data. Using the *VideoCapture()* function we start capturing the video data. *cap.read()* is used to read/capture frames one-by-one. *imwrite(name, frame)* is used to write the frames with proper numbering, etc. into a predefined directory, in PNG format.

In the same way, the output coloured frames need to be stitched back. Using the *VideoWriter()* function from OpenCV, we can stitch back the frames to create a video in MP4 format.

B. Final Input Processing

During the learning phase, the images in the training set need to be first converted to single channel (black and white), and then to its CIE Lab space equivalent (with only L values). This will be fed to system to train on, and the coloured image or the actual image will be used as the target for the system to back-propagate and modify weights based on the error.

During testing phase, the images used will be in RGB black and white format, so they will be converted to their CIE Lab space equivalent with only the Lightness channel values.

C. Comparison of Accuracy

TABLE I. Comparison of System Accuracy

Serial No.	Colourisation Accuracy	
	System Name	*Accuracy (%)*
1.	CIC [1]	56.0%
2.	ICDCNN [2]	60.2%
3.	AIVCUDL (Ours)	68%

Fig. 2. Comparison of accuracy on colourisation systems.

D. Colourisation Test Case Results

The following are some of our success cases, the image on the left is the input, and on the right is the colourised output.

Fig 3. Colourisation Test Case Results

The system does a good job at colourising the images in moderate to very bright environments by eliminating the green-tones and oversaturation. The system could colour the bird better or differently, as there could have more possibilities of colours for the bird. This can be further

improved my training the system on more images of similar birds or just birds in general.

V. Conclusion & Future Works

In conclusion, we have found that our system classifies and colours objects with improved accuracy, when compared with previous systems from [1] and [2]. In videos, we have eliminated the frame-by-frame colour inconsistencies, which results in smoother colours and overall an aesthetically pleasing viewing experience. This system can be used in the future of coloring of old black and white movies and images, as well as color-restoration of old images. To further improve this system, one should work towards improving the CNN model with better/faster alternatives, which can classify faster but without loss of image information. The use of LSTM can be further improved with respect to information retention functions for better accuracy in frame matching.

References

[1] Richard Zhang, Phillip Isola, Alexei A. Efros. "Colourful Image Colourisation" In *University of California, Berkeley, October 2016. http://richzhang.github.io/Colourisation/*

[2] Jeff Hwang, You Zhou. "Image Colourisation with Deep Convolutional Neural Networks" In *Stanford University Research Projects, 2016.*

[3] R. Dahl. "Automatic Colourisation" In *http://tinyclouds.org/colourise/ 2016.*

[4] Ioffe, S., Szegedy, C. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." In *arXiv preprint arXiv:1502.03167, 2015.*

[5] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning" In *arXiv preprint arXiv:1602.07261v2, 2016.*

[6] Krasin I., Duerig T., Alldrin N., Ferrari V., Abu-El-Haija S., Kuznetsova A., Rom H., Uijlings J., Popov S., Veit A., Belongie S., Gomes V., Gupta A., Sun C., Chechik G., Cai D., Feng Z., Narayanan D., Murphy K. "OpenImages: A public dataset for large-scale multi-label and multi-class image classification, 2017." In *https://github.com/openimages*.

[7] Liu, X., Wan, L., Qu, Y., Wong, T.T., Lin, S., Leung, C.S., Heng, P.A. "Intrinsic Colourisation." In *ACM Transactions on Graphics (TOG). Volume 27, ACM (2008) 152.*

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks." *In Advances in neural information processing systems, pages 1097–1105, 2012.*

[9] Iizuka, S., Simo-Serra, E., Ishikawa, H. "Let there be Colour!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colourisation with Simultaneous Classification." In *ACM Transactions on Graphics (Proc. of SIGGRAPH2016) 35(4) (2016).*

[10] Welsh, T., Ashikhmin, M., Mueller, K. "Transferring colour to greyscale images." In *ACM Transactions on Graphics (TOG) 21(3) (2002) 277-280.*

[11] Simonyan, K., Zisserman, A. "Very deep convolutional networks for large-scale image recognition." In *arXiv preprint arXiv:1409.1556, 2014.*

[12] Mauricio Marengoni, Denise Stringhini. "High Level Computer Vision Using OpenCV", In *Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 2011 24th SIBGRAPI Conference on 28-30 Aug. 2011.*

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "ImageNet: A large-scale hierarchical image database." In *CVPR, 2009.*

[14] Gupta, R.K., Chia, A.Y.S., Rajan, D., Ng, E.S., Zhiyong, H. "Image Colourisation using similar images." In *Proceedings of the 20th ACM International Conference on Multimedia, ACM (2012) 369-378*

[15] D. Kingma and J. Ba. Adam. "A method for stochastic optimization." In *arXiv preprint arXiv:1412.6980, 2014.*

[16] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition." In *arXiv preprint arXiv:1512.03385, 2015.*