Colorization of Monochrome Images: A CNN based Approach

Vivek Shivkumar Gupta, Tarun Dhirendra Singh, Shreyas Sanjay Walinjkar
Information Technology Department
VCET
Vasai, Palghar, India

***Abstract*—The color information is the strong descriptor of an image and such information are, brightness known as luminance and color known as chrominance. Colorization of images is done manually for a long time. It is mostly done with the help of Adobe Photoshop or various other software. This process is very tedious and time consuming. This is a very difficult task since it is an ill-posed problem that usually requires human intervention to achieve high-quality colorization. A color image has both luminance and chrominance values while a monochrome or Grayscale image has only luminance value. Here we are attempting to develop an automatic process which can produce corresponding chrominance values from given luminance values of the target image. Along with luminance, semantics of an image is important. Semantics define different scenes from image to image and these are categorized into different classes and the target image is colorized with reference to a particular class. Detecting the exact class of the image becomes an important step now and we used an object detection algorithm to identify the class of the target image. Our colorization model focuses on neural network implementation and learning based approach. Initially we used *YUV* color space but with *Lab* color space we obtained better results and employed *Lab* color space and autoencoder architecture in the final model.**

***Index Terms*—Colorization, Yolo Classifier, Lab Colorspace, Convolution Neural Network(CNN)**

I. INTRODUCTION

Colorization is the process of adding color to monochrome images. Automated colorization of grayscale images has been subjected massive research within the computer vision and machine learning communities. Here, we take a statisticallearning-driven approach which helped us towards solving this problem. We design and build a Convolution Neural Network (CNN) that accepts a grayscale images as an input and generates a colorized version of the image as its output in Fig: 1 . The system generates its output which is solely based on images it has learned from in the past, with no further human intervention. CNN is all about self-learning which tries to accurate more and more result. The more you train, the more accurate and top-notch result you obtain. In recent years, CNN has emerged as the factor standard for solving image classification problems, achieving error rates lower than ImageNet Dataset challenge [1]. CNN plays a vital role In the whole software. We can say that CNN is the backbone of the entire system. In recent years, CNN has developed a lot and made a lot of things easier which do not seem possible back then. We have also used YOLO classifier which classifies the object present in the image and from there on the colorization process becomes easy. Classification is the main concern because the if image is not classified correctly then eventually the colorization will fail.

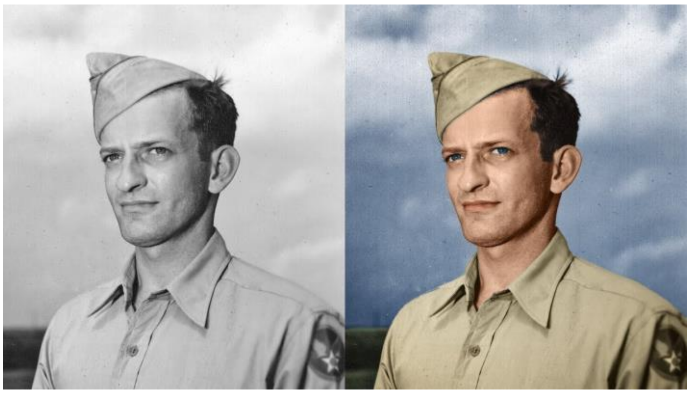


Fig. 1. Grayscale to RGB Image

II. RELATED WORK

A. Scribble Based Colorization

Earlier, colorization process was divided into two parts segmentation and filling. The error ratio is very high in these techniques. Therefore, to reduce this error a lot of human intervention was required. In 2004, Levin et al. [2] proposed a method of assigning a color of pixels based on the similarities of intensities. This method has reduced human intervention and decrease an error ratio. The main limitation of Levin et al. [2] method is that its algorithm is taking a lot of time to give the result. In 2005, Huang et al. [3] proposed a non-iterative method combined with adaptive edge extraction to reduce the colorization technique. The principle for colorization is that neighbouring pixels with similar intensities should have the same color. This was proposed by Levin et al. and further used by Huang et al. [3] Both Levin et al. [2] and Huang et al. [3] used a YUV colorspace, where, *Y* represents a luminance channel and *U* and *V* represent a chrominance value. Here *Y* will be an input and *U* and *’V’* will be output and *U* and *V* will be decided by minimizing the cost function.

