ISSN No:-2456-2165

Colorizing Images using CNN in Machine Learning

Siddhartha Kaushik
Information Technology Engg.
Galgotias College of Engineering and Technology
Greater Noida, India

Ujjwal Jagtiani
Information Technology Engg.
Galgotias College of Engineering and Technology
Greater Noida, India

Vinamra Kumar Jain
Information Technology Engg.
Galgotias College of Engineering and Technology
Greater Noida, India

Suresh kumar
Assistant Professor
Galgotias College of Engineering and Technology
Greater Noida, India

Javed Miya
Associate professor
Galgotias College of Engineering and Technology
Greater Noida, India

Abstract:- Our research paper proposes a model of fully automatic Convolutional Neural Network for converting greyscale image to colored image. The issue is under constrained, because of which earlier methodologies have either resulted in unsaturated color production or relied on considerable user involvement. Our deep neural network introduces a fusion layer that allows us to effectively merge low-level information extracted from multiple small image patches with overall features extracted from the entire image. The makes a direct use the greyscale image (L channel) and predicts A and B channels for LAB color space. The predicted values of AB channel are concatenated with the input L channel and then it is converted to RGB color space for visualization. Additionally, our model can take and process images of any resolution, this makes our model different from other approaches based on CNN. We compare our approach against the state of the art [Z Cheng's Model] and validate the results with a user study, where we demonstrate considerable improvements.

Keywords:- *Colorization, Convolutional Neural Network, Machine Learning.*

I. INTRODUCTION

Conventional methods of image colorization require considerable user involvement. Like looking at related images, placing numerous color scribbles, or performing segmentation. In this paper, we propose an end-to-end trained network that fully automate the conversion of greyscale images into colorized ones. Our approach uses a combination of low-level extraction features network, which generates information by computing small image patches and Overall features extraction network, which generates information from the entire image. Low-level features represent the different object or textures at a particular location while Overall features provides information at an overall image level such as different lighting conditions when the image was taken at day or night or what kind of weather was there as it affects the color production. By combining information from both the networks, we can obtain great results without human involvement.

Our model is based on deep learning class called Convolutional neural network (CNN). We propose a different architecture that can perform the colorization of image by fusing the output extracted from overall and low-level features from an image. Our main model consists of three different networks: a low-level features extraction network, an overall features extraction network, and a colorization network.

Theoretically, the network operate as follows: a set of shared low-level features are extracted from the image and a set of overall image features are computed from them. Then both features are fused together, and the result is fed to the colorization network that outputs the final color palette. This palette is merged with the greyscale image to create a colorized image. This is not a sequential procedure; rather, a concurrent one as both overall and low-level features extraction network can run simultaneously.

II. RELATED WORK

Propagation of user-specific colored scribbles to the entire image is one among more traditional approaches for image colorization. A framework that is optimization-based was proposed by Levin [2004] used the same traditional approach for colorizing a greyscale image. The difference in intensities of two neighboring pixels helped in deriving a quadratic cost function that made optimization possible. Huang [2005] improved this method with an aim to prevent the color bleeding onto the boundaries of an object. A faster colorization technique was proposed by Sapiro and Yatziv [2004]. This technique of colorization was made using chrominance blending which was in turn based on weighted geodesic distances. For a better and effective propagation of colors, texture similarity was employed by Luan [2007]. Cartoon colorization also makes use of Texture classification. Based upon an optimized framework that is graph-cut-based, Sykora [2009] proposed a flexible tool, that are conveniently applicable to different drawing styles more specifically a

ISSN No:-2456-2165

colorization tool used for cartoons that are hand-drawn. Various affinity-based methods for enabling tonal editing or long-range propagation for image decolorization, have also been put forward, such as:

- Optimization with all-pair constraints on a global level,
- Radial Basis Function interpolation,
- Manifold learning.

However, user input is what these methods are heavily dependent upon and to obtain an acceptable result it require trial and error.

