Study of Three State-of-the-art Image Colorization Techniques of Last Three Decades

Term Paper: CSci 5561 - Computer Vision

Md Jahidul Islam (5188102)

Abstract

Our project (team: YAL) is on designing a generative adversarial network (GAN) based model for automatic image colorization. I worked on the generator part; while doing so, I explored three colorization algorithms, which have been considered the state-of-the-art over three different time-spans. This report briefly discusses these algorithms, related implementation issues, results, and lessons learned while exploring these methods.

1 Introduction

Image colorization [1, 2, 3] refers to colorizing a given gray-scale image so that it appears real. A large amount of photographs, videos and movies, mainly antique, lack color; image colorization can provide a modern and vivid view to these images. In addition, surveillance cameras often capture (or store) gray-scale images for convenience. Several underwater inspection and surveillance applications [4, 5] often have to deal with color-less images due to lack of visible light in deep-water. Robust and efficient image colorization techniques can be used in these applications with substantial benefits.

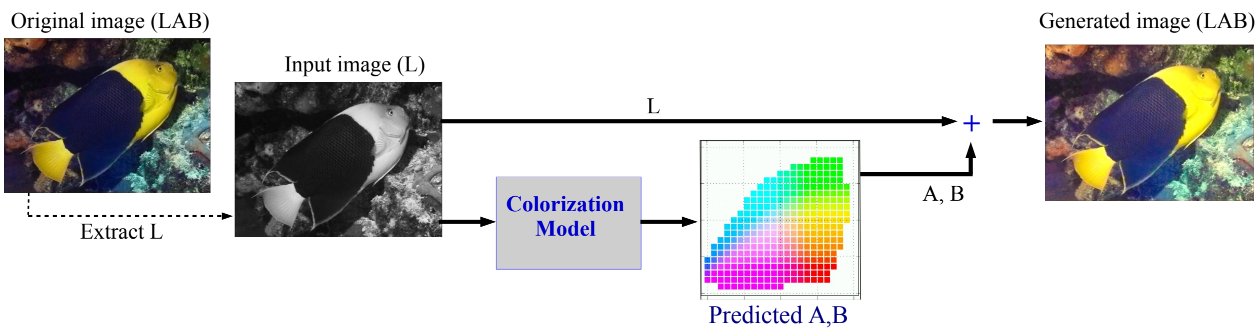


Figure 1: Basic image colorization procedure is shown. LAB color-space is generally used for convenience (*i.e.*, one less unknown dimension); given the lightness channel L, task for the colorization model is to predict A and B channels so that the colorized image appears natural. (Further discussion on discretized values for A, B channels are provided in Section 4)

Colorizing a gray-scale image (*i.e.*, only intensity values are known) is a difficult and ill-posed problem. Computer vision community have approached this problem in different ways over the last few decades [1, 2, 3, 6, 7, 8]. Before the advent of deep-learning [9], researchers have tried many classical techniques [6, 7, 8, 10, 11] to capture relationships between color components (RGB or LAB) and image level features. Due to multi-modality and ill-posed nature of the problem, optimization based techniques [10, 6] and probabilistic models [11] were the only ones that achieved decent colorization performance in few specific applications. However, overall performance of these techniques, in general, were still poor due to the high non-linearity and abstract nature of color-feature relationship.

Recently, deep-learning based image colorization techniques [1, 2, 12, 13], trained over millions of images, have shown significantly better performance over the earlier classical methods. For instance, the current state-of-the-art, 'colorful colorization' [1], can fool a human observer 32% of the time in a *colorization Turing-test* scenario. Additionally, the generalized performance of these techniques in different lighting conditions is also very good.

However, there are still plenty of rooms for improvement. In our project, we explored GAN-based models for colorization. GAN [14] is a generator-discriminator framework for adversarial learning. Many learning-based applications have experienced a significant boost in performance with such adversarial learning. While details of our project will be discussed in the project report, this report discusses three papers I explored while working with the generator of our model.

The rest of the report is organised as follows: first, the background and related work is presented in Section 2. Subsequently, a detailed discussion on the three relevant colorization algorithms is presented in Section 3. Finally, the adopted scheme, implementation details, and related results are presented in Section 4.

2 Background and Related Work

As mentioned in the previous Section, image colorization is an ill-posed problem due to multi-modality and ambiguity. While some natural objects commonly hold the same color (e.g grass is *usually* green), many are left up for interpretation. For example, given a gray-scale image of someone wearing a dark colored shirt, there is no way of figuring out the true color. Instead, the objective is to come up with a colorization that appears real, *i.e.*, natural.

User-based approaches [10, 8, 15, 16] were popular for being fast and relatively accurate as user can provide a good prior for the inherent color distribution. However, these methods are not applicable for large scale automatic colorization, which led researchers to adopt optimization and probabilistic approaches [6, 3, 11]. These approaches model a likelihood based color approximation for each pixel given the neighborhood information. Few methods introduce additional step for spatial coherency through image based segmentation as well. However, overall colorization performance of these approaches are not very appealing [17] for general usage in a large scale. This is because the prior distribution of color-space is domain-dependant; for instance, face images, underwater images, outdoor and satellite images, all have different color distributions. Besides, it is difficult to capture the highly non-linear and abstract color-feature relationships without large-scale training.