B. Example Based Colorization

Example based colorization is transferring a color from reference image to grayscale image. In 2012, Welsh et al. [4] proposed these methods in which the user has to give a referenced image and its algorithm will transfer color from these referenced images to grayscale image. There are 2 ways we can give the referenced image:

1) Colorization Using Web Supplied Example:

To release the users burden of finding a suitable image, Liu et al. [5] and Chia et al. [6] utilize the massive image data on the Internet. Liu et al.compute an intrinsic image using a set of similar reference images collected from the Internet. This method is robust to illumination difference between the targets and reference images, but it requires the images to contain the identical object(s)/scene(s) for precise per-pixel registration between the reference images, and the target grayscale image. Chia et al. [6] propose an image filter framework to distil suitable reference images from the collected Internet images. It requires the user to provide semantic text label to search for suitable reference image on the Internet and human-segmentation cues for the foreground objects.

2) Colorization Using User Supplied Example:

To release the users burden of finding a suitable image, Liu et al. [5] and Chia et al. [6] used the massive image dataset. Liu et al. [5] compute an intrinsic image using a set of similar reference images collected from the Internet. This method is robust between the targets and reference images, but it requires the images to contain identical object(s)/scene(s) for precise per-pixel registration between the reference images, and the target Grayscale image. It requires the user to provide semantic text label to search for suitable reference image on the Internet.

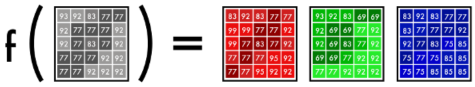
C. Learning Based Colorization

Colorization can be a powerful pretext task for selfsupervised feature for learning, acting as a cross-channel encoder. Consider a grayscale image, if we look it seems less graceful because the picture is not appealing and the color features which are possessed by the objects in it are lost and it seems very hard to digest. If we pay a little close attention at it, we know that certain semantics possess same features like: the sky is typically blue, and the grass is typically green. As we know the prediction of color is free, and we can use any color photo to train the model. The prediction of the colors is multimodal which means several objects can take on several colors. For example, a mango is typically yellow, orange or green but can never be purple. Distribution for each pixel is appropriately model. This helps their model to work on the full diversity of the large scale data on which the model is trained. A final colorization is taken place by annealedmean of distribution. The result colorization which is more vibrant and realistic. Conversion of a grayscale image is very difficult and the objective is to present an image which is appealing to the human eye. Converting a grayscale image colorized image requires a three-dimensional RGB format [2]. The RGB colors required always have the same luminance value but varies in saturation and hue.

III. Experiment

A. Initial Setup

We studied and experimented from reference based colorization to learning based colorization method. As image has 3 layers of colors, Red, Green, Blue which are stacked together, which is converted to *Lab* color space in fig: 2. The logic is simple, only a single reference image is selected and it is converted to *Lab* values, where *L* is luminance value while *a* and *b* are chrominance values. This *L* and *ab* values are input to the simple Convolutional Neural Network model and it is trained with *L* values correspondingly *ab* values. The same image is converted to grayscale image and it is then input for colorization. The *ab* values are generated for corresponding *L* values of the grayscale image. The colorization output improved as we increased the number of epochs during the training of the model. The Major drawback of this setup was, we cannot colorize image other than the input images grayscale version. We need to have need to have almost similar image with respect to input image which is practically difficult to find very similar image.



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

83	92	83	77	77
99	99	77	77	92
99	77	83	77	92
77	77	77	95	92
77	77	95	92	92

93	92	83	69	69
92	69	69	77	92
92	69	83	77	92
69	69	77	92	92
77	77	92	92	92

83	92	83	77	77
83	77	77	77	92
92	77	83	75	85
75	77	75	85	85
75	75	85	85	85

Fig. 2. The mapping function where luminance is mapped into 3 chrominance channels

For colorization we needed broader variety of categories to colorize a specific grayscale image. Our architecture serves multiple example images to the target grayscale image for colorization. Considering the requirement and resource limitation of hardware, the Natural Images Dataset [1] was suitable. This Dataset was developed by 8 effects of degradation on Deep Neural Network architecture, with classes as airplane, car, cat, dog, flower, fruit, motorbike and person with 727, 968, 885, 702, 843, 1000, 788 and 986 images respectively. Out of these images we will use first 100 images for training and next 10 images for testing as initial Images in the dataset for training and colorization. The other dataset used here is ImageNet Dataset [1] for pretrained model of Dark net.

