B&W IMAGE COLORIZATION USING DEEP LEARNING

Kaladevi Ramkar
Saveetha Engineering College,
Chennai,INDIA
kalaramar@yahoo.com

Vijaya raghavan.V
Saveetha Engineering College,
Chennai, INDIA
vj123chethanavenky@gmail.com

Rupa Sree S
Saveetha Engineering College,
Chennai,INDIA
sangapurupasree@gmail.com

Abstract: Computerized colorization of black and white images began in the 1970's, since then image colorization has come a long way. RGB images are widely used in many fields to get additional information about the image which cannot be obtained from B&W images. Some areas where RGB images might be used are Determining Chemical composition of a material, Chromatographic Spectroscopy and studying the biology of plants. In this project we approach the image colorization problem using Deep Learning techniques. We achieve this using Conditional Generative Adversarial Networks (cGAN) which will be trained on the COCO dataset. This is a fully automated model and requires no human help to achieve artifact-free quality images. The recent achievements in Generative models are the inspiration behind this project. We present two models in this project out of which the second model produces better results with only the one-fourth of the training time of the first model.

Keywords- Deep Learning, GAN, ML, B&W, cGAN

1. INTRODUCTION

The B&W Image colorization using deep learning is an attempt to colorize black and white images automatically. RGB images are represented by a 3-dimensional array containing the height, width and color information of the image. The color array in the RGB color space contains 3 values for each pixel indicating how much Red, Green and Blue that pixel is. Whereas in LAB Color Space [1], we have 3 values for every pixel but the values aren't like that of the RGB color space. The first L channel which stands for Lightness channel, encodes the Lightness of each which when visualized gives us a black and white image. The 'a' and 'b' channels encode how much red-green and blue-yellow the pixel is, respectively.

The reason why we chose LAB color space to train our model is that when using LAB color space, we can give our model the grayscale image obtained from separating the L channel and make the model predict the a and b channel and finally concatenate all the channels to obtain the colorized image. But if you use RGB color space, you have to convert your image into a grayscale image and then feed it to the model and hoping that it will predict three values for us which is a time and resource wasting task. Considering that we have an 8-bit integer image which means we will have 256 choices to make for each channel, in a RGB color space we will have to make more than 16 million choices, but using LAB color spaces reduces the number choices to about 65000.

Deep learning technology demands high resources. It requires high-performed and more powerful GPU's, large amounts of space to store the data that is used to teach the models, so on. Unlike the traditional machine learning, this technology takes more time to be trained. Though deep learning has all the above-mentioned challenges, it is still being used because it has been discovering new improved methods of unstructured big data analytics day-by-day. Many organizations and businesses gain significant benefits through deep learning. Implementation of deep learning are, it automatically adds sound to silent movies or videos, if can perform automatic machine translation, it can classify objects and detects photographs, it can generate handwriting and text automatically, it can also generate captions for images, it can create chatbots and can also recognise pictures of the similar persons.

In the past colorization of black and white images required a lot of human interaction and hardcoding or providing a reference image [2,3,4] but with the power of AI and deep learning this whole process can be done end-to-end. Even with deep learning the results weren't fascinating enough and often the results were of low quality and full of artifacts. In addition, it also required a large amount of data [5,6] and hours and hours of training to achieve a fairly decent model.

This project provides the functionality of converting a black and white image into a color image with no human interaction other than giving input. This model produces artifact free quality color image of the input B&W image. The output can fool most humans but the predicted colors are not completely accurate. This project is to change those by using only 10000 images and a fairly short training time.

2. RELATED WORK

Bo Li, Fuchen Zhuo, Zhuo Su Xiangguo Liang, Yu-Kun Lai, Rosin PL. [7] proposed a Based Image Colorization Using Locality Consistent Space Representation. This model colorizes a black and white image given a reference color image by sparse pursuit. The drawback to this model is that it requires additional input and the colors we obtain are probably not accurate since it is based on a reference image which is different from the original image.

