Grayscale Portrait Colorization using CNNs and Pretrained VGG-Face Descriptor

Naimul Haque
Manarat International University
Department of Computer Science and Engineering
Dhaka, Bangladesh
naimul011@gmail.com

Samin Shahriar Tokey
Southeast University
Department of Computer Science and Engineering
Dhaka, Bangladesh
saminshahriar@rocketmail.com

***Abstract*—Colorization problem is a process of adding colors to a grayscale image. Traditionally it requires human-labeled color scribbles on the grayscale image and the image is colored by propagating the scribbled colors throughout the image using optimization techniques. It is not very long ago that colorization without the intervention of human was almost impossible. Recently colorization using Neural Networks has achieved big success due to the availability of huge datasets. This paper attempts to color only portrait images using Convolutional Neural Networks along with a pretrained VGGFace descriptor as a global feature extractor. It has been shown that using separate networks, one to extract 3x3 patch sized features or local features and another to extract global features, result in higher quality of colorization outputs. In portrait colorization, we used VGGFace descriptor as the global feature extractor that encodes a grayscale portrait into a vector representation which later can be used to train the system to have overall coloring effects on the grayscale portrait using fusion technique. To measure the colorfulness of the output image we introduce the use of colorfulness metric that has the potential to capture the perceptual colorfulness essence from the output images. To evaluate the model performance we used Mean Squared Errors (MSE) and Mean Absolute Errors (MAE).**

I. Introduction

Colorization of historical black and white photos not only fade away years of separation but also give new insights about those photos since color carries a great deal of information. The colorization requires segmentation of the image into different regions and putting appropriate colors for them. Conventionally, the process initiate with human-labeled color scribbles on the grayscale image and the colors are then propagated over different regions throughout the image using optimization techniques.

The Colorization problem is a difficult problem that requires extensive labor of adding colors to grayscale that has never been seen in colors. The experts have to know the history of the image and must know the colors of the objects that are seen on the image before coloring. Despite the struggle, researchers have tried to automate the colorization after some human color labels are given on an image. Fully automated Colorization has been done only in the recent years with the help of Neural Networks which is done by training the Deep Neural Networks (DNNs) [1] with huge amounts of reference images collected from ImageNet [2]. In recent attempts, the coloring architecture uses two CNNs: one as an Encoder-Decoder, the other is to extract the global features from the whole image to give an overall coloring on the image. This technique is first introduced in the Let there be Color! [3], where the second network is an object classifier. It is seen that having a piece of prior knowledge about what is seen in the image scene results in better coloring. In a similar attempt, Deep Koalarization [4] used Inception-resnet-v2 as the global feature extractor. Inception-resnet-v2 is Google's of one the latest object classifiers, which is trained and ready to use. This paper shows a similar approach using VGGFace to color only portrait images.

With the advance of machine learning and deep learning, researchers took the automation of colorization more seriously [1] [5] [3]. In the recent years, transfer learning [6] of deep learning opened up a new window for researchers to color grayscale image not just with a single Convolutional Neural Network (CNN) but also with a pretrained classification model, acting as a global feature descriptor. In recent past, coloring images with Inception-ResNet-v2 [7] have given commendable output. In this paper, we demonstrate the use of pre-trained model VGGFace [8] as global feature extractor and measure the perceptual colorfulness of the resultant colored images using colorfulness metric [9], in Colorization of the black and white portrait which is unprecedented. We also evaluate our results using Mean Squared Errors (MSE) and Mean Absolute Errors (MAE), compare the results with the previous approach explain in [4] and by using different pretrained model VGG19 [10] as global feature extractor.

In Section II, the approach is broken down, discussing the color space (CIELAB), pretrained model VGGFace, architecture of the model and about evaluation metrics that we used to measure the output performance. In Section III, the results are discussed along with the colorfulness interpretation and the performances of the CNNs are evaluated using different metrics.

978-1-7281-5842-6/19/$31.00

II. Approach

A. CIE L*a*b* (CIELAB) color space

CIE L*a*b* (CIELAB) [15] is a color representation method that reduces the number of color channels to 2 whereas RGB color space has 3 color channels. It puts us in the advantage to predict only 2 color channels instead of 3. In total there are 3 components in this colorspace. The first of the three components of CIELAB represent the luminance of the pixel, L* = 0 represent black and L* = 100 indicates diffuse white. The two other components dictate the chrominance of a pixel: negative values of the component a* indicate green while positive values indicate magenta while negative values of b* indicate blue and positive values indicate yellow.