B. Initial Learning Based Setup

The next experimental setup involves clustering of images for different classes and ensemble learning based colorization method. The images are input to the VGG16 model [7] to extract the features from the images, so we can use these features for clustering the images with matching features together. The VGG16 model is classification We used auto encoders as the Final colorization model. The auto encoder works in way by recreating the input. The auto encoders copy the input to output not exactly but approximately. The model has two parts encoder responsible for features extraction and decoder for recreating network and this model is pre-trained on ImageNet dataset [1], the input from those features. the output will be the identified class of the images. We are not interested in getting this output, we require all the important features extracted from the images, so the last layer of VGG16 is removed and the output generated is features from the images. The output generated is *n* dimensional vectors and it is flattened before passing to the clustering.

The histogram analysis method was used to identify the input image belongs to which class and for this all the images in cluster were converted to black and white images. he histogram comparison between the images from clusters and input image returns a value which will be the probability of similarity between the image. We have 4 methods of comparing the histograms of two images, one of them is Correlation method. The highest values from comparison shows the most similarity between the image.

C. Final Experimental Setup

The major drawback of clustering and histogram analysis method was that it misclassified few images to the wrong class. So a better detection of the objects in image was needed and there are a lot of networks and detection algorithms are available. One such network is You Only Look Once (YOLO) version 3 [7] which is better for faster processing and accurate detection of objects in images. The YOLO V3 has 106 layers and requires direct input of 3 color channel image. But the input grayscale image will always have only single color channel, hence we need to make this grayscale image as 3 channel image. The single color channel of the image is replicated 3 times, hence now we have 3 channel version of grayscale image which can be input to the YOLO algorithm. We cross checked multiple times by providing the color and grayscale version of the same image for detection by YOLO V3, no major difference was found.

We used Auto encoders as the Final colorization model. The auto encoder works in way by recreating the input. The auto encoders copy the input to output not exactly but approximately. The model has two parts encoder responsible for features extraction and decoder for recreating the input from those features.

The encoder compresses the input to its latent space representation and the function is represented as *h=f(x).*

The decoder reconstructs the image from the latent space representation and it is represented as *r=f(x).* Here Convolutional Auto encoder Architecture is used, the convolutional layers learn to extract features of image and optimal filters. This convolutional operation over the image results in an activation map which is wrapped around a nonlinear activation function to improve the generalization capabilities of the network. This way the training procedure can learn non-linear patterns in the image. After this, we run a pooling operation on the activation maps to extract dominating features and reduce the dimensionality of the activation maps for efficient computation. The model includes 17 layers of convolution, where 3 layers are UpSampling layer, Table [III-C1]. The input dimensions for input layer is (256, 256, 1) where first two parameters are size of the image and last parameter is luminance value. The resultant parameters are (256, 256, 2) where again first two parameters are size of the output image and last paramters has the two chrominance value.

Using colorspace CIE *Lab* and considering size of image as *H*W*, the Luminance component is $X_L \in R^{H*W*1}$. Our Model will estimate the color component as FULL generated color version $X_L \in R^{H*W*3}$.

So mapping Function of our Model is:

$$F : X_L \to (\widehat{x}_a, \widehat{x}_b) \qquad (1)$$

Where, xb_a,xb_b components of the reconstructed image and X_L is original Lightness components of the reconstructed image and X_L.

