# Old Movie Colorization Using Deep Learning Techniques

***A B.Tech Project Report Submitted***
***in Partial Fulfillment of the Requirements***
***for the Degree of***

**Bachelor of Technology**

*by*

**Mukul Ranjan**(170102084)
**Chandan Govind Agrawal**(170102077)

*under the guidance of*

**Prof. P. K. Bora and Dr. A. Rajesh**


**to the**

**DEPARTMENT OF ELECTRONICS AND ELECTRICAL ENGINEERING**
**INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI**
**GUWAHATI - 781039, ASSAM**

# CERTIFICATE

*This is to certify that the work contained in this thesis entitled "**Old Movie Colorization Using Deep Learning Techniques**" is a bonafide work of **Mukul Ranjan (Roll No. 170102084)** and **Chandan Govind Agrawal (Roll No. 170102077)**, carried out in the Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati under my supervision and that it has not been submitted elsewhere for a degree.*

Supervisors: **Dr. P. K. Bora**

**Dr. A. rajesh**

Professor/Associate Professor,

November, 2023 Electronics & Electrical Engineering,

Guwahati. Indian Institute of Technology Guwahati, Assam.

# Abstract

*Colors can express emotions of the scene even without a word being spoken. This work attempts to use different deep learning techniques for automatic colorization of old black and white pictures/movies. Colorization is a one to many task in which a single channel image is converted into a 3 channel image hence it has many possible solutions. But if a model is able to understand a scene well and given proper domain knowledge there remains very few plausible solutions. We use Generative Adversarial Networks (GANs) to generate realistic color images in which two networks trying to overpower each other end up making each other better at their respective jobs. To express a scene, multiple parts of an image have to work in harmony with multiple far away parts. We tried to translate this non local harmony into our model using a Self-Attention module along with CNNs. This module solves the problem of color bleeding in many instances. We also have created our own dataset whose domain is close to the most of the old black and white movies. Some of the colorized video clips can be found here.*

# Contents

# List of Figures

# Chapter 1

# Introduction

People of the younger generation are always curious to know how the things in the past were. At the same time the older generation wants to relive the memories of their childhood and youth. Old Bollywood movies are the best sources which can cater to both of this generation's fancy. Movies tell us a lot about the culture and people of the times they are released in. When we watch an old Bollywood black and white movie, we always crave for a more detailed and clear picture so that we all can take a trip into the past and live the magic of those black and white classics. Colorization has definitely added more realism to the movies we watch. So here in this project we make an effort to colorize those old black and white movies.

In India, K. Asif's magnum opus romance Mughal-E-Azam was the first colorized re-release in 2004. Huge manual effort, time and money was put into it to make it happen. But now with the rise of deep learning, automatic colorization is possible with very little human intervention. Convolutional neural networks, Generative modelling and Attention mechanisms are with time being used for improving the quality of colorization in deep learning models.

Before the deep learning era, image colorization was broadly performed using the following two approaches in computer graphics: 1) User-guided edit propagation - In this method intensive user interaction is required. A user has to draw colored strokes on the grayscale image and based on user inputs and hand defined image priors, the algorithm generates a colorized image. Each differently colored region even with a little uncertainty must be explicitly indicated by the user. In this method, even if the user knows what general color a region should have, it can be tough to select chrominance appropriately for the region. 2) Data driven automatic colorization – This method is slightly better than the first one. For a given grayscale image, it searches for a similar reference color image from the database, and transfers or "steals" colors from the photo to the grayscale image. But this method is good as long as we can find a similar reference image in the database.

The most recent methods in this paradigm are all deep learning methods. [24], [12] and [9] propose a fully automatic method using deep neural networks. They made colorization easy and cheap but results often contain incorrect colors and they are obvious to be identified as a fake image. [10] used conditional GANs for colorization improved the results. [25] combines user guided colorization along with a deep learning model and leverages both human knowledge and data driven approach to generate a lot of variations for a single image.

The main challenge in colorization is the uncertainty associated with the color of the object/region. A t-shirt, car can be of several colors such as red, blue, green. Color shade of different objects kept in open may change depending upon what part of the day it is. Hence there are a lot of possibilities when it comes to what is plausible color scheme of the image. Previous works have observed that a model fails to colorize well on the image which is different from the training dataset. It is called to be having 'dataset bias'. Larsson et al. [12] have observed that semantic information is very important to choose right colors for

a region. A system should be able to interpret semantic composition (detect objects) from the image and localize objects in order to colorize well. In this project, we aim to tackle all such problems and build a robust colorization model.

**Organization of The Report**

This chapter provides basic introduction and importance of our work. The next chapter contains various works related to our task and we will briefly introduce all of them. In the third chapter we will introduce the various methods which we are using for this work, the chapter contains the detailed explanation of various Generative networks which we have used for this work. The fourth chapter gives the detailed explanation about various experiments we did and also the results of our work and its comparison with other related models. In the last and the fifth chapter we explained about various challenges and ideas which we proposed to solve in future.

# Chapter 2

# Literature Review

Image colorization being one of the most pertinent and challenging problem of computer vision, a lot of work have been done both by using traditional and deep learning based methods. In this chapter we have discussed various methods available in the colorization literature.

Traditional based methods are generally high human intensive methods such as having needed to draw color strokes on black and white images ([14], [8]). These color strokes are incorporated into a colored image by similarity metrics. Over the time as these similarity metrics improved, the amount of user effort reduced. Later with the surge of deep learning, neural networks are also used to learn similarity between pixels and user recommended color strokes [2]. Another method is more data dependent one. For a given grayscale image, one or more reference color images are found from a database. Then color from relevant regions of these images is transferred to the corresponding region of grayscale image following Image Analogy [7]. Manga Colorization [17] techniques uses color propagation over the region containing pattern-continuity and intensity-continuity. Yatziv *et al.* developed a method which provides user to interactively get the desired result by using a reduced chrominance scribbles. Natural Image Colorization[16] also provides an interactive system for users that uses two separate stage of color labelling and color mapping.

