Atlantis Highlights in Intelligent Systems (AHIS), volume 1
3rd International Conference on Control, Automation and Artificial Intelligence (CAAI 2018)

Facial Colorization Using Transfer Learning

Fucheng You, Yangze Zhao*, Shuren Lai and Hechen Gong
Beijing Institute of Graphic Communication. No. 1, Xinghua Street (two section), Daxing District, Beijing,China
*Corresponding author

***Abstract*—This paper applies transferring learning to facial colorization and The main body is based on DNN. However, considering the less complexity of the dataset and the equipment constraints. We extracte global features from VGG16 pre-tained model instead of Inception-resnet-v2. In adition ,we cmpare the test results with the colorful images and assess the acceptance of the generated image.**

Keywords—facial colorization; transfer learning; CNN; feature extraction

I. INTRODUCTION

At the beginning, colorization refers to the process that transfer a monochrome into a full-color image. Nowadays, it can also mean old black-and-white video coloring, which has greatly expanded its usage. Coloring hand painting manually is a time-consuming and laborious process because we have to carefully choose colors of different positions in an image. And it has been the preserve of human artists to make image real. Advances in technology have made it possible for machines to do the same. There are about three coloring techniques: scribble-based colorization[10,13] ,example-based colorization[1,2] and neural network colorization[6,7,8].The first one requires to scribble on the the target grayscale images. Although it has a good performance on small samples, coloring a big number of images may be time-consuming, particularly for amateur. In terms of this issue,[3] put up with an example-based technology and then was adjusted bu [1, 11].

The second technology transfers the color information from same category to the target monochrome image. Unlike scribble-based colorization methods, the example-based methods transfer the color information from a reference image to the target grayscale image. The example-based colorization methods can be further separated into two categories according to the source of reference images: Nevertheless, looking for a similar and appropriate image may not be easy.[14] sim plies the process by means of making use of the image data on the Internet and propose filtering methods and propose filtering schemes to select suitable reference images. Obviously, they have the same discourages. This method demands identical Internet object for precise per-pixel registration. In a word ,images that match each other have to be limited with a rigid shape.

A fully-automatic method is proposed to deal with the issue. Initially, as there are not enough details in one image,[4] requests similar reference images. But, the matching noise heavily affects the final performance once a large-scale database is adopted in practice. Then DNN(deep neural network) was applied to solve this problem. It has shown powerful learning ability that even outperforms human in some fields and learning methods have been proved by experiments[7].It has been applied in image classification[5,12],pedestrian detection, image super-resolution, photo adjustment etc. This motivates us to explore its potential application in our context. Our work treats transferring color as a regression problem and DNN architectures are applied in this paper We guess colorization may improve the performance on face recognition.. Restricted to hardware,a small amount of face pictures was applied in our experiment. And because of the low-level amount of pictures and just one category ,we replaced the Resnet with VGG16.In order to optimize the network, we make use of the advantages of batch-normalization and Dropout[18,19,23].Finally, our neural network can complete colorization task. Although the training is very slow due to the adoption of a large database, the learned model can be directly used to colorize a target grayscale image efficiently.