L*a*b* color space has been used in colorization problems because the problem with RGB is that, there is no separate luminance channel. Using this color space, we can split the image into components and use the luminance (L*) component to predict the other two chrominance components (a* and b*) which makes the problem much easier by reducing overhead. Another reason is that CIELAB is designed to be perceptually similar to how humans perceive colors, thus have the quality to be modeled with neural networks.

B. Dataset

The portrait images that we used to train our model is gathered from Helen dataset [14] which is an effort to build a facial feature localization algorithm. The images contain mostly human faces with an arbitrary background. The images are in variable sizes and in jpg format. We gathered 2000 images for a training set and 800 for the validation set. For our use, we resized the images into 256×256 dimension for CNNs. In case of using VGGFace descriptor, the images are resized to 224×224.

C. VGG-Face descriptor

VGG-Face [8] descriptor is Convolutional Neural Network (CNN) which generates a vector coding for a subjects identity. This CNN is trained on 2.6M face images from 2.6k different people. The descriptor takes an image of 224×224 dimension and output a vector of 2622 softmax values. The vector is unique marker for each portrait image and it tells the probabilities of how similar the subject is to the 2.6k identities.

D. Architecture

The architecture is similar to Inception-ResNet-v2 [3] except for slight changes. The Figure 1 shows the architecture of our model. The architecture consists of four main components. The encoding and global feature extraction components extract the local features and global features, respectively. New feature layers are obtained by fusing output of encoding and global feature extraction in the fusion component. Then the new features are used in decoding to predict the output.

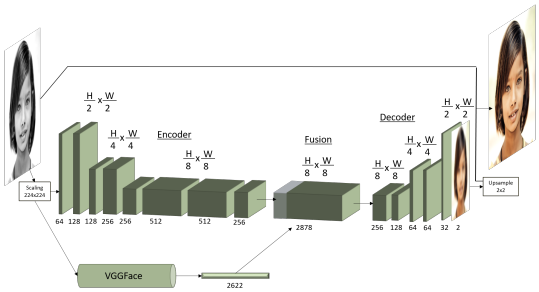

Fig. 1. Portrait Colorization CNN Architecture with VGG-Face descriptor as the global feature extractor.

1) Encoder / Local Feature Extractor: Let, the input grayscale portrait has the dimension of $N \times M$. The encoding network takes this image and outputs a $\frac{N}{2^3} \times \frac{M}{2^3} \times 256$ using convolution process where the information of the image encoded in such volume representation that can be trainned using optimizer. The dimension of the input image is reduced 8 times by dividing the image into half 3 times using a stride of 2.

TABLE I
Encoder for Portrait Colorization

Layer	Kernels	Stride	Activation
Conv	64 x (3x3)	2x2	ReLU
Conv	128 x (3x3)	2x2	ReLU
Conv	256 x (3x3)	2x2	ReLU
Conv	512 x (3x3)	1x1	ReLU
Conv	256 x (3x3)	1x1	ReLU

2) Global Feature Extractor: As the Figure 1 shows VGG-Face which is our global feature extractor encodes rescaled image of 224×224 dimension and output a vector of size 2622.

3) Fusion Layer: The fusion layer then takes 2622 sized vector and repeats it $\frac{MN}{2^6}$ times to create a volume of $\frac{N}{2^3} \times \frac{M}{2^3} \times 2622$. Adding this to the output of the encoding layer results a feature representation of $\frac{N}{2^3} \times \frac{M}{2^3} \times 2878$. Then, this volume is convolved with 256 kernels of size 1×1 to get $\frac{N}{2^3} \times \frac{M}{2^3} \times 256$ output.

TABLE II
Fusion for Portrait Colorization

Layer	Kernels	Stride	Activation
Fusion	-	-	
Conv	256 x (1x1)	1x1	ReLU

4) Decoder: The decoder takes this volume and after series of convolutional and up-sampling layers in order to obtain the final dimension of $M \times N \times 2$. Adding this output in the initial grayscale image of $M \times N$, we obtain the final output image of $M \times N \times 3$.