Pierre, Fabien et al. [8] proposed a Unified Model for Image Colorization. This method provides two ways to colorize an image both requiring some manual input from the user. In the first method the user needs to scribble on certain parts of the images with certain colors to obtain the result and the second way requires a reference color image which will be used to color the black and white image. The colorized image will only contain colors from the reference image.

Richard Zhang, Phillip Isola, Alexei A. Efros [9] proposed a Colorful Image Colorization. This was a novel approach back in 2016 which used classification with class-rebalancing at training to predict the colors but the it required a large amount of data the results were not up to mark.

Iizuka, Satoshi, Edgar Simo-Serra, and Hiroshi Ishikawa.[10] proposed a Let there be color! Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. This was another deep learning approach to this problem but instead of using a classification task they approached it as a regression task to predict the colors. This also required a large amount of training and data and the results were not up to mark

3. PROPOSED SYSTEM

Our proposed system takes in a grayscale image and outputs a colorized version of the input image the grayscale image acts as the condition and is seen by both the generator and discriminator. The generator generates color image which is the final output, the same image is given to the discriminator for comparison with the real image. We use a special U-net architecture [11] where when we reach the middle part of the net by down sampling we then up-sample the modules to the right of that middle module at every iteration until it reaches the output module thus retaining the image size.

3.1. Generator(U-Net)

The GANs [12] don't have control over the types of images generated. The generator simply starts with random noise and continuously generates images that hopefully converge towards representing the training images over time.

A U-Net GAN uses a segmentation network as the discriminator. This segmentation network predicts two classes: real and fake. The generator model is responsible for generating new plausible examples that ideally are indistinguishable from real examples in the dataset.

The generator is provided with a sample vector of random noise. By using de-convolutional layers, an image is generated. We use De-Convolutional layers because unlike Convolutional layers which extract features from an image as an output, A de-convolutional layer tries to create an image as the output given a set of features. De-Convolutional layers perform the reverse operation of the convolutional layers.

GANs are effective at image synthesis, that is, generating new examples of images for a target dataset. Some datasets have additional information, such as a class label, and it is desirable to make use of this information.

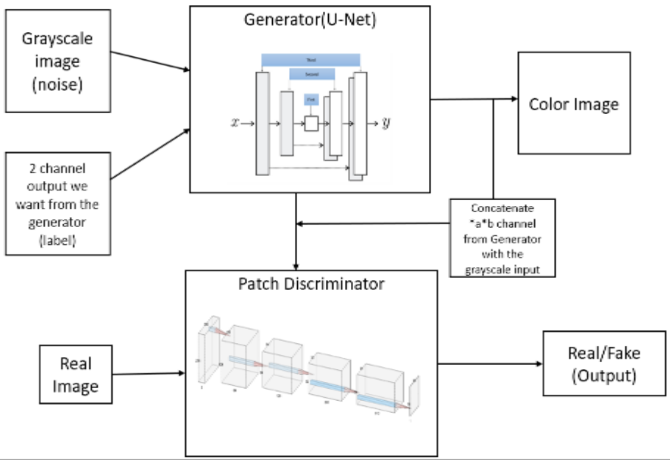
In cGANs [13], a conditional setting is applied, meaning that both the generator and discriminator are conditioned on some sort of auxiliary information (such as class labels or data) from other modalities. As a result, the ideal model can learn multi-modal mapping from inputs to outputs by being fed with different contextual information.

3.2. Discriminator

The discriminator is used to compare the generated super resolution image to the original high-resolution image. The discriminator model is responsible for classifying a given image as either real (drawn from the dataset) or fake (generated). The models are trained together in a zero-sum or adversarial manner, such that improvements in the discriminator come at the cost of a reduced capability of the generator, and vice versa. The generator feeds into the discriminator net, and the discriminator produces the output we're trying to affect. The generator loss penalizes the generator for producing a sample that the discriminator network classifies as fake.