The advent of deep learning[13] emerged as a boon for various computer vision tasks including image classification ([11], [6],[20]), object detection([15], [18], [3]) and object segmentation([5], [19]). Beginning from the Deep Colorization[1] several successful attempts([24], [12], [9]) have been made for the image colorization using deep learning based methods. All three are similar approaches which involve CNN and large scale data. The loss functions and network architectures are different. Zhang *et al.*[24] and Larsson*et al.* [12] uses classification loss while [9] uses regression loss. Zhang *et al.* [24] and Larsson *et al.*[12] train models on ImageNet while Iizuka *et al.*[9] trains on Places dataset.

Zhang *et al.*[25] tries to combine both user-guided colorization and deep learning based automatic colorization. In this method a user can suggest a color for a particular region in grayscale image and the deep learning model then generates a color theme based on the suggestion and colorizes the image. Lot of different samples with different color themes for the same image can be generated with this method. Isola *et al.*[10] introduces conditional GAN (also called Pix2Pix) for image to image translation. The generator of the model is given an input grayscale image for which it generates a color image which the discriminator tries to distinguish as fake or not conditioned on the given input grayscale image. Image generation with the GAN [4] is found to be promising by the deep learning community in a kind of task which has ample data available. Hence it seems logical to proceed with GANs for this task.

Deep learning architectures are continuously being improved and the self attention module[21] is the most recent such a development in this line. It is found really useful in modelling long range dependencies. [23] incorporates this self attention module into GAN and shows its effectiveness in image generation. As this is the most improved form of GAN available, we will proceed with SAGAN to build our model. DeOldify (2019, non published work) also follows SAGAN to build its colorization model. But DeOldify is mainly con-

strained to western lifestyle settings and as the lifestyle, culture etc. changes the domain of images greatly, we will build a separate colorization model with the focus on Indian settings.

# Chapter 3

# Method

As we discussed earlier the problem of Image colorization problems can be modelled as a **self supervised learning** problem where a deep Convolutional Neural Network is trained end-to-end in **Lab** color domain of an image where a RGB image is first converted to LAB color space whose L channel is fed to the network which produces ab channel. Hence any color photo can be used as a training example, simply by taking the image's L channel as input and its ab channels as the output. The work of [24] can be seen in the image 3.1.

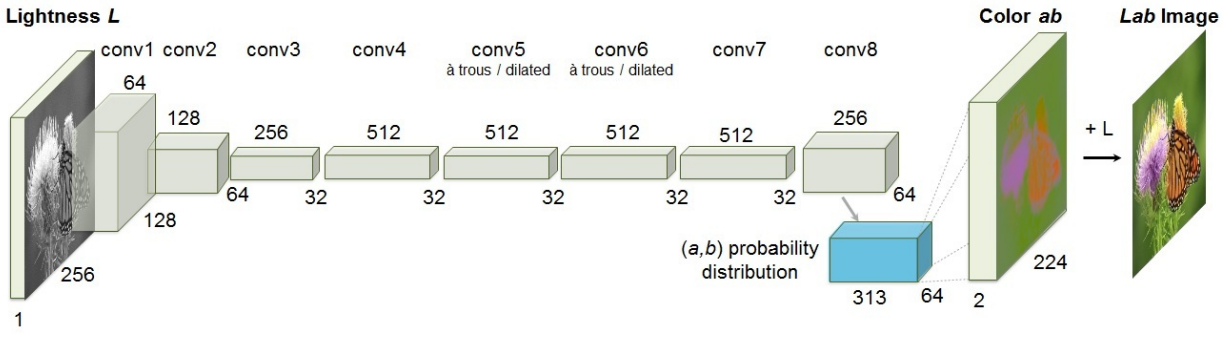


**Fig. 3.1**: Network Architecture of Zhang et al.(2016)

Although the method proposed in [24] opens up a new domain for the task of image colorization which is much cheaper and much more efficient than traditional methods it comes with a its own problems. The model seems to confuse a lot even in case of some of the obvious object and it can not be considered to be reliable. Since our task of colorizing

old Indian movies is much more difficult than the simple image colorization task because of domain difference and inherently different context. We choose to use Conditional Generative Adversarial Network(conditional GAN) based methods for our problem. In order to understand the methods used by us let us first look into some of the introductory concepts which we used throughout our projects.

## 3.1 Generative Adversarial Network(GAN)

**Generative models** are a group of unsupervised machine learning models which learn the distribution of the input data in such a way that the model can be used to generate new examples that could have been drawn from the original data distribution. Consider an example of generating a cat image with a size of $n \times n$ pixels. This cat image can be represented as a $N = n^2$ dimensional vector by flattening the image matrix. This flattened $n^2$-dimensional vector represents a specific probability which when converted to a square matrix of size $n \times n$ gives the cat image. Hence over this N-dimensional vector space there exists specific probability distributions for the images of human, tables and so on.Therefore the problem of generating cat's image can be modelled as problem of generating a new vector from N-dimensional latent space with specific **cat(any specific) probability distribution**. Can we solve this problem by transforming a simple N-dimensional uncorrelated random variable and applying it to a very complex function(read Neural Network)?.

The answer to the above function is Generative Adversarial Networks[4], which are a class of neural network based unsupervised generative models which generates data distribution using two separate neural networks namely **Generator** and **Discriminator**. Root of GAN originates from the game theoretic approach. We can consider GAN as a two player game where the two component **Generator** and **Discriminator** of the model are adversaries and always are in battle throughout the training process.