The methods based on scribble make use of user-given colors but in the case of example-based colorization techniques, the similarity between the colors of input image and the reference image are exploited. color transfer techniques are widely used for recoloring a color image. A general technique that took inspiration from the color transfer, was given by Welsh [2002]. This technique was capable of colorizing grayscale images, and this was done by matching the texture information between images and luminance. A supervised classification scheme resulted in an improvement in this technique. This scheme analyzed low-level features. A global optimization framework was proposed by Charpiat [2008] that deals with multi-modality. This helps in predicting the possible colors probability at each specific pixel. To perform the colorization, Gupta [2012] match superpixels, using feature matching and space voting, between the reference image and the input image. However, the methods here involve a time-consuming task which is the supply of suitable images as a reference by the user. These reference images are like the input image. In comparison to these models, our model doesn't require any annotation from the user at all.

Liu [2008] proposed an example-driven colorization where user doesn't need to provide reference images, the approach was resilient towards the illumination differences between reference and input images obtained directly from web search. However, its application is limited where exact matches can be found of the famous landmarks.

Some time back, a fully automatic approach was proposed by Cheng [2015] in which certain patches of an image are colorized and various features of it are extracted using a small neural network and the results are improved by using a Joint bilateral filtering but they used very less data to train, which limits the type of images and its capability. Moreover, their model is a high-performance segmentation based which makes them dependent on segmentation heavily which results in poor images. This provides limitation to the application of the approach. Whereas, our approach does not depend upon any pre-trained model and we perform everything in an end-to-end way starting from a large dataset which makes our model to be able to generalize too many types of images.

The neural networks have been exploited because of back propagation and for a diversity of tasks, nearly thirty years ago. In the beginning, the focus of research was on the outputs which are having a small set. However, they are now applied successfully to different tasks in which the output of the model is an image such as optical flow [Fischer 2015], super-resolution [Dong 2016], contour detection [Shen 2015], and semantic segmentation [Long 2015]. These can process any resolution images and are based on C. N. N. [Fukushima 1988; LeCun 1998]. The network we propose in this work can jointly handle two tasks, while most approaches tackle single tasks. [Eigen and Fergus 2015] used this for depth estimation, where depth, surface normal and semantic labels are predicted simultaneously and for learning feature embeddings [Bell and Bala 2015].

III. COLORIZING MODEL

The approach of our colorization model is based on Convolutional Neural Networks (CNN) that can learn complex patterns from large amounts of data just like the neurons present in a human brain. Our model comprises several components that contain important variation and form a directed flow with widely used general models. Our model can:

- produce sharper edges,
- extract local and overall features of the image for better color reproduction,
- process images of any resolution.

Fig. 1 shows an overview of the model and its various components. It has three main components: an overall feature extraction network, a low-level feature extraction network, and a colorization network that generate color palette. The components are trained using Places dataset in an end-to-end fashion. Our model generates a color palette as output that fuses with the greyscale image to produce an LAB output, it is then converted to traditional RGB format.

A. Convolutional Neural Networks

Neural Networks is a network that is formed by multiple layers. These networks predict continuous values of output from the input. Neural networks consist of layers that forms a function like:

$$Y = \sigma (b + Wx)$$

where x are the n input vector and Y are m output vector for the layer, σ is a non-linear transfer function applied component-wise, b is a bias vector and W is an m × n matrix of weights.

Convolutional Neural Networks are specialized cases of neural networks where weights of an image are shared perceptually. It consist many layers like Input layer, convolutional layer, pooling layer etc. We build a model by sticking many of these layers consecutively. This has the results in reduction of the number of parameters required by a layer to gain intuition to translate the image. At the end the network uses normal fully connected layer, which makes a specific sized output and due to this the input is also fixed. Thus, only fixed sized images are processed by these networks. A limitation that is not present in our model.

We employed Rectified Linear Unit (ReLU) to transform non-linearity for neural networks.

ReLU(a) = maximum (0, a).