Here the discriminator uses convolutional layers to predict whether the images generated by the generator is real or fake. The goal of the generator thus becomes to produce images that can fool the discriminator and in our case the generator's goal is to produce realistically coloured images that can hopefully fool the discriminator. The discriminator's job is to be critical of the success of the generator. To help the generator produce more realistic images, the discriminator has to be really good at differentiating real and fake images. The better the discriminator is, the better the generator will be. After each iteration, based on the suggestion from the discriminator the generator tunes its learnable parameters like the weights and biases to fit the suggestion and thus improve the quality of the images produced.

The learnable parameters are updated using backpropagation of the discriminator's output gradients in regards to the produced image. Essentially, the discriminator tells the generator how it should tweak each pixel so that the image can be more realistic. The generator in turn tries to fool and the discriminator thus reducing the loss.



3.3. Base cGAN:

We use a conditional GAN, a cGAN is a type of GAN that involves the conditional generation of images by a generator model. Image generation can be conditional on a class label, if available, allowing the targeted generation of images of a given type. In our project the condition is the input grayscale image which is seen by both the generator and the discriminator and is expected to be taken as a condition. in a GAN the generator and discriminator work with each other to solve a problem. In our setting, the generator model is provided with a 1-channel grayscale images and generated a 2-channel image for channels a and b. The discriminator then takes these two produced channels and concatenates them together with the input grayscale image and predicts whether the new colorized image is real or fake. The discriminator will be trained beforehand on some real images in LAB colour space that are not produced by the generator to learn to distinguish between real and fake images.

Loss for cGAN:

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z))]$$

Fig 3.3.1: Adversarial loss

Here x is the grayscale image, z is the input noise for the generator, and y is the 2-channel output we want from the generator, G is the generator model and D is the discriminator. This loss function ensures that our model produces colourful images that seem real.

L1 Loss:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1]$$

Fig3.3.2: L1 loss

This loss function ensures that our model produces colorful images that seem real.
This L1 loss compares the predicted colors with the actual colors.
Using L1 loss alone still colorizes the images but the predicted colors are not saturated and are usually gray or brown.

Combined loss function:

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda\mathcal{L}_{L1}(G)$$

Fig .3.3.3. Combined Loss

3.4. cGAN with ResNet backbone:

This is an improved model based on the previous version where we use the same architecture but implemented it with Resnet18 as the backbone of the U-Net.
To tackle the problem of "The blind leading the blind" in GAN where neither the generator nor the discriminator knows anything about the task at the beginning of training, we pretrain [14,15] the generator in a supervised and deterministic manner [16].
Pretraining occurs at two stages, First The backbone of the generator in the down sampling path is a pretrained model on classification using the ImageNet dataset, secondly the whole generator is pretrained on the task of colorization with L1 loss. We use a pretrained ResNet18 in the U-Net and we train the U-net with only L1 loss and obtain the pretrained model. Then we proceed by combining the adversarial loss [17]and L1 loss, as we did in the previous version. To further simplify things, we built the U-Net using fastai's Dynamic U-Net module.

4. Generative Results:

Results from the base cGAN:

First row: grayscale image(input), *Second row*: Colorized image(output), *Third row*: Original image

Our first model has some basic understanding of some of the most common objects in images like land, sky, plants etc. But the output is far from something appealing and it cannot decide on the colour of rare objects. It also displays some colour spill overs and circle-shaped mass of colour which is not good at all. This why we built a better version of the model since this approach was efficient. The training time is about 12 hours for 50 epochs in Google Colab environment running on a GPU.

Here are some more results from cGAN:

You can see the colour spill overs in the 1st picture of the second row and the desaturated colours in the 4th picture of the second row.