The Generator in the GAN create a fake data to the given distribution while Discriminator detects whether the generated data fake or not. To a real life analogy consider

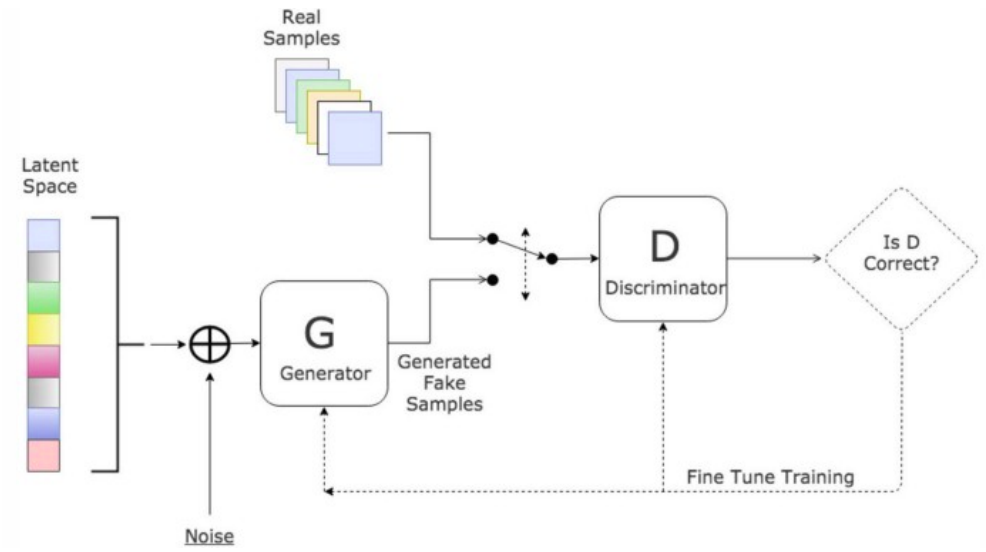


**Fig. 3.2**: Understanding Generative Adversarial Networks

generator as a human art forger who create a fake world of art and discriminator as an art expert. Throughout the training process Discriminator can be considered as an adversary of the Generator. Generator tries to create the data in such a way that Discriminator isn't able to distinguish it fake anymore. This two player game continue until Generator succeeds in creating new data which discriminator can not identify as fake.

A neural network $G(z, \theta_1)$ is used to model Generator which maps input latent variable $z$ to the desired data space x(images) while another neural network $D(x, \theta_2)$ models the Discriminator and output the probability that data comes from real datasets. $\theta_i$ in both the cases represents weight of the neural network model.

Thus Discriminator is trained to correctly classify input data as real and fake. Thus the loss function used for Discriminator maximizes the function $D(x)$ at the same time also minimizes the function $D(G(z)$. On the other hand Generator is trained to trained to fool the Discriminator by generating data as real as possible, which means that the loss function for generator maximizes $D(G(z))$.

Since both Discriminator and Generator is trying to optimize opposite loss function, the problem can be modelled as minimax game with the value function of $V(G, D)$. In this game generator is trying to maximize the probability of its output being identified as real

while Discriminator is trying to minimize the same value. Hence the loss function can be written as follows.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{3.1}$$

## 3.2 Conditional GANs (Pix2Pix)

Pix2Pix[10] is a Generative adversarial network designed for the purpose of image to image translation. As we have seen in the previous section that GAN learns a mapping from random noise vector $z$ to output image $y$, $G : z \to y$. In contrast, conditional GANs learn a mapping from observed image $x$ and random noise vector $z$, to $y$, $G : \{x, z\} \to y$. The training process of the pix2pix can be understood from the figure 3.3.We observe that the conditioning image, $x$ is applied as the input to the generator and also to the discriminator.

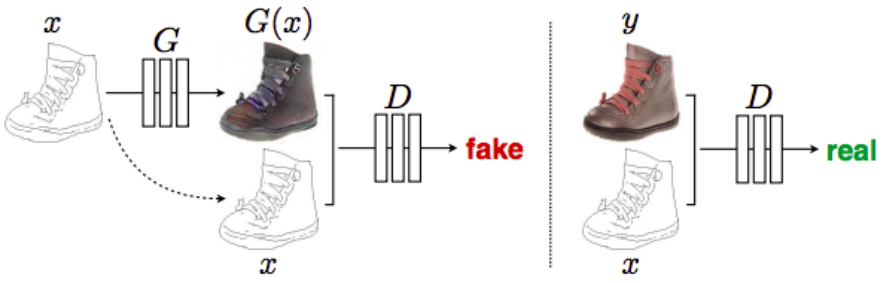


**Fig. 3.3**: A high-level view of the Image-to-Image translation architecture as depicted in [10]

The simplest way for image-to-image translation is to directly pass the input image from a neural network and optimize the $\mathcal{L}_1$ or $\mathcal{L}_2$ loss function.But directly designing such model will fail to capture many important characteristics between these images. The objective of a conditional GAN can be expressed as

$$\begin{aligned}\mathcal{L}_{cGAN}(G,D) =& \mathbb{E}_{x,y}[\log D(x,y)] + \\ & \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z))],\end{aligned} \tag{3.2}$$

where $G$ tries to minimize this objective against an adversarial $D$ that tries to maximize it, i.e. $G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D)$.

Authors found that adding addition $\mathcal{L}_1$ loss function given by equation 3.3 improves the performance of the model and reduces blurring.