TABLE III
DECODER FOR PORTRAIT COLORIZATION

Layer	Kernels	Stride	Activation
Upsamp	-	-	-
Conv	128 x (3x3)	1x1	ReLU
Conv	64 x (3x3)	1x1	ReLU
Upsamp	-	-	-
Conv	32 x (3x3)	1x1	ReLU
Conv	16 x (3x3)	1x1	ReLU
Conv	2 x (2x2)	1x1	tanh
Upsamp	-	-	

E. Evaluation Metrics

1) Colorfulness Metric: To measure the colorfulness of the image, we use the colorfulness metric [9]. The metric quantifies the overall colorfulness in a natural image. This is an approximation of the colorfulness based on human perception. The metric is built by asking people to rate images using 7 categories of colorfulness and then correlating these categories with 90% of the experimental data. Table IV shows the attributes and their corresponding colorfulness value in this metric.

TABLE IV
CORRESPONDENCE BETWEEN THE COLOURFULNESS METRIC, AND THE COLOURFULNESS ATTRIBUTES

Attribute	Colorfulness Value
not colorful	0
slightly colorful	15
moderately colorful	33
averagely colorful	45
quite colorful	59
highly colorful	82
extremely colorful	109

The metric can be used to get the sense of the colorfulness difference of predicted colored images from the ground truth images.

$$\Delta M_i = M(h(x_i)) - M(y_i) \tag{1}$$

where $M(x)$ [9] is the colorfulness function and $h(x)$ is output mapping mapping function for y.

2) Mean Squared Error (MSE): One of the metric to evaluate this kind of regression problem is to measure the Mean Squared Error(MSE) between predicted outputs and target outputs.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - h(x_i))^2 \tag{2}$$

The MSE metric can give us good insights about how well the model is working in respect to coloring but it lacks to measure the perceptual colorfulness.

3) Mean Absolute Error (MAE): The Mean Absolute Error(MAE) is calculated taking the mean of the absolute errors better the predicted outputs and the targets.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - h(x_i)| \tag{3}$$

Similar to MSE, this metric lacks to capture the perceptual colorfulness from the image, yet it is very to to evaluate our output images in simple numbers.

F. Training

The network is trained on 2000 training images for approximately an hour on Tesla K80 GPU. A batch size of 50 for each epoch of learning using Adam optimizer and is trained for 21 epochs with 40 iterations in each epochs. MSE loss function is used as the cost function to train the model using adam optimizer.

1) Batch Normalization: The batch normalization is added after every convolution layer. It normalize the activations of the previous layer at each batch that maintains the mean activation close to 0 and the activation standard deviation close to 1. Adding this norlization to the hidden layers, in neural network, can speed up learning significantly by reducing Internal Covariate Shift (ICS) [12] as it is one of the barrier for very deep neural networks [13] to be trained without getting overfitted.

2) Adam optimization: The network is trained using Adam optimizer [11] which is an optimization algorithm, is a variant of Stochastic Gradient Decent (SGD), to update network weights and biases based on the cost function. Stochastic gradient descent maintains a single learning rate (termed alpha) for all weight updates and the learning rate does not change during training. The method is combination of two other variants of SGD namely Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). The Figure 2 shows the MSE loss for each epoch for up to 21 epochs.

III. RESULTS

The Figure 3 shows the result of colorization grayscale image at the leftmost, the colorized image in the middle while ground truth image at the right.

The Figure 4 shows the color distribution of 3 color channels of colorized and ground truth image. The colorfulness of the colorized image and the actual image is approximately 26 and 47, respectively.

The average colorfulness is calculated from the test set of 799 output color images. Table V show colorfulness of resultant images when colorization is done separately for VGGFace, Inception-Resnet-V2 [4] and another pretrainned model VGG19 which performed well in terms of perceptual colorfulness. The colorfulness result of the output images shows that VGG19 results are more colorfulness and closer to

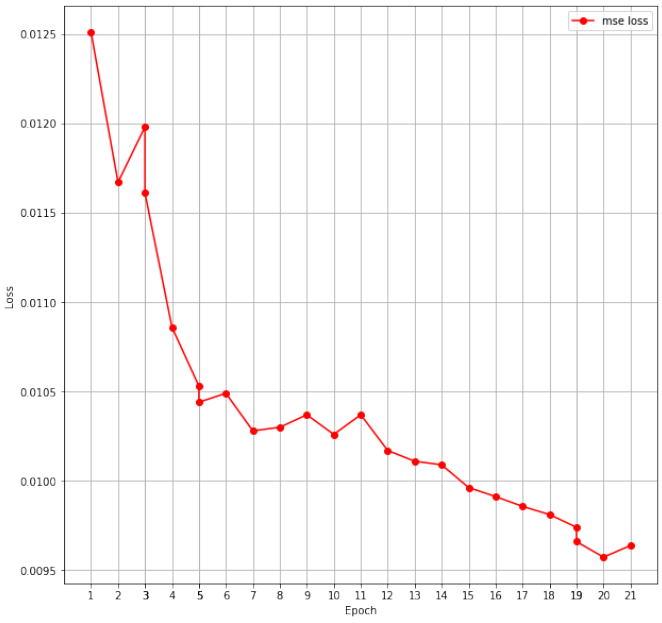

Fig. 2. The loss vs epoch diagram for 21 total epochs.