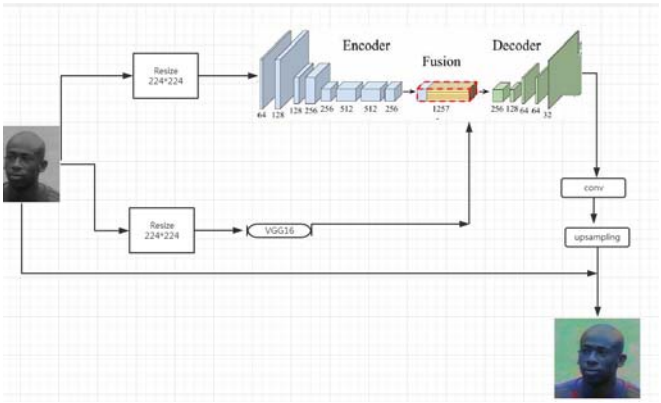

FIGURE I. THE MAIN ARCHITECTURE

II. MAIN ARCHITECTURE

Figure 1 present the main architectures of our work. Just like other deep learning approaches[20,21] ,after pre-processing, the images are fed in neural network. The proposed method has two major steps: training a neural network using a large set of example of reference images and using the learned neural network to colorize a target grayscale image. The main approach changes the color channel from RGB to Lab. Because Lab space only has two color layers, which means that we can use the original grayscale image in the prediction, and only need to predict two channels at the same time. In addition, about 94% percent of the cells in the human eye detect brightness, with only 6percent sensing color. That is another reason that we need to keep grayscale layer. In order to convert one to two, we need to use the convolution filter. Just thing of them as blue/red polarizations in 3D glasses. Next several paragraphs in this part, A will introduce the preprocessing .B and C will talk about the

Encoder and Feature Extractor. At last ,D and E will describe the Fusion and Decoder respectively.

Layers	kernels	stride	Layer	kernels	stride	layer	kernel	stride
Conv	64*(3*3)	2*2	Fusion			conv	128*(3*3)	1*1
Dropout(0.4)								
conv	128*(3*3)	1*1	Conv	256*(1*1)	1*1	upsamp		
Dropout(0.5)								
conv	128*(3*3)	2*2				conv	64*(3*3)	1*1
Dropout(0.5)								
conv	256*(3*3)	1*1				conv		
Dropout(0.6)								
conv	256*(3*3)	2*2				Upsamp	32*(3*3)	1*1
Dropout(0.6)								
conv	512*(3*3)	1*1				Conv	2*(3*3)	1*1
Dropout(0.7)								
conv	512*(3*3)	1*1				conv		
Dropout(0.6)								
Conv	256*(3*3)	1*1				Upsamp		

FIGURE II. MAIN PARAMENTERS

A. *Preprocessing*

We first delete some broken images and mismatching images that which owns some similar features, like size and full black or white pixels with code writing by ourselves .To ensure correct learning, the pixel values of all three image components are scaled in order to obtain values within the intervalof [0, 1].The formula is as follows:

$$Xnorm = \frac{\mathrm{X}-\mathrm{Xmin}}{\mathrm{Xmax}-\mathrm{Xmin}} \tag{1}$$

Xnorm denotes the final data. X is the raw data. Xmax and Xmin refer to the maximum value and the minimum value.

B. *Encoder*

The Encoder process H*W gray-scale images and outputs a resized feature representation. And we uses convolutional layers with 3*3 kernels. Padding is used to preserve the layer's input size. Furthermore, the concrete information is in Table1.To avoid overfitting in layers which have major nodes, we add Dropout layers to randomly delete some hidden neurons in the network, keep input and output neurons unchanged. As for the probablities of remaining the nodes ,we choose numerical value consistent with the number of layer nodes.

C. *Feature Extractor*

Like many other papers[16],this paper classifies features into local feature and global features. Local features extract independent image blocks from the image. In general, the firststep is to extract some space-time interests and then extract corresponding image blocks. Finall, we combine these image blocks. The advantage of local feature is that it does not depend on the segmentation and localization and tracking of human body at the bottom, and it is not very sensitive to noise and occlusion. However, it needs to extract enough stable interest points related to the action category, so it requires a lot of preprocessing. Global characteristics[22, 23] is interested in the detected the whole of the image, typically by background subtraction figure or tracking methods. But These features are sensitive to noise, partial occlusion, and changes in perspective. We scale the input image into 224*224.To extract mid-level features with a respectively simple method, we use a pre-trained VGG16 model. This results in a 1001*1*1 embeding.

D. *Fusion*

Fusion layer duplicate the feature vector from VGG16 256 times and then attaches it to the feature volume outputted by the encoder along the depth axis. This method was introduced by [16,17,18] and is shown in FIGUTRE 1. This approach obtains a single volume with the encoded image and the mid-level features of shape H/8*W/8*1257. By mirroring the feature vector and concatenating it several times we ensure that the semantic information conveyed by the feature vector is uniformly distributed among all spatial regions of the image. Moreover, this solution is also robust to arbitrary input image sizes, increasing the model feasibility. Finally, we apply 256 convolutional kernels of size 1*1, ultimately generating a feature volume of dimension.