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x, z)\|_1]. \tag{3.3}$$

Hence the final objective function of the conditional gan be given by

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G). \tag{3.4}$$

## 3.3 Self-Attention GAN

To understand this, we should first break this term into "Attention", "Self-Attention" and "Self-Attention GAN" and understand each of them separately.

### 3.3.1 Attention

In cognitive science, selective attention means how we restrict our attention to particular object. It helps us focus, so we can tune out our attention from irrelevant information and concentrate on what really matters. We can apply a similar type of attention mechanism in deep learning as well.

For ex. If we are asked a question that "what is the color of t-shirt of the guy who is kicking the ball?". To answer this, important regions which we should look at are the ball and the t-shirt of the guy who is kicking the ball.

Let's say the question is transformed into a feature vector $q = f(question)$, we call it a query. Let's represent image as $x = [x_1, x_2, .., x_N]$. Here each $x_i$ is a feature vector. We want to determine for a given q, what is the weight $\alpha_i$ of feature vector $x_i$, to determine the output. First we transform image x into keys $k_i = g(x_i)$ and values $v_i = h(x_i)$, for

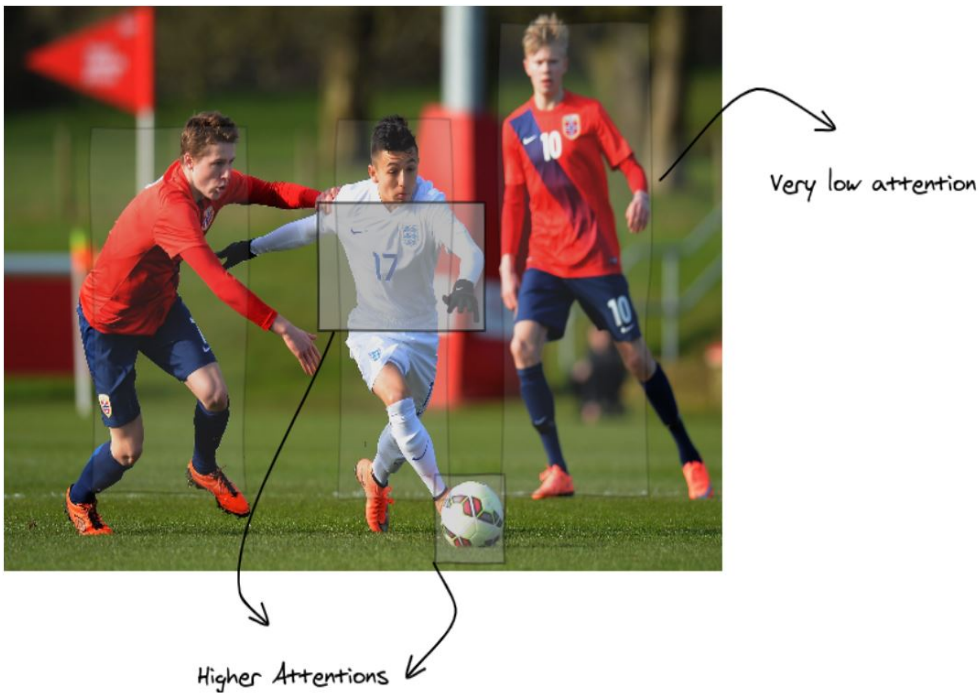


**Fig. 3.4**: Understanding Attention

$i = i = 1, 2, \ldots, N.$

$$\alpha_i = attention(q, k_i)$$
$$\alpha_i = softmax(\alpha_i) \tag{3.5}$$

Where attention(.) has several possibilities one of which is a dot product. Now, to determine output, we transform features vectors $v_i$, with $\alpha_i$.

$$o = output\left(\sum_{i=1}^{i=N} \alpha_i * v_i\right) \tag{3.6}$$

Where output(.) can be any linear/non-linear transforming function.

### 3.3.2 Self-Attention

Let's say we have a bunch of feature vectors to represent an entity such as a sentence or an image. Self-attention can be thought of as making each of the feature vectors in this bunch "aware" of the others. The representation of each input feature vector obtained after a self-attention mechanism applied on the inputs can be termed as the contextual representation of the input.

The difference between attention and self attention is that, in the latter we don't have a separate input for a query. Here one of the feature vectors acts as a query and we determine what is the relevance of other feature vectors for the feature vector in consideration. We do the same for all other feature vectors. Mathematically, Let's represent image as $x = [x_1, x_2, .., x_N]$. Here each $x_i$ is a feature vector. Here $q_i = f(x_i)$ are queries, $k_i = g(x_i)$ are keys and $v_i = h(x_i)$ are values for $i = 1, 2, .., N$. We compute weights $\alpha_{ji}$ as follows:

$$\alpha_{ji} = attention(q_j, k_i) \tag{3.7}$$

We compute output $o_j$ for each $q_j$ as follows,

$$o_j = output\left(\sum_{i=1}^{i=N}(\alpha_{ji} * v_i)\right) \tag{3.8}$$

So the output becomes $O = [o_1, o_2, .., o_N]$.

### 3.3.3 Self-Attention GAN

We incorporate the self attention mechanism described above into both generator and discriminator networks of GAN. The convolutional neural networks are infamous for not being able to model long range dependence. That where self attention comes. It exactly complements convolutions in that aspect.

The images coming from self attention module have information at location very well coordinated with information at other distant locations of the image. Also, the discriminator enforces these global relationships and hence gives a better prediction of fake images. Hence, self attention enables both generator and discriminator to efficiently model relationships between widely separated spatial regions. This method is called Self-Attention Generative Adversarial Network (SAGAN). Here, we will also describe what are f(.), g(.), h(.), attention(.), output(.) functions in our case. Along with it we will describe methods for stable training of GANs.