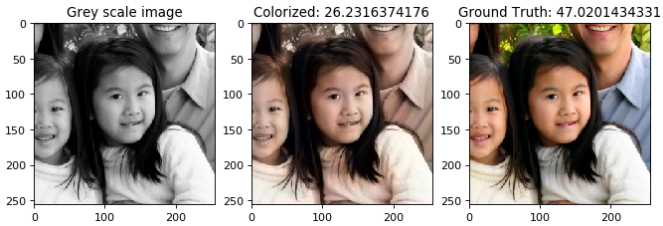

Fig. 3. A result of colorization with VGGFace.

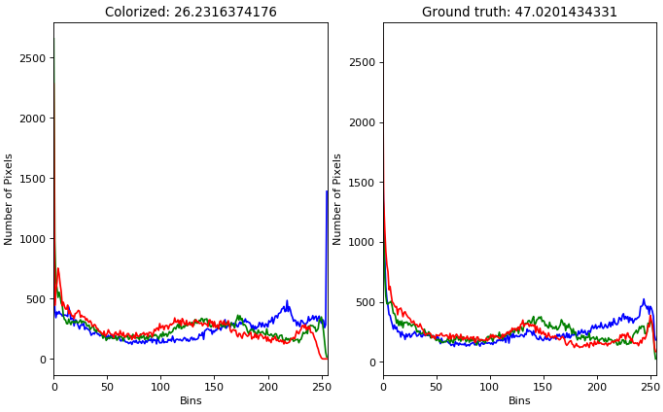

Fig. 4. Color distribution of the results of VGGFace.

the ground truth images on the basis of perceptual colorfulness of the outputs.

To evaluate the final results two other metrics, MSE and MAE, are used which can calculate the pixel wise errors in the final output images compared to the ground truth images. The colorfulness metric is also used according to the recommended procedure mentioned in [9]. The Table VI summarizes the performance of our method compared to methods where Inception-resnet-v2 [4] and VGG19 [10] as global feature extractor. From the table, it can be inferred that VGGFace performs slightly better than Inception-Resnet-v2 and VGG19 in respect to MSE and MAE metric. In case of colorfulness metric (ΔM), which is caluculated by following formula:

TABLE V
AVERAGE COLORFULNESS OF 799 OBTAINED USING INCEPTION-RESNET-V2 VSVGG19 VS VGGFACE

Global feature extractor	Colorfulness Value
Inception-resnet-v2	22.09
VGG19	28.89
VGGFace	25.10
Ground Truth	41.87

$$\overline{\Delta M_i} = \frac{1}{n}\sum_{i=1}^{n}\Delta M_i \tag{4}$$

VGG19 gives more colorfulness images closer to ground truth images but this does not imply that VGG19 performs better in coloring grayscale images pixel-wise.

TABLE VI
MSE, MAE AND $\overline{\Delta M_i}$ OF 799 OBTAINED FOR INCEPTION-RESNET-V2, VGG19 AND VGGFACE

Global feature extractor	MSE	MAE	$\overline{\Delta M_i}$
Inception-resnet-v2	374.23	12.06	-19.78
VGG19	387.15	12.71	-12.98
VGGFace	368.57	11.88	-16.77

The Figure 5 shows the comparison of the results using different pretrained models. The figure further confirms the bad performance of Inception-Resnet-v2 in case of coloring only portrait images, the perfomance of the VGG19 and VGGFace is quite similar with nuanced differences. The more coloring effects are seen in the output images of VGG19 model but still the metrics suggest the VGGFace model's performance is better with respect to ground truth images.

IV. CONCLUSION

The CNNs that are used have lots of layers and many filters in each layer that results in a very large number of parameters to be trained. The networks are easily over-fitted if the size of the dataset is little. The lack of resources compelled us to narrow down our problem domain and work on coloring portraits from the Helen dataset only.