Sigmoid transfer function was also used by the colorization network, which is defined as:

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}$$

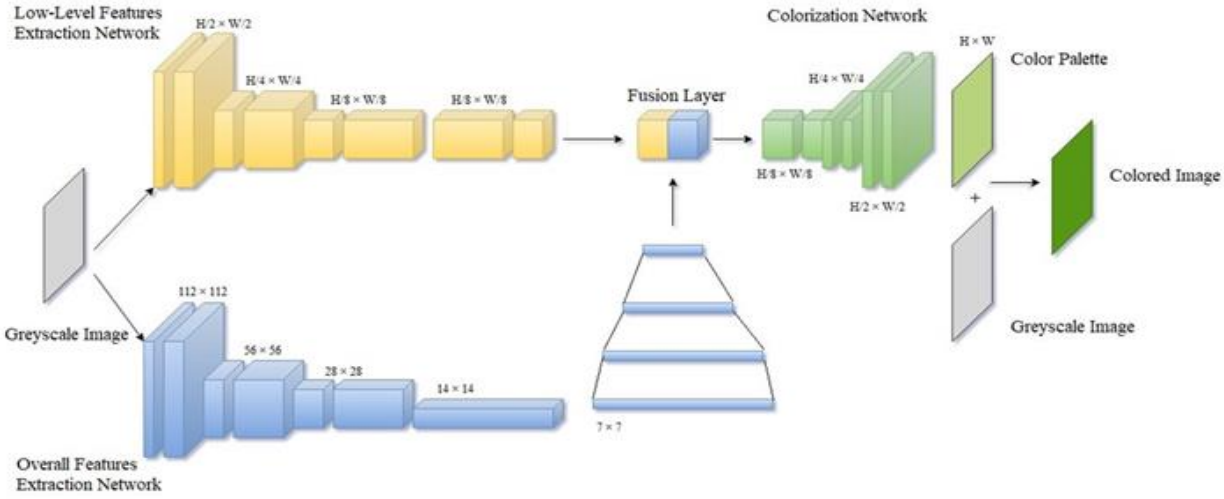

Fig. 1. An overview of the model and its components

B. Architecture

The approach we used to combine both overall and low-level features together was different. The overall features stipulate the kind of image we used as an input. For example, if the overall features stipulate that it is an outdoor image, the low-level features network will be more biased to add color of grass or sky to the image, instead of adding colors suitable for indoor lighting condition.

1. Low-Level Features Extraction Network

We obtain low-level features directly from the input image by using an 8-layer Convolutional Neural Network. It takes 224×224 image as input then the convolutions extract low-level features like edges. We are using convolution layers with increased strides to reduce the size for next convolution instead of using max-pooling layers. The structural support of each layer also increases by this. Every other pixel is calculated instead of computing values for succeeding pixels if we take a stride of two. If layers are padded, then the result is half the size of the input layer. This allows the model to not use the max-pooling layers and maintain performance. We use a padding of 1 × 1 and convolution kernels of 3×3 exclusively and to confirm that the input is the same size as the calculated output. An overview of the architecture of the shared low-level features is shown in Table 1.

Type	***Stride***	***Kernel***	***Outputs***
conv.	2 × 2	3 × 3	64
conv.	1 × 1	3 × 3	128
conv.	2 × 2	3 × 3	128
conv.	1 × 1	3 × 3	256
conv.	2 × 2	3 × 3	256
conv.	1 × 1	3 × 3	512
conv.	1 × 1	3 × 3	512
conv.	1 × 1	3 × 3	256

Table 1

2. Overall Features Extraction Network

The overall features of the image are obtained by processing another four convolutional layers executing concurrently with low-level features extraction network taking the same greyscale image as input. This output is a 256-dimensional vector representation of the image. An overview of the architecture of the overall features network is shown in Table 2.

Due to the nature of the linear layers in this network it requires the input of the low-level features network to be exactly 224 × 224 pixels. However, this limitation does not affect it's working and the model is able to extract overall features of image.