Here, we will give detailed mathematical formulation of SAGAN. We represent image features from previous hidden layer as x $\in \mathbb{R}^{C\times N}$. Here C is number of channels and N is number of feature locations in the hidden layer. Similar to previous methods, here also we will transform these features into queries, keys and values with $g(x) = W_g x$, $f(x) = W_f x$ and $h(x) = W_h x$ respectively.

Lets compute the amount of attention given to $i^{th}$ location when synthesising output for $j^{th}$ location.

$$\begin{aligned} \alpha_{ji} &= attention(q_j, k_i) \\ \beta_{ji} &= \frac{\exp(\alpha_{ij})}{\sum_{i=1}^{i=N} \exp(\alpha_{ij})} \end{aligned} \tag{3.9}$$

The output of the attention layer is $o = [o_1, o_2, .., o_j, .., o_N] \in \mathbb{R}^{C\times N}$, where,

$$\begin{aligned} o_j &= v\left(\sum_{i=1}^{i=N} \beta_{ji} h(x_i)\right) \\ v(x_i) &= W_v x \end{aligned} \tag{3.10}$$

In the above formulation, $W_g \in \mathbb{R}^{\tilde{C}\times C}$, $W_f \in \mathbb{R}^{\tilde{C}\times C}$, $W_h \in \mathbb{R}^{\tilde{C}\times C}$ and $W_v \in \mathbb{R}^{C\times \tilde{C}}$ are learned weight matrices, which are implemented as $1 \times 1$ convolutions. Here $\tilde{C} = C/8$.

In addition, we further multiply the output of the attention layer by a scale parameter

and add back the input feature map. Therefore, the final output is given by,

$$y_i = \gamma o_i + x_i \tag{3.11}$$

$\gamma$ is a learnable parameter and it is initialized to 0, i.e the network is allowed to trust the convolutions to to learn the required features. $\gamma$ is then slowly changed to give more weightage to self attention mechanism.

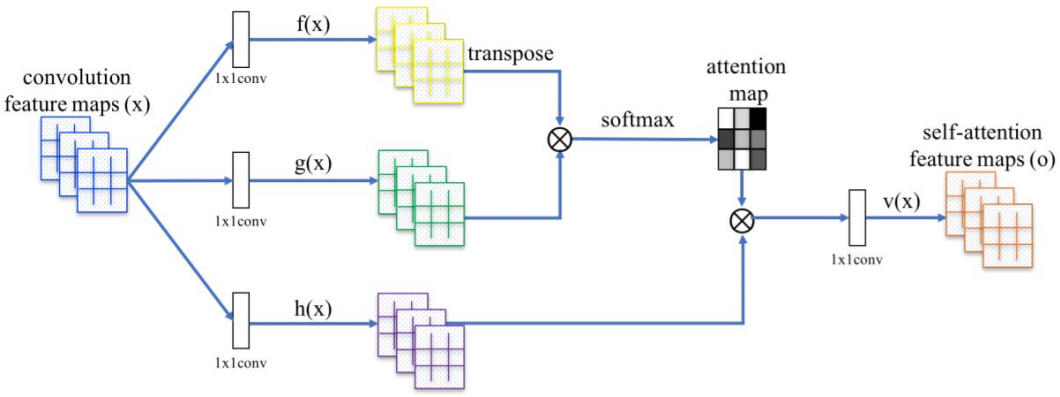


**Fig. 3.5**: The proposed self-attention module for the SAGAN. The $\otimes$ denotes matrix multiplication. The softmax operation is performed on each row.

The loss used to train this model is hinge version of the adversarial loss,

$$\begin{aligned} L_D = &- E_{(x,y)\sim p_{data}}[min(0, -1 + D(x, y))] \\ &- E_{z\sim p_z, y\sim p_{data}}[min(0, -1 - D(G(z), y))] \end{aligned} \tag{3.12}$$

$L_D$ is discriminator loss and $L_G$, the generator loss is given by

$$L_G = -E_{z\sim p_z, y\sim p_{data}} D(G(z), y) \tag{3.13}$$

### 3.3.4 Stable Training of GAN

As training of GAN is like a two player game both trying to undermine each other, the models suffer some major problems like non-convergence, mode collapse, diminished gradi-

ents. The SAGAN paper proposes some ideas to stabilize the GAN training and prevent the problems mentioned above.

1. **Spectral Normalization**: This is a normalization technique which restricts the spectral norm of each feature layer. It is applied in both generator and discriminator and it doesn't require any additional hyper parameter setting. The computational cost is also low. This has been shown to be effective in stable training of GAN. Miyato et. al. 2018

2. **Two Time Scale Update Rule**: GAN training is generally done by updating discriminator multiple times (e.g. 4) against single update of the generator. But Heusel et. al have shown that using two different learning rates results in fewer discriminator updates per generator update. Better results are expected in this same time if this approach is followed.

## 3.4 Conclusion

In this chapter we explained all the important concepts which we are using to build our colorization model. Please note that the GAN, Pix2Pix model and Self-Attention GAN are not independent concepts but are different versions of GAN developed over the time. The model which we aim to build for our colorization task is of Pix2Pix type with self-attention module incorporated at different levels in its network. More details about our model are explained in the chapters to come.

# Chapter 4

# Experiments and Results

## 4.1 Baseline and End-to-End CNN

Here we first present the comparison of the results of some prior works. As mentioned earlier [24], [12] are automatic colorization models while [25] is the user guided image colorization method.

As we can observe, in fig. 4.1 a lot of colored-images deviate a lot from the given ground truth. It is still acceptable if a generated colored image deviates from the ground truth as long as it looks plausible to the human eye when viewed separately. But here, there are some images in which a dominant color (the color which is present in the large part of the image) leaks beyond the object it should be confined in. Also there are some images in which some part of the image remains uncolored (grayscale) probably because the object was not detected well. So our objective is to tackle all such problems with our model.

