B&W IMAGE COLORIZATION USING DEEP LEARNING

Kaladevi Ramkar
Anna University of Technology, Chennai
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 rank-3 (height, width and color) array with the last axis containing the color data for our image. These data represent color in RGB color space and there are 3 numbers for each pixel indicating how much Red, Green, and Blue the pixel is. In L*a*b color space[1], we again have three values for each pixel but the meaning of these values is different. The first channel L, encodes the Lightness of each pixel and when we visualize this it appears as a black and white image. The *a and *b channels encode how much green-red and yellow-blue each 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 the L channel to the model (which is the grayscale image) and want it to predict the other channels (*a, *b) and after its prediction, we concatenate all the channels and we get our colorful image. But if you use RGB, you have to convert your image to grayscale, feed the grayscale image to the model and hope it will predict 3 numbers for you which is a way more difficult and unstable task due to the many more possible combinations of 3 numbers compared to two numbers. If we assume we have 256 (we say 256 specifically because that's the number of choices for an 8-bit integer image) choices for each number, predicting the three numbers for each of the pixels is choosing between 256^3 combinations which is more than 16 million choices, but when predicting two numbers we have about 65000 choices.

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 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 input 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

In our proposed system the condition is the input grayscale image which both the generator and discriminator see and expect that they take this condition into consideration. 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 the we reach middle part of the net by down sampling and up-sample the modules to the right of that middle module at every iteration until it reaches the output module.

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 repeatedly creates images that hopefully tend 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 takes a latest sample, a vector of random noise in as input. By leveraging de-convolutional layers, which is essentially the reverse of convolutional layers, an image is produced. Convolutional layers are responsible for extracting features from an input, de-convolutional layers perform the reverse as it takes the features in as input, and produces an image as the output. Let's say you're playing charades with friends. The person who's acting the phrase is performing de-convolutions, while the other players guessing the phrase is performing convolutions. The acting is analogous to the image, while the guessing is analogous to the features of the image. In essence, the convolutional layers identify the features of the image, and the de-convolutional layers construct an image given it's features.

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.

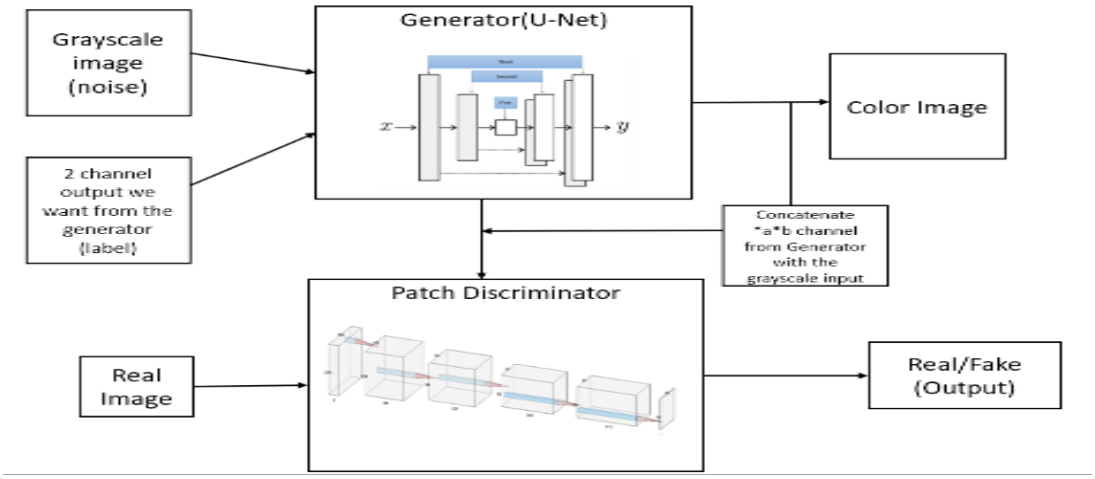
In cGANs [13], a conditional setting is applied, meaning that both the generator and discriminator are conditioned on some sort of auxiliary information (such as class labels or data) from other modalities. As a result, the ideal model can learn multi-modal mapping from inputs to outputs by being fed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

The discriminator , leverages convolutional layers for image classification as its job is to predict whether the image produced from the generator is real or fake. The objective of the generator is to produce images that are as realistic as possible and successfully fool the discriminator into thinking that the generator images are real. The discriminator's job is critical to 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 every iteration, the generator's learnable parameters the weights and biases are refined according to the suggestion given by the discriminator.

The networks updates the learnable parameters by backpropagating through the gradients of the discriminator's output in regards to the generated image . Essentially, the discriminator tells the generator how it should tweak each pixel so that the image can be more realistic. The generator , however, replies on the discriminator to be successful.



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 both the generator and discriminator see and expect that they take this condition into consideration. in a GAN we have a generator and a discriminator model which learn to solve a problem together. In our setting, the generator model takes a grayscale image (1-channel image) and produces a 2-channel image, a channel for *a and another for *b. The discriminator, takes these two produced channels and concatenates them with the input grayscale image and decides whether this new 3-channel image is fake or real. The discriminator will also be trained on some real images in LAB color 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 colorful 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 brow.

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

The baseline mode has some basic understanding of some most common objects in images like sky, trees etc. Its output is far from something appealing and it cannot decide on the color of rare objects. It also displays some color spill overs and circle-shaped mass of color 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.

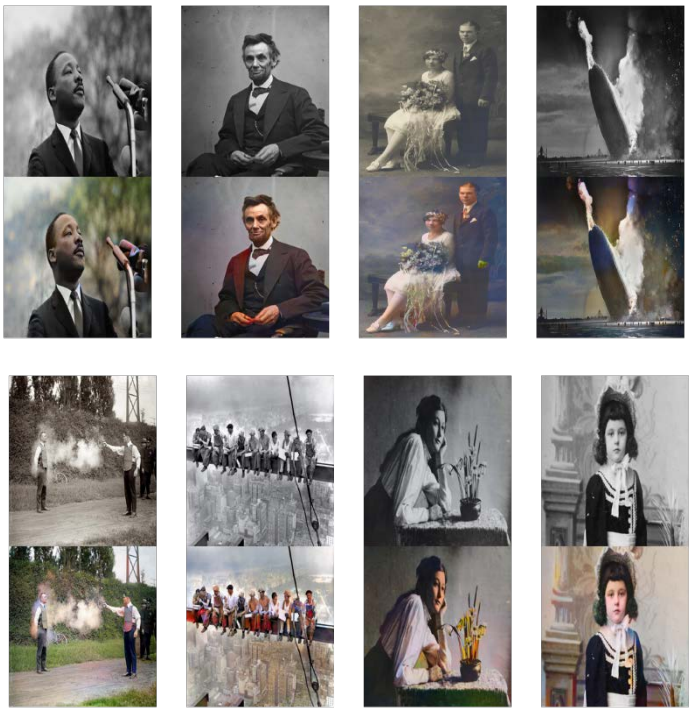
Here are some more results from cGAN:

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

Results from cGAN with ResNet backbone*:*

For the 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 color spillovers 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 color spill overs, no artifacts and colors are more saturated and realistic. Since our model was trained on “artificial” grayscale images by removing the ab channels from color 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

We have presented a method using deep learning techniques to create artifact-free quality color 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% thought that the colorized version of legacy black and white images were real and additionally 70% of the people thought the generated images was 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)