Type	***Stride***	***Kernel***	***Outputs***
conv.	2 × 2	3 × 3	512
conv.	1 × 1	3 × 3	512
conv.	2 × 2	3 × 3	512
conv.	1 × 1	3 × 3	512
FC	-	-	1024
FC			512
FC			256

Table 2

3. Fusing Overall and Low-level Features

To be able to combine low-level features (a Height/8 × Width/8 × 256-dimensional volume) of the image with overall features (a 256-dimensional vector), we include a fusion layer. The work of this layer is to assert the overall features into low-level features. The result of the fusion layer for coordinates (p, q) is given as:

$$y_{p,q}^{fusion} = \sigma\left(\mathrm{b} + \mathrm{W}\left[\frac{y^{overall}}{y_{p,q}^{low}}\right]\right)$$

where $y_{p,q}^{fusion}$ is the fused feature at (p, q), $y^{overall}$ is the overall feature vector, $y_{p,q}^{low}$ is the low-level feature at (p, q), b is a bias and W is a 256 × 512 matrix of weight.
W and b are learnt during the training of the network.

It may be imagined as fusing the overall features with low-level features at every structural location and then processing them through a single layered network. This efficiently combines the overall features and the low-level features to create a new 3D volume from a feature map. Resolution does not have any restriction on the resulting features from the overall features network.

4. Colorization Network

After the features of low-level and overall features extraction layers are fused together, they are further processed by a set of upsampling layers and convolutions, the upsampling layers simply upscale the input such that the output of one layer is twice as tall and twice as wide. It is done by using the nearest neighbor technique. The upscaling is done until the final output is same as the size of the original input. The output of colorization network is a color palette of the input greyscale image. Finally, the resulting color image is produced by combining the evaluated color palette with the input greyscale image. Here the target output is the AB channel of LAB color space while the input greyscale image is the L channel. We use Sigmoid transfer function to normalize the AB channel, so they lie in the range of [0, 1]. After that we calculate the Mean Squared Error as the loss between the output of network and target output. All the parameters of the model are updated by backpropagating loss through the networks (Overall features network, low-level features network). The architecture can be seen in Table 3.

Type	***Stride***	***Kernel***	***Outputs***
Fusion	-	-	256
conv.	1 × 1	3 × 3	128
Upsample	-	-	128
conv.	1 × 1	3 × 3	64
conv.	1 × 1	3 × 3	64
Upsample	-	-	64
conv.	1 × 1	3 × 3	32
conv.	1 × 1	3 × 3	2

Table 3

C. Learning

Our model is most efficient when the input images are 224 × 224 pixels (although it can process images of any size), as the output from low-level features network can shared. Even when the size of image is different the low-level feature extraction network share weights, but the overall features network uses a rescaled image. This is done by computing both the rescaled image and the original image through the low-level features extraction network, this however increase both the processing power and memory consumption. Since the processing time is generally under a second for evaluation this in not a problem, however during training the model need to process millions of images. Therefore, during training, it is important to be as efficient as possible. That is the reason why the model was trained exclusively with input size of 224 × 224 pixels. This is made possible by using data augmentation after scaling the training images and performing random crops to the final size.

IV. RESULTS AND DISCUSSION

We compute many iterations of our model as well as compared it against Z Cheng's Model. Our model was evaluated on a huge and diverse set of images that includes close-up images, historical greyscale images, etc. We also evaluate our model in a user study, and it further confirmed that the results of our model are considered to be natural looking 89.6% of the cases.

The model was trained on Places dataset by B. Zhou, which consists of more than 2 million training images and 20,000 testing images. Overall, the dataset has more than 200 different categories of images containing various scenes such as historic architecture, meeting room, or mountains and rivers. We removed some existing greyscale images from the dataset and some images with distorted or unnatural colors. This resulted in roughly 2.3 million training images and around 20,000 testing images. We trained our model for 100,000 iterations using a batch size of 256 equivalent to roughly 20 epochs. We randomize the training images and use those images for optimizing our model parameters. Only images from testing dataset are used for results.

A. *Results*

In Fig. 2, We demonstrate results of colorization on both outdoor and indoor images. We have used validation set of the Places dataset for these results. Mark that these images are very difficult to colorize. Our model exploits the semantic context of each image with the global features, allowing it to properly colorize fields, skies, humans, etc. All these results are automatically generated without any user involvement.