## 4.2 Pix2Pix

In these recent times, GANs [4] have shown remarkable performance in image generation and distribution modelling. Different versions of GANs have been able to create deep fakes, perform style transfer, generate realistic photographs and create cartoon characters

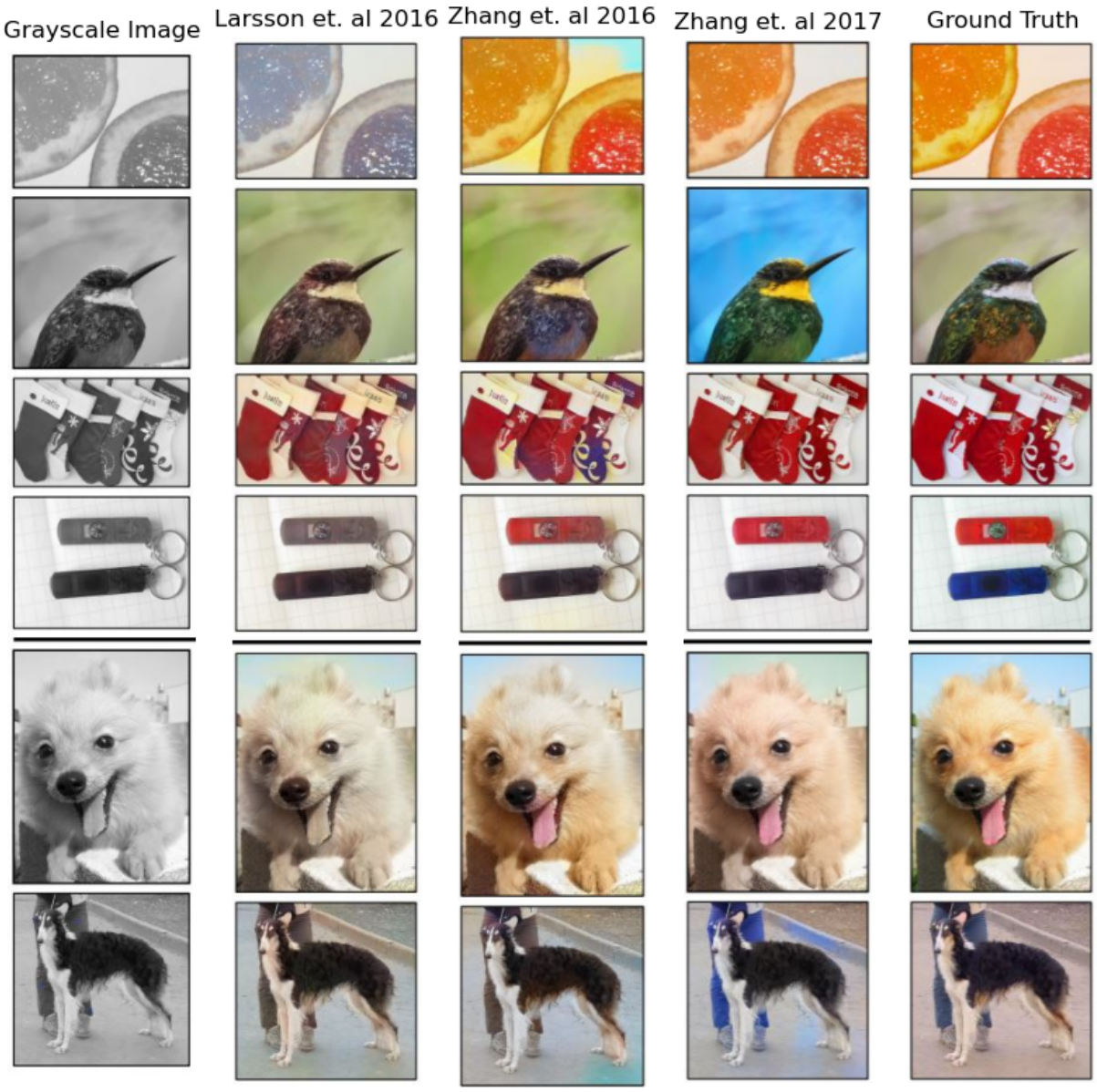


**Fig. 4.1**: These are one of the best performing examples. Note that the results from [25] are based on their interactive colorization scheme, i.e. a human user was required to suggest appropriate colors to get these results from the given grayscale image. All these source images are taken from ImageNet dataset.

convincingly well. Hence we feel that GANs are mighty enough to perform a complex task like image colorization. We start with a basic image to image translation model with GAN i.e. Pix2Pix [10] which is explained above. To interpret a colored image, we use Lab color space. L represents lightness, a and b represents color dimensions. L alone basically is a grayscale version of color image given in Lab color space. Lab is designed to be perceptually linear and (a, b) vector defines a Euclidean space where the distance to the origin determines chroma. The generator of our Pix2Pix model is given a grayscale image i.e. L, a single channel image and it outputs 2 channel ab image of same height and width. We use UNet as our generator for $64 \times 64$ images.

Note that we also use $L_1$ loss with weight parameter gamma along with the adversarial loss. The discriminator is a simple convolutional network which outputs a probability of the (input, output) pair is fake or not. The discriminator and generator loss are given in equation 3.4.

We train our model on two batches of ImageNet64 dataset i.e. on around 200k images. Given below is the comparison of our results with [25].