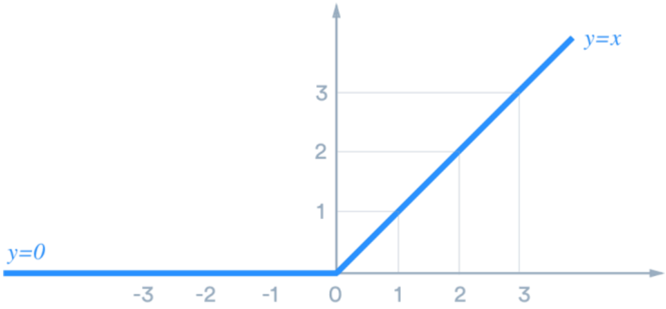


Fig. 3. Relu Activation Function Graph

Mean Square Error (MSE) is the most commonly used regression loss function. MSE is the sum of squared distances between our target variable and predicted values.

$$MSE = \frac{\sum_{i=1}^{n}(y_i - y_i^p)^2}{n} \qquad (2)$$

1) Algorithm:

Given the luminance component of an image, the model estimates *a* and *b* components and combines them with the input to obtain the final estimate of the colored image.

1) Extract components of images as *L* and *ab* components.

2) Feature extraction performed in encoder layer 3) Decoder recreates color image from encoded layer.

4) Model is ensembled for all classes

TABLE I
MODAL SUMMARY

Layer(type)	Output shape
input_1(*InputLayer*)	(None, 256, 256, 1)
conv2d_1(*Conv2D*)	(None, 128, 128, 64)
conv2d_2(*Conv2D*)	(None, 128, 128, 128)
conv2d_3(*Conv2D*)	(None, 64, 64, 128)
conv2d_4(*Conv2D*)	(None, 64, 64, 256)
conv2d_5(*Conv2D*)	(None, 32, 32, 256)
conv2d_6(*Conv2D*)	(None, 32, 32, 512)
conv2d_7(*Conv2D*)	(None, 32, 32, 512)
conv2d_8(*Conv2D*)	(None, 32, 32, 256)
conv2d_9(*Conv2D*)	(None, 32, 32, 128)
upsampling2d1(UpSampling2)	(None,64,64,128)
$\text{conv2d}_1$0(*Conv2D*)	(None, 64, 64, 64)
upsampling2d2(UpSampling2)	(None,128,128,64)
$\text{conv2d}_1$1(*Conv2D*)	(None, 128, 128, 32)
$\text{conv2d}_1$2(*Conv2D*)	(None, 128, 128, 16)
$\text{conv2d}_1$3(*Conv2D*)	(None, 128, 128, 2)
upsampling2d3(UpSampling2)	(None,256,256,2)

IV. RESULTS AND DISCUSSION

An epoch is a measure of the number of times the training vectors are used once to update the weights. In artificial neural network, an epoch means one cycle throughout the complete training dataset. Usually, training a neural network takes quite some epochs. In other words, if we feed a neural network the training data for quite one epoch numerous patterns, we hope for a higher generalization when given replacement unseen input (test data). An epoch is commonly needed with an iteration. Iterations the number of batches or steps through partitioned packets of the training data, needed to finish one epoch.

The Model was trained on 700 images from each class and with the validation split of 0.2, 140 images were selected for validation during the training. We checked results at different epochs 100, 500 and 1000. Good results were achieved on 1000 epochs. Few classes still remained black and white after 100 epochs but showed colorization after 500 epochs. Due to resource limitations, we cannot go beyond 1000 epochs. The colorization results achieved were up to natural color levels. The results shown below are images of person and flowers at different epochs.

On observing the input image belonging to flower class, we can see yellowish tint at 50 epochs and slight colorized effect after 500 epochs. We can see natural colors popping after 1000 epochs.

We can see the same results with person images Fig: ??. A comparison of colorized and original images is shown belonging to person class.

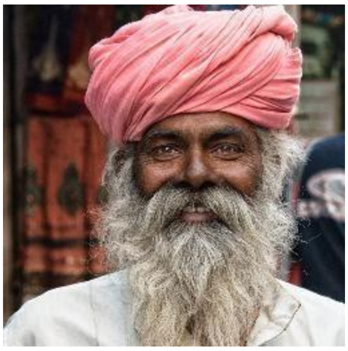


Fig. 4. Original Image

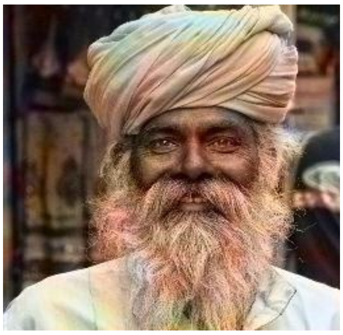


Fig. 5. Color Output Of Person Imag

After so many layers of processing, we saw distortion in colorized images. To regain the quality of images we applied sharpness algorithm mentioned in [9]. We can observe the difference in following images.