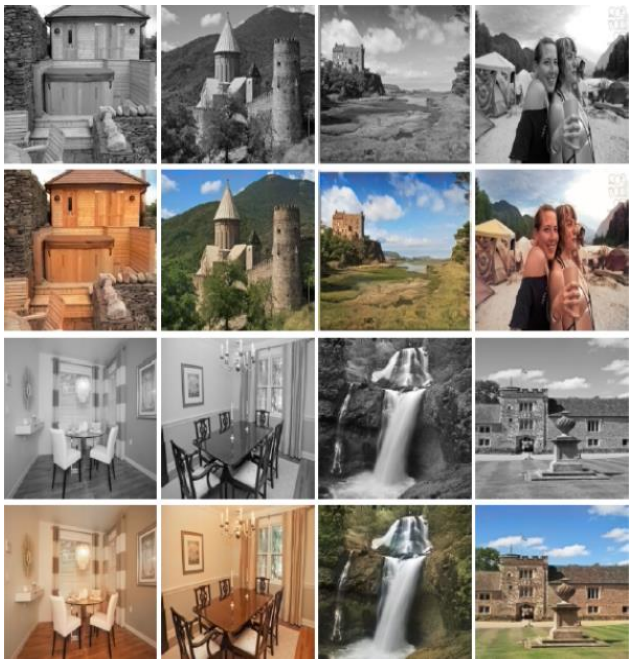

Fig. 2. – Images are from "Places dataset"

ISSN No:-2456-2165

B. *User Study*

"Is the image looking natural?" - This was the question we asked in this study when we tried to evaluate the naturalness of the images. We randomly chose the images and showed them to the users one after the other. There were 20 different users, and each user was shown nearly 100 images of each type. Each image was shown at a resolution of 224 × 224 pixels and the users were told to use their intuition and not to spend too much time looking at the finer details. The median of naturalness in our approach came out to be 89.6% which indicates that our model is able to generalize and create realistic colorization for majority of image.

C. Significance of Overall Features

The Overall features that we compute in our model play an important role in constructing the ambience of the scene like what lighting conditions are there or the environment in which the image was taken. Just computing some small image patches leaves a lot of ambiguities which makes it hard for low-level features extraction network to handle. For example, our model that does not include the Overall features network makes some serious errors such as coloring the images of oceans and lakes in brown color of ground or coloring the indoor images with blue color of sky. Our user study further confirmed this where the results of our model are considered to be natural looking 89.6% of the cases.

D. Computation Time

We used both GPU and CPU for evaluation. We used NVIDIA GeForce GTX 1060 and Intel® Core™ i7 Processor 8750H as GPU and CPU, respectively. The time taken by our model to process images of different resolutions was evaluated and the results were shown in Table 4. The mean of hundred different conversions was evaluated for a reliable time value. It is evident from the results, that for small images, both the CPU and the GPU are in order of less than a few seconds with GPU performing almost three times faster than the CPU. For larger images, GPU allows the colorization to be done within few seconds. Therefore, it is suitable to say that our approach is good enough for real-time usage and can even be used to render video files.

Image Size	***GPU***	***CPU***	***GPU performance over CPU***
224 × 224	0.153	0.459	3.0×
512 × 512	0.644	1.869	2.9×
1024 × 1024	1.867	5.974	3.2×
2048 × 2048	8.534	23.896	2.8×

Table 4

E. *Limitations`*

The main drawback of our model is that it is highly dependent on the classes of images used in the training set as it follows a data driven approach. The training of our model is done using a diverse and huge set of images containing scenes from inside and outside with different lighting conditions to reduce this limitation. But this dataset does not contain paintings and drawings. If we want to evaluate old paintings and drawings, it is recommended to train a different model for that.

Colorization of an image is also a fundamentally ambiguous problem: is the color of car red or green? Or the color of shirt is blue or purple? because this ambiguity has no single answer, our model tends to use the assertive colors out of many that it has learnt from the data, as shown in Fig. 3. Additionally, there is no accurate way for the user to control the production of colors apart from setting different overall features manually. Although, the addition of an extra optimization layer on the colorization network would likely handle the ambiguity. However, our paper does not explore this possibility.

Fig. 3. – Predicted vs Real respectively.