The results obtained had brownish hue which is also known as the 'Sepia' effect across L*a*b* color space. One of the reason is that the CNNs are overfitted to the training data and more data is needed which should also show the overall hidden layers of the VGGFace descriptor.

Fig. 5. Comparisons among the coloring results for different global feature extractors. At the leftmost is the grayscale image,then outputs from Inception-Resnet-v2, VGG19, VGGFace and finally Ground Truth images are shown rightmost side of the figure.

The results obtained are not as good as previous works are concerned. This is because of our smaller dataset that might be overfitting our networks as they are very large. In the future, we look forward to training the Networks with larger datasets.

The value of MSE and MAE of the pretrained models VGGFace is less than that of Inception-Resnet-v2 and VGG19 which that VGGFace has the potential to be used for portrait colorization. The analysis from the outputs images of different pretrained trained models suggest the VGGFace has the potential to perform with higher accuracy if the dataset can be increased.

The transfer learning technique, which is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task, helped researches to use separate models to predict the little patch size features as well as features which are global in an image. A CNN object classifier such as Inception-resnet-v2 can be used as global feature descriptor as it tells the decoding network what objects are seen on the image. The decoding network can be trained to add a globally embedded coloring effect on the image, e.g. if the background of the scene has a sky the decoding network color the image with a bluish hue. We extended this idea and used a facial descriptor VGGFace [8] as the global feature extractor for the portrait colorization. Using the facial vector representation vector from the descriptor networks can be trained to color based on ethnic groups, gender, and age. The output layer of the VGGFace model can stripped off and and extend the model can be trained to fine tuned on only limited numbers of target features using softmax. This fine-tuned model can remove unnecessary coloring information or noise in the VGGFace model's output vector. In future, we are looking forward to do that and measure the performance of the output images.

The loss vs epoch diagram in Figure 2 show the training can be further extended to higher epochs since the any kind of plateau is not visible in the optimization. In future, we are looking for to train the model in larger epoch size with larger dataset.

REFERENCES

[1] Cheng, Z., Yang, Q., Sheng, B. i, *Deep Colorization*, Proceedings of the IEEE International Conference on Computer Vision, 2015.
[2] J. Deng et al, *ImageNet: A Large-Scale Hierarchical Image Database*, CVPR09. 2009
[3] Satoshi Iizuka, Edgar Simo-Serra, Hiroshi Ishikawa, *Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification*,ACM Transactions on Graphics (Proc. of SIGGRAPH 2016), 35.4 (2016).
[4] Lucas Rodes-Guirao, Federico Baldassarre, Diego Gonzalez-Morin, *Deep-Koalarization: Image Colorization using CNNs and Inception-ResNet-v2*, ArXiv:1712.03400 (Dec. 2017).
[5] Zhang, Richard and Isola, Phillip and Efros, Alexei A, *Colorful Image Colorization*, ECCV, 2016.
[6] Peizhong Liu, Jing-Ming Guo, Chi-Yi Wu, Danlin Cai, *Fusion of Deep Learning and Compressed Domain Features for Content-Based Image Retrieval*, IEEE Transactions on Image Processing, 2016.
[7] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex A. Alemi, *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*, ICLR 2016 Workshop.
[8] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*, British Machine Vision Conference (2015).
[9] David Hasler, Sabine E. Suesstrunk, *Measuring colorfulness in natural images*, Proc.SPIE, 5007, 5007 - 5007 - 9, 2003.
[10] Simonyan, Karen and Zisserman, Andrew. *Very Deep Convolutional Networks for Large-Scale Image Recognition*, CoRR abs/1409.1556 2014.
[11] Kingma, Diederik Ba, Jimmy, *Adam: A Method for Stochastic Optimization.*, International Conference on Learning Representations. 2014.
[12] Sergey Ioffe, Christian Szegedy, *Batch normalization: accelerating deep network training by reducing internal covariate shift*, ICML'15 Proceedings of the 32nd International Conference on International Conference on Machine Learning, Volume 37, Pages 448-456
[13] Simonyan, K. Zisserman, A. *Very Deep Convolutional Networks for Large-Scale Image Recognition.* CoRR, abs/1409.1556. 2014.
[14] Helen dataset, url=ifp.illinois.edu/vuongle2/helen/, Accessed: 29.10.2018
[15] Luo, Ming Ronnier, *CIELAB. Encyclopedia of Color Science and Technology.*, Springer Berlin Heidelberg, (2014).