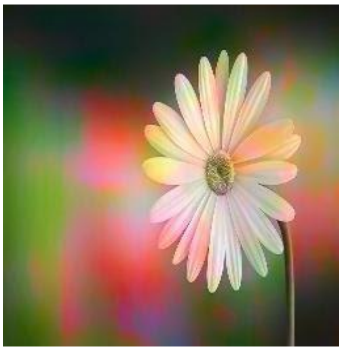


Fig. 6. Original Image

Peak Signal to Noise Ratio (PSNR) is used to measure the quality reconstruction of lossy compression. The signal is the original data, and noise is the error which is introduced by compression. We plotted PSNR values of two classes at different epochs, which can be seen in the graph . As we increase the number of epochs we can see the increase in the PSNR values.

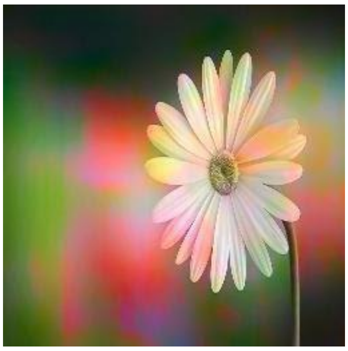


Fig. 7. Image After Sharpening

Higher PSNR generally indicates that the reconstruction of quality images is higher, whereas in some cases it may not.

Psnr equation

$$PSNR = 20\log\frac{MAX_f}{\sqrt{MSE}} \tag{3}$$

whereas MSE is mean square error

Generally, PSNR has been shown to perform poorly compared to other quality metrics when it comes to estimating the image quality and particularly images as perceived.

As we can see an increase in the quality of imgages color wise with respect to increasing PSNR values of 2 classes person and flower in table II and table III

TABLE II
PSNR VALUESOFAPERSON

Epochs	Output1	Output2	Output3
100	24.68	23.82	24.93
500	22.86	23.46	24.98
1000	23.01	22.63	23.64

TABLE III
PSNR VALUESOFA FLOWER

Epochs	Output1	Output2	Output3
100	14.68	15.82	18.93
500	17.86	16.46	18.98
1000	23.01	22.63	23.64

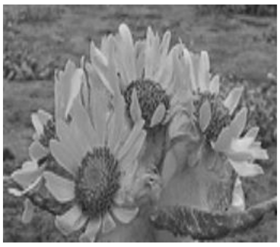


Fig. 9. Grayscale Image

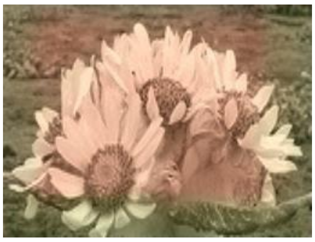


Fig. 10. Colorize Output After 100 Epoch

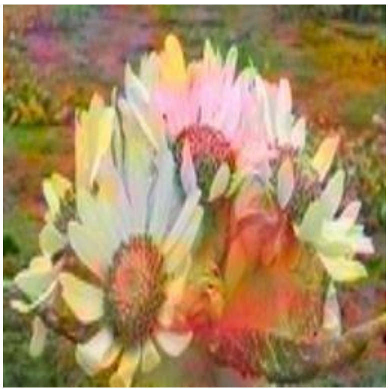


Fig 11. Colorize Output After 1000 epoch

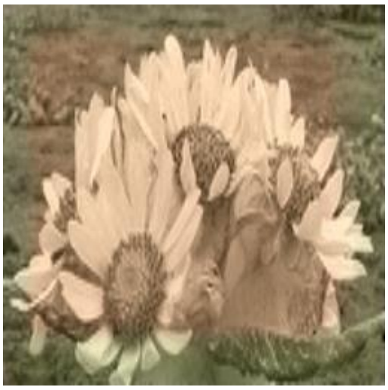


Fig 12. Colorize Output After 500 epoch

V. CONCLUSION AND FUTURE WORK

As of now, we have only used a limited number of classes to categorize and colorize the images. More categories and classes are to be incorporated in this project to produce more accurate results. With the help of more classes and categories the system will be able to detect the object more finely. Better object detection and categorization in image more accurately the result. An effort to obtain more accurate and detailed results are planned. Aim of incorporating real time object detection in image and to produce a colorful output is the next goal of our project.