V. CONCLUSION

In our research, we have demonstrated a different approach for the colorization of greyscale images by combining information gained from both overall and low-level feature extraction network. Our model is based on a class of deep learning called Convolutional neural network and the colorization of an image is carried out without any intervention from the user. We trained the entire model end-to-end using huge number of images for environment recognition with a combined overall and low-level feature extraction network that feeds information to a colorization network that understand the colors and adjusts the colors according to the scenery of each image like the indoor lighting conditions are not same as outdoor or the color of the sky on a sunny-day image is not the same as on a rainy-day image. Our architecture is not limited to a single resolution, and it takes the input images of any resolution, unlike majority of CNN based colorization models. Ultimately, we evaluated our model performance on a huge and diverse set of images from places dataset consisting different lighting conditions and demonstrated that it can generate very plausible results. The model is compared with widely used Z Cheng's model and we carried out a study that validates the results. Our model can run significantly faster and has many important applications such as fast colorization of historical photographs and old greyscale movies.

ISSN No:-2456-2165

REFERENCES

[1]. Chang Y, Saito S, Nakajima M Example-based color transformation of image and video using basic color categories. Trans Img Proc 16(2):329---336. 2007.

[2]. CHIA, A. Y.-S., ZHUO, S., GUPTA, R. K., TAI, Y.-W., CHO, S.-Y., TAN, P., AND LIN, S. Semantic colorization with internet images. ACM Trans. Graph. 30, 6, 156:1–156:8. 2011.

[3]. CHENG, Z., YANG, Q., AND SHENG, B. Deep colorization. In Proceedings of ICCV 2015, 29–43. 2015.

[4]. DONG, C., LOY, C. C., HE, K., AND TANG, X. Image super-resolution using deep convolutional networks. PAMI 38, 2, 295–307. 2016.

[5]. D. Varga, T. Szirányi. Fully automatic image colorization based on Convolutional Neural Network, IEEE 23rd International Conference. 2016.

[6]. EIGEN, D., AND FERGUS, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In ICCV. 2015.

[7]. F. Baldassarre, D. G. Morín, L. Rodés-Guirao. Deep Koalarization: Image Colorization using CNNs and Inception-ResNet-v2. 2017.

[8]. FISCHER, P., DOSOVITSKIY, A., ILG, E., HAUSSER ¨ , P., HAZIRBAS, C., GOLKOV, V., VAN DER SMAGT, P., CREMERS, D., AND BROX, T. Flownet: Learning optical flow with convolutional networks. 2015.

[9]. Flownet: Learning optical flow with convolutional networks. FUKUSHIMA, K. Neocognitron: A hierarchical neural network capable of visual pattern recognition. Neural networks 1, 2, 119–130. 1988.

[10]. GUPTA, R. K., CHIA, A. Y.-S., RAJAN, D., NG, E. S., AND ZHIYONG, H. 2012. Image colorization using similar images. In ACM International Conference on Multimedia, 369–378. 2012

[11]. IOFFE, S., AND SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML. 2015

[12]. IRONY, R., COHEN-OR, D., AND LISCHINSKI, D. Colorization by example. In Eurographics Conference on Rendering Techniques, 201–210. 2005.

[13]. KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. Imagenet classification with deep convolutional neural networks. In NIPS. 2012.

[14]. Manga colorization. ACM Transactions on Graphics. 2017

[15]. QU, Y., WONG, T.-T., AND HENG, P.-A. Manga colorization. ACM Trans. Graph. 25, 3 (July), 1214–1220. 2006

[16]. Ryan Dahl. Automatic Colorization. January 2016.

[17]. SIMONYAN, K., AND ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. In ICLR. 2015

[18]. S. Iizuka, E. Simo-Serra, and H. Ishikawa. "Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification". ACM Transaction on Graphics (Proc. of SIGGRAPH), 35(4):110, 2016.

[19]. ZEILER, M. D. ADADELTA: an adaptive learning rate method. CoRR abs/1212.5701. 2012.

[20]. ZHOU, B., LAPEDRIZA, A., XIAO, J., TORRALBA, A., AND OLIVA, A. Learning deep features for scene recognition using places database. In NIPS. 2014.