**Fig. 4.2**: Comparison of our method with [25]. The grayscale image has been obtained from the movie Pather Panchali.

In fig. 4.2 and fig. 4.3, for now our results are not that great compared to [25]. In our case as well the dominant color leaks beyond the intended objects and affects the color of other objects. As you can see lower part of the image is all in reddish color and upper part in greenish. As we have recently built this model, we expect to see improved result as we finetune the model and increase the dataset size. We can also experiment with hyper parameters of loss functions to change the objective of the model slightly. We also plan to


**Fig. 4.3**: Results of our Pix2Pix colorization model on flowers dataset.

include a self attention module in future, which we believe will definitely give some exciting results that we can look forward to.

In the next section we talk about major challenges faced in colorization task and our future plans to tackle them.

## 4.3 Pix2Pix with Self-Attention

In this section, we added Self-Attention (SA) module described in section 4848 to generator and discriminator. SA mechanism helps to process non-local information effectively. We have run our experiments on two different datasets:

- **Pascal VOC 2012:** This dataset contains around 17000 images curated for the task of object detection and segmentation. Images contain a wide variety of scenes and objects. Images are of dimension $128 \times 128$.

- **Movies Dataset:** We have created this dataset especially for the task of old movies colorization. We have selected around 10 color movies in which stories are based on the 1900 to 1950 era. In this way, we have kept the domain of our dataset close to most of the black and white movies. To create a training dataset, we sample 2 frames from each second of the clip uniformly till 150 minutes of the movie. Remainder of

the movie frames go into a test dataset. We resize each frame to $256 \times 512$. The size of this dataset is around 170,000 images.

One disadvantage of using SA module is, it has $O(n^2)$ complexity. For an Image of size $H \times W$, time complexity of an attention module is $O(H \times W \times H \times W)$. Images in Movies dataset are of size $256 \times 512$ and training different experiments on that dataset would take a lot of time. So we performed same experiments with a smaller dataset Pascal VOC with image size $128 \times 128$. We picked the best performing experiment out of these and ran it again with Movies dataset. Following is the list of experiments we performed: In each of the following experiments we used generator as UNet128, discriminator as 4 layer CNN, image size $128 \times 128$, learning rate 0.0002 and were trained for 500 epochs.

| Experiment No | Self-Attention(SA) | Loss Function | SA position |
|---|---|---|---|
| $Expt_1$ | No | Log Loss | _ |
| $Expt_2$ | No | Least Square Loss | _ |
| $Expt_3$ | Yes | Log Loss | On feature Size $32 \times 32$ |
| $Expt_4$ | Yes | Least Square Loss | On feature Size $32 \times 32$ |
| $Expt_5$ | Yes | Log Loss | On feature Size $64 \times 64$ |
| $Expt_6$ | Yes | Least Square Loss | On feature Size $64 \times 64$ |
| $Expt_7$ | Yes | Log Loss | On feature Size $128 \times 128$ |
| $Expt_8$ | Yes | Least Square Loss | On feature Size $128 \times 128$ |

**Table 4.1**: Summary of Different Experiments

We found that experiment 5 gives more visually plausible results and the best SSIM values and hence we ran the experiment 5 with same parameters on Movies dataset. All the results are shown in fig( 4.4 and 4.5) and table(4.2).

## 4.4 Performance Metrics

We have used following two different performance metrics to assess the quality of generated images.

| Experiment No | PSNR | mean SSIM |
|---|---|---|
| $Expt_1$ | $20.668 \pm 2.275$ | $0.8520 \pm 0.0602$ |
| $Expt_2$ | $20.903 \pm 2.512$ | $0.8557 \pm 0.0580$ |
| $Expt_3$ | $16.675 \pm 1.586$ | $0.6794 \pm 0.0644$ |
| $Expt_4$ | $20.419 \pm 2.238$ | $0.8476 \pm 0.0618$ |
| $Expt_5$ | $20.307 \pm 2.082$ | $0.8613 \pm 0.0577$ |
| $Expt_6$ | $20.830 \pm 2.366$ | $0.8385 \pm 0.0585$ |
| $Expt_7$ | $19.447 \pm 2.805$ | $0.7381 \pm 0.1077$ |
| $Expt_8$ | $22.169 \pm 3.96$ | $0.8629 \pm 0.1186$ |

**Table 4.2**: Result across Different Experiments

- **Peak Signal to Noise Ratio (PSNR)**: It is the log ratio between maximum possible value of an image to average mean sqaure error. PSNR generally measures how the generated image differs from the ground truth on a pixel level.

$$\mathcal{M}SE = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|] \tag{4.1}$$

$$\mathcal{P}SNR = 20\log(\frac{\max(\max(y))}{\sqrt{MSE}}) \tag{4.2}$$

  Here y is the ground truth image and G(x, z) is the generated image.

- **Structural Similarity Index Measure (SSIM)**: SSIM [22] is used measure structural similarity between two images. This metric assumes that human visual perception is highly dependent on extracting structural information from image scenes. Hence this metric focuses on measurement of degradation of structural information. SSIM extracts 3 key features from an image: **Luminance**, **Contrast** and **Structure**. This features are then combined to calculate total SSIM of an image.