E. *Decode*

Finally, the decoder takes this H/8*W/8*256 volume and applies a series of convolutional and up-sampling layers in order to obtain a final layer with dimension H*W*2. Up-sampling is performed using basic nearest neighbor approach so that the output's height and width are twice the input's.

III. EXPERIMENT

The experiment can be divided into two parts: training and test. There is a small difference between training and test experiment. In the training process, in order to reduce outfitting we adjust Dropout method, which is widely used in many experiments and proved to be really useful. We will discuss the details about the experiment below.

A. *Loss Function*

We adopt MSE(mean-square error) as loss objection. Our goal is to optimal the model parameters until loss function reaches a goal. One reason for which we choose MSE is that we want to quantify the loss between the predicted pixels colors and their true values. The main formula:

$$\mathrm{c}(\mathrm{X},\theta) = \frac{1}{2HW}\sum_{k\in\ (\mathrm{a,b})}\ \sum_{i=1}^{H}\sum_{j=1}^{W}(\mathrm{X}_{ki,j}-\check{X}_{ki,j})2 \tag{2}$$

θ denotes all model parameters, Xk,j and X˘_(ki,j) denotes the predicted pixel value of the k:th component of the target and reconstructed image, respectively.

B. *Dropout*

The term "dropout" refers to dropping out units (both hidden and visible) in a neural network. It means the model will be simplified by thi way.In general ,we just applied Dropout method to train g process and we keep the layer nodes according their depth and node numbers. In fact dropout can effectively reduce the occurrence of over-fitting and achieve the effect of regularization to a certain extent.

C. Training and Test

Of the approx 700 original images, we held out the 10% to be used as validation data during training. The results presented in this report are drawn from this validation set and therefore the network never had the chance to see those images during training Adam optimizer was used during approximately 10 hours of training. Complete details about the architecture, the image processing pipeline and our implementation in Keras and Tensor Flow. The network was trained and tested using the Tegner nodes of The PDC Center for High-Performance Computing at the KTH Royal Institute of Technology, leveraging the NVIDIA CUDA Toolkit and the NVIDIA Quadro M2000 Accelerator GPU to speed up the computations. A batch size of 16 ruled out the risk of overflowing the GPU memory.

Once trained, we fed our network with some images. The results turned out to be quite good for some of the images, generating near-photorealistic pictures. However, due to the small size of our training set our network performs better when certain image features appear. For instance, natural elements seem to be well recognized ,such as follows. In the woman picture ,there is a lot of noise in the final result. Original image selection have a strong impact on the performance. And our dataset is pretty small owing to equipment constraints. Another import reason is the parameters optimization which is the main direction we are going to study. Next we will aim at improving the speed improvement by adding batch normalization.

FIGURE III. RAW IMAGE A

FIGURE IV. RESULT A

FIGURE V. RAW IMAGE A

FIGURE VI. RESULT B

IV. CONCLUSION

This project validates that an end-to-end deep learning architecture could be suitable for some image colorization tasks. In particular, our approach is able to successfully color high-level image components. Nevertheless, the performance in coloring small details is still to be improved. As we only used a reduced subset of whole images, only a small portion of the spectrum of possible subjects is represented, therefore, the performance on unseen images highly depends on their contents. To overcome this issue, our network should be trained over a larger training dataset. We believe that a better mapping between luminance and a*b* components could be achieved by an approach similar to variation auto encoders, which could also allow for image generation by sampling from a probability distribution.

Finally, it could be interesting to apply colorization techniques to video sequences, which could potentially re-master old documentaries. This, of course, would require adapting the network architecture to accommodate temporal coherence between subsequent frames

Overall, we believe that while image colorization might require some degree of human intervention it still has a huge potential in the future and could eventually reduce hours of supervised work.