ACKNOWLEDGEMENT

We owe our deep gratitude to our project guide Dr. Ashish Vanmali, who took keen interest in our project work and guided us all along, till the completion of our project work by providing all the necessary information and guiding us. Without everyone's contribution this project would not have been possible. We Would also like to thank all our family and friends for their support. Also the team members dedication and enthusiasm that has helped us to complete our project.

REFERENCES

[1] J.Deng and W. Dong and R. Socher and Li-jia Li and Kai Li and Li Fei-fei,Imagenet: A large-scale hierarchical image database, 2009.

[2] A. Levin, D. Lischinski, and Y. Weiss, Colorization using optimization, in Proc. SIGGRAPH, 2004, pp. 689-694.

[3] Huang, Yi-Chin and Tung, Yi-Shin and Chen, Jun-Cheng and Wang, Sung-Wen and Wu, Ja-Ling, An adaptive edge detection based colorization algorithm and its applications, in Proc. 13th Annu. ACM Int. Conf. Multimedia, 2005, pp. 351–354.

[4] T. Welsh, M. Ashikhmin, and K. Mueller, Transferring color to greyscale images, ACM Trans. Graph., vol. 21, no. 3, pp. 277-280, Jul. 2002.

[5] X. Liu et al., Intrinsic colorization, ACM Trans. Graph., vol. 27, no. 5, pp–152, 2008.

[6] Chia, A. Yong-Sang and Zhuo, Shaojie and Gupta, Raj Kumar and Tai, Yu-Wing and Cho, Siu-Yeung and Tan, Ping and Lin, Stephen, Semantic colorization with Internet images, ACM Trans. Graph., vol. 30, no. 6, pp–156, 2011.

[7] Simonyan, Karen Zisserman, Andrew.,(2014), ” Very Deep Convolutional Networks for Large-Scale Image Recognition.” [8] J. Redmon and A. Farhadi, ”YOLOv3: An Incremental Improvement”,2018.

[8] A. Vanmali and V. Gadre, ”Visible and NIR image fusion using weight-map-guided Laplacian - Gaussian pyramid for improving scene visibility”, Vol. 42, No. 7, July 2017, pp. 1063-1082.

[9] Z. Cheng, Q. Yang and B. Sheng, ”Colorization Using Neural NetworkEnsemble,” in IEEE Transactions on Image Processing, vol. 26, no. 11,pp. 5491–5505, Nov. 2017.

[10] Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin,Image analogies, in Proc. 28th Annu. Conf. Comput. Graph. Interact.Tech., 2001, pp. 327-340.

[11] Reinhard, M. Adhikhmin, B. Gooch and P. Shirley, Color transferbetween images, IEEE Comput. Graph. Appl., vol. 21, no. 5, pp. 34-41,Sep./Oct. 2001.

[12] R. Irony, D. Cohen-Or, and D. Lischinski, Colorization by example, inProc. Eurograph. Symp. Rendering, vol. 2. 2005, pp. 201-210.

[13] S. Lee, S. Park, P. Oh and M. G. Kang, Colorization-Based CompressionUsing Optimization, in IEEE Transactions on Image Processing, vol. 22,no. 7, pp. 2627–2636, July 2013.

[14] R. K. Gupta, A. Y.-S. Chia, D. Rajan, E. S. Ng, and H. Zhiyong,Image colorization using similar images, in Proc. 20th ACM Int. Conf.Multimedia, 2012, pp. 369-378.

[15] Cheng, Zezhou and Yang, Qingxiong and Sheng, Bin. Deep colorization.In: Proceedings of the IEEE International Conference on ComputerVision. (2015) 415-423.