Results from cGAN with ResNet backbone*:*

For our second model we use the same data but here we use a pretraining in two parts, first is the resnet18 from torchvision which is a pretrained model for classification and the next is the whole generator which is pretrained on the task of colorization with only the L1 loss. The colour spill overs and desaturation which occur output of the previous model is not present in the output of this model, you can clearly see the difference in the results there are no colour spill overs, no artifacts and colours are more saturated and realistic. Since our model was trained on “artificial” grayscale images by removing the ab channels from colour images, here we show some of the examples with legacy black and white photos and our model was still able to produce good colorizations. This model also takes only 1/4th of the training time required

Here are some more results cGAN with ResNet backbone:

5. CONCLUSION AND FUTURE ENHANCEMENT

From the results we can see that using deep learning techniques we can create artifact-free quality colour images from black and white images. We have presented a model that can colorize black and white images using cGAN with a U-Net architecture. Based on the results collected personally we found that about 80% of the people surveyed thought that the colorized version of legacy black and white images were real and additionally 70% of the people thought the generated images were real instead of the ground truth image. This system could also be a powerful pretext task for self-supervised feature learning [18,19,20], acting as a cross-channel encoder.

The proposed system is currently limited to only images but there are black and white videos which are available which can be colorized. Therefore, the next step of the proposed system is to implement a model that will be able to colorize videos and possibly do it in real time.

6. REFERENCES

[1] Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to greyscale images. ACM Transactions on Graphics (TOG) 21(3) (2002)

[2] Gupta, R.K., Chia, A.Y.S., Rajan, D., Ng, E.S., Zhiyong, H.: Image colorization using similar images. In: Proceedings of the 20th ACM international conference on Multimedia, ACM (2012)

[3] Charpiat, G., Hofmann, M., Sch�olkopf, B.: Automatic image colorization via multimodal predictions. In: Computer Vision {ECCV 2008. Springer (2008)

[4] Liu, X., Wan, L., Qu, Y., Wong, T.T., Lin, S., Leung, C.S., Heng, P.A.: Intrinsic colorization. In: ACM Transactions on Graphics (TOG). Volume 27., ACM (2008)

[5] Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: Proceedings of the IEEE
International Conference on Computer Vision. (2015) 415-423

[6] Dahl, R.: Automatic colorization. In: http://tinyclouds.org/colorize/. (2016)

[7] Bo Li, Fuchen Zhao, Zhuo Su, Xiangguo Liang, Yu-Kun Lai and Paul L. Rosin, "Example-based Image Colorization using Locality Consistent Sparse Representation", *Journal of latex class files*, vol. 13, no. 9, September 2014.

[8] Pierre F., Aujol JF., Bugeau A., Ta VT. (2015) A Unified Model for Image Colorization. In: Agapito L., Bronstein M., Rother C. (eds) Computer Vision - ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science, vol 8927. Springer, Cham. https://doi.org/10.1007/978-3-319-16199-0_21

[9] Richard Zhang, Phillip Isola, Alexei A. Efros "Colorful Image Colorization", arXiv pre-print arXiv:1603.08511 [cs.CV], October 2016.

[10] Iizuka, Satoshi, Edgar Simo-Serra, and 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 (ToG) 35.4 (2016): 1-11.

[11] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.

[12] Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems* 27 (2014).

[13] Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." *arXiv preprint arXiv:1411.1784* (2014).

[14] Tan, Chuanqi, et al. "A survey on deep transfer learning." *International conference on artificial neural networks*. Springer, Cham, 2018.

[15] Frégier, Yaël, and Jean-Baptiste Gouray. "Mind2Mind: transfer learning for GANs." *arXiv preprint arXiv:1906.11613* (2019).

[16] Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros "Image-to-Image Translation with Conditional Adversarial Networks" arXiv pre-print arXiv:1611.07004 [cs.CV], November 2016.

[18] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[19] Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence 35(8)(2013)

[20] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep
learning. In: Proceedings of the 28th international conference on machine learning
(ICML-11). (2011)