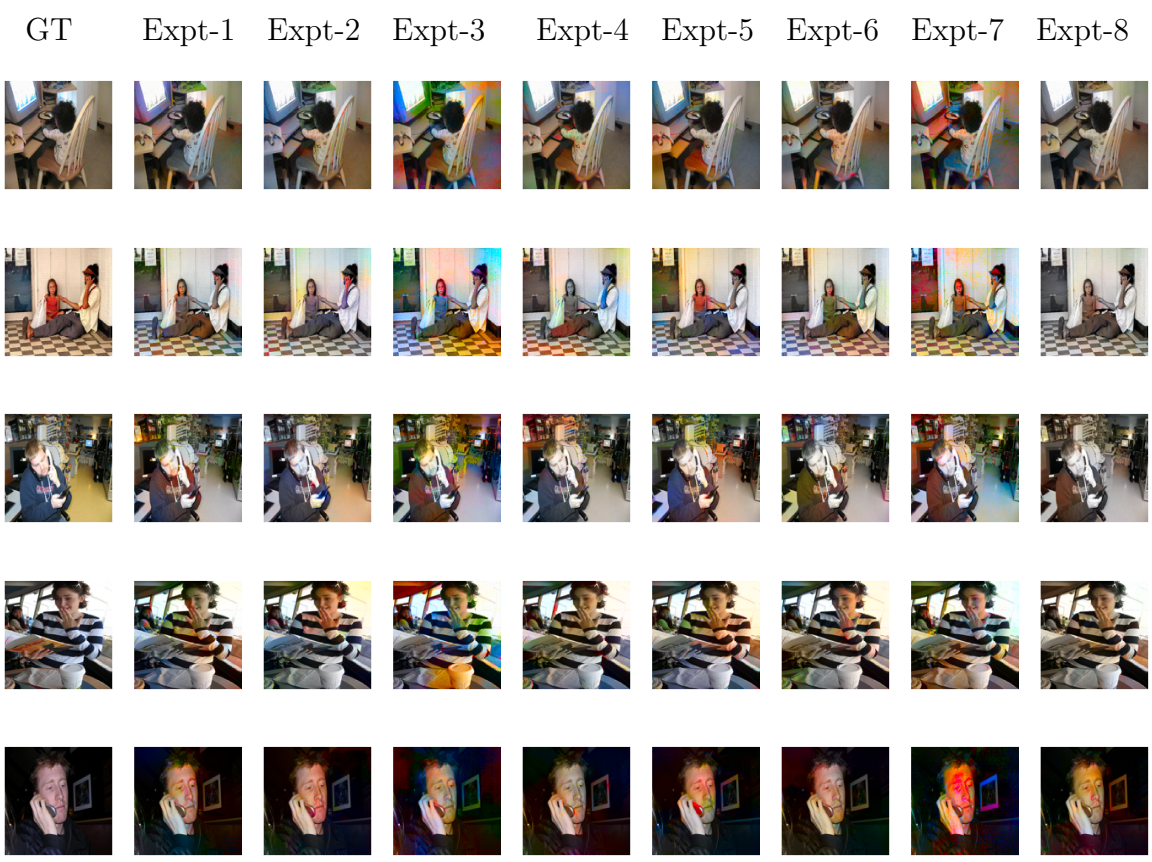


**Fig. 4.4**: Images across different experiment on VOC Datasets. We chose experiment 5 as it has maximum SSIM Score. Row 4 and Row 5 contains some of the failure cases of the experiment 5

**Fig. 4.5**: Images for our best experiment- expt-5 from Movie Datasets: Grayscale Image, Original Image, Generated Image from left to right

# Chapter 5

# Major Challenges and Future Work

From the study of prior works and our initial work in this area, we have identified some major challenges associated with the colorization task.

- Colorization is a complex task which requires all the objects in an image to be detected, localised and segmented well to color it appropriately. If the model is unable to detect what an object is, it can't decide with what color it should colorize the object and it will colorize it with a random color or leave it uncolored. It can also classify these undetected objects into frequently occurring elements like sky or tree and colorize them blue or green. Also if the model is unable to segment the exact region of a particular object, it can have a hard time deciding when to stop colorizing this object and that's when colors may leak outside of the objects.

  **Possible Solution:**

  - COCO2014 is a large dataset designed for the tasks like object detection, localization and segmentation. We can use pre-trained networks on this dataset as the generator of our model.
  - We can also perform multi-task learning with our model. The tasks would be object detection, localization (bounding box prediction), segmentation and colorization. All these tasks would use a single backbone network with separate

heads. As the backbone is the same, lot of information sharing can happen from other tasks to the colorization task improving its performance.

  - We can also use a pre-trained segmentation model to get the segmentation map for the input image, then this input image with the segmentation map can together be fed to the generator of colorization model. This way the generator knows about different regions of image beforehand and hence can give a nicely colorized output image.

- The colorization we have done so far has been frame by frame. It considers all the frames as independent. As the colorization is a one to many task, same type of objects in continuous frames could get colorized differently. When we compile those colorized frames into a video, some fluctuations in colors can be observed.

  **Possible Solution:** We can extract temporal information from continuous video frames using sequential models such as BiLSTM, GRU and BERT. One more method to extract temporal information is 3D convolutions.

  These are the approaches we think can be followed to further improve the quality of colorization.

# References


[1] Z. Cheng, Q. Yang, and B. Sheng. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 415–423, 2015.

[2] Y. Endo, S. Iizuka, Y. Kanamori, and J. Mitani. Deepprop: Extracting deep features from a single image for edit propagation. In *Computer Graphics Forum*, volume 35, pages 189–201. Wiley Online Library, 2016.

[3] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340, 2001.

[8] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu. An adaptive edge detection based colorization algorithm and its applications. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 351–354, 2005.

[9] S. Iizuka, E. Simo-Serra, and H. 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):1–11, 2016.

[10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[12] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In *European conference on computer vision*, pages 577–593. Springer, 2016.

[13] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[14] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *ACM SIGGRAPH 2004 Papers*, pages 689–694. 2004.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[16] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum. Natural image colorization. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 309–320, 2007.

[17] Y. Qu, T.-T. Wong, and P.-A. Heng. Manga colorization. *ACM Transactions on Graphics (TOG)*, 25(3):1214–1220, 2006.

[18] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[19] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[22] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[23] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363. PMLR, 2019.

[24] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[25] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*, 2017.