ACKNOWLEDGMENT

This work was partially supported by Joint Funding Project of Beijing Municipal Commission of Education and Beijing Natural Science Fund Committee (KZ201710015010), Project of National Scientific Found (No.61370188), Project of Beijing Municipal College Improvement Plan (PXM2017_ 014223_ 000063) and New Project of Green Printing and Publishing Technology by Cooperative Creating Center (PXM_ 014223_ 000025) and BIGC Project (Ec201803 Ed201802 Ea201806)

REFERENCES

[1] G. Charpiat, M. Hofmann, and B. Sch¨olkopf. Auto Automatic image colorization via multimodal predictions. In ECCV, pages 126–139. Springer, 2008. 1, 2, 6, 7

[2] A. Y.-S. Chia, S. Zhuo, R. K. Gupta, Y.-W. Tai, S.-Y.Cho, P. Tan, and S. Lin. Semantic colorization withinternet images. In TOG, volume 30, page 156. ACM,2011. 1, 2, 5, 6

[3] D. Comaniciu and P. Meer. Mean shift: A robustapproach toward feature space analysis. PAMI,24(5):603–619, 2002. 7

[4] Cheng Z, Yang Q, Sheng B. Deep Colorization[J]. 2016.

[5] Baldassarre F, Morín D G, Rodésguirao L. Deep Koalarization: Image Colorization using CNNs and Inception-ResNet-v2[J]. 2017.

[6] Zhang R, Isola P, Efros A A. Colorful Image Colorization[C]// European Conference on Computer Vision. Springer, Cham, 2016:649-666.

[7] Larsson G, Maire M, Shakhnarovich G. Learning Representations for Automatic Colorization[M]// Computer Vision – ECCV 2016. Springer International Publishing, 2016:577-593.

[8] Yarin Gal, Zoubin Ghahramani. Dropout as a Bayesian approximation: representing model uncertainty in deep learning[J]. 2015:1050-1059.

[9] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.

[10] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the Inception Architecture for Computer Vision[C]// Computer Vision and Pattern Recognition. IEEE, 2016:2818-2826.

[11] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the Inception Architecture for Computer Vision[C]// Computer Vision and Pattern Recognition. IEEE, 2016:2818-2826.

[12] Welsh T, Ashikhmin M, Mueller K. Transferring color to greyscale images[J]. Acm Trans Graphics, 2002, 21(3):277-280.

[13] Guillaume Charpiat, Matthias Hofmann, Bernhard Schölkopf. Automatic Image Colorization via Multimodal Predictions[J]. Lecture Notes in Computer Science, 2010, 5304:126-139.

[14] Deshpande A, Rock J, Forsyth D. Learning Large-Scale Automatic Image Colorization[C]// IEEE International Conference on Computer Vision. IEEE Computer Society, 2015:567-575.

[15] Tuzel O, Porikli F, Meer P. Region Covariance: A Fast Descriptor for Detection and Classification[M]// Computer Vision – ECCV 2006. Springer Berlin Heidelberg, 2006:589-600.

[16] Zeng X, Ouyang W, Wang X. Multi-stage Contextual Deep Learning for Pedestrian Detection[C]// IEEE International Conference on Computer Vision. IEEE, 2014:121-128.

[17] Zheng S, Yuille A, Tu Z. Detecting object boundaries using low-, mid-, and high-level information[C]// Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on. IEEE, 2007:1-8.

[18] Huang, YiChin, Tung, et al. An adaptive edge detection based colorization algorithm and its applications[J]. Acm Multimedia, 2005:351-354.

[19] Zhang X, Zou J, He K, et al. Accelerating Very Deep Convolutional Networks for Classification and Detection[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 38(10):1943-1955.

[20] Dong C, Chen C L, He K, et al. Learning a Deep Convolutional Network for Image Super-Resolution[M]// Computer Vision – ECCV 2014. Springer International Publishing, 2014:184-199.

[21] Sýkora D, Dingliana J, Collins S. LazyBrush: Flexible Painting Tool for Hand - drawn Cartoons[C]// Computer Graphics Forum. Blackwell Publishing Ltd, 2009:599–608.

[22] Deshpande A, Rock J, Forsyth D. Learning Large-Scale Automatic Image Colorization[C]// IEEE International Conference on Computer Vision. IEEE Computer Society, 2015:567-575.

[23] Iizuka S, Simo-Serra E, Ishikawa H. Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification[M]. ACM, 2016.