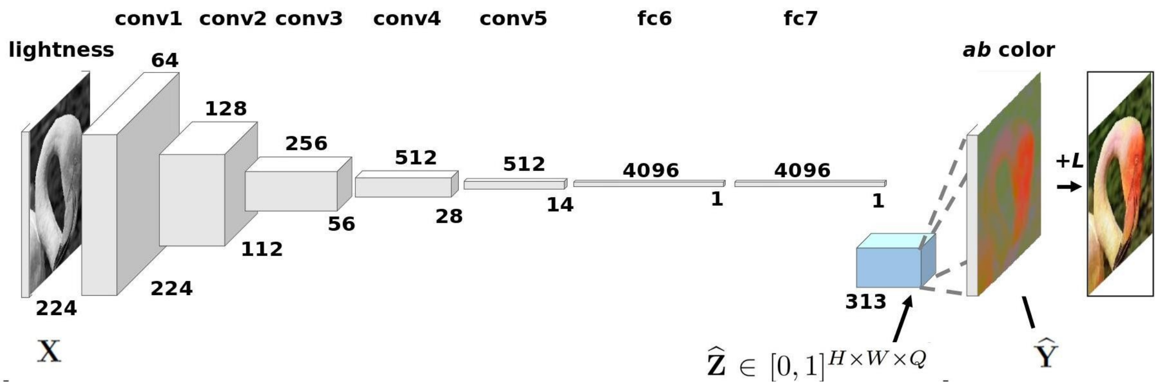


Figure 2: A simple architecture [1] for image colorization with 5 convolutional and 2 fully connected layers. Given input X (gray-scale image 224×224), the first layer generates 64 feature maps, second layer 128 feature maps, and so on. Spatial down-sampling is performed through layers to capture more abstract features; eventually, up-sampling to appropriate dimensions is performed at the final layer to predict A, B values (Z) for each pixel.

In recent times, deep-learning based approaches [1, 2, 12, 13] have produced significantly better colorization performance as they can extract highly non-linear spatial relationships if trained over large datasets. The convolutional layers learn appropriate filters to produce good feature-space representations from raw images. These feature extraction and filtering is performed over multiple layers to explore complex spatial relationships within the image-space. Often fully connected layers are used for classification or object recognition tasks. However, for image-to-image translation tasks such as colorization, it is common to use only convolutional layers [1, 2]. Figure 2 shows a simple architecture for image colorization and overall transition of feature maps over layers. Detailed discussion on how the training is done and colorization is achieved, is provided in Section 4.

3 Relevant Aproaches and Implementation Issues

As mentioned in the previous Section, state-of-the-art image colorization techniques were based on probabilistic and optimization models during the last two decades (1995-2015). Current state-of-the-art is based on deep-learning, which exhibits significant boost in performance. These algorithms provide a chronological insight of how colorization techniques in the literature have evolved over the years. This report focuses on the following three algorithms: *colorization using optimization* [10], *colorization via multimodal predictions* [6], and *colorful image colorization* [1].

3.1 Colorization using Optimization (2004)

Colorization using optimization [10] is one of the first algorithms that achieved good colorization performance. It is based on the assumption that similar regions (i.e, intensity values) in the image-space are likely to have similar colors. An optimization problem based on *quadratic cost function* is formed to govern image segmentation and tracking of similar regions. The target is to minimize the difference between color $U(r)$ at pixel r and weighted average of colors at neighboring pixels, using the cost function: $C(U) = \sum_r [U(r) - \sum_{s \in Nighbor(r)} U(s) \cdot w_{rs}]^2$. (here, w_{rs} is a Gaussian kernel)

This method is fast and performance is decent enough; however, it is not automatic. That is, user provides a patch of colors within the image-space to begin with- which significantly boosts the performance. First, the user indicates a template by scribbling the desired color in the interior of the region. Then, using these user supplied markers, this technique automatically propagates colors to the remaining pixels in the image sequence. A demonstration is illustrated in Figure 3.

Figure 3: Demonstrations of how colorization is performed in [10] using user-specified markers (code is available at [18])

3.1.1 Pros and Cons:

This technique introduces a clever way for assigning colors to similar regions in the image-space, without explicitly performing computationally expensive image based segmentation. Additionally, it is fast and colorization is spatially coherent. However, such insuitions are not readily applicable or extendible automatic image colorization techniques (such as ours). Therefore, it is not suitable for large scale image colorization.

3.2 Automatic Image Colorization via Multimodal Predictions (2008)

Automatic image colorization via multimodal predictions [6] works as follows: first, it discretizes the color-space (LAB) into 73 bins and measures approximate distribution of colors for each pixel based on the training data. Then, each local description around a pixel is modeled by a 30-dimensional vector based on SURF descriptor. Subsequently, both the distribution of color-space and corresponding feature vectors are fed to a color predictor. The conditional probability of the color c_i at pixel p, given the local description $\mathbf{v}$ of its grey-scale neighborhood is expressed as: $P(c_i|\mathbf{v}) = [\sum_{j:c_j \in B_i} K(\mathbf{v}_j, \mathbf{v}) / \sum_j K(\mathbf{v}_j, \mathbf{v})]$, where B_i denotes the color bins and K is a Gaussian kernel.

Following the prediction, a refinement step is performed based on expected variation of color at each pixel. Finally, graph-cut algorithm is used to refine the image coloring at the global level. A demonstration is illustrated in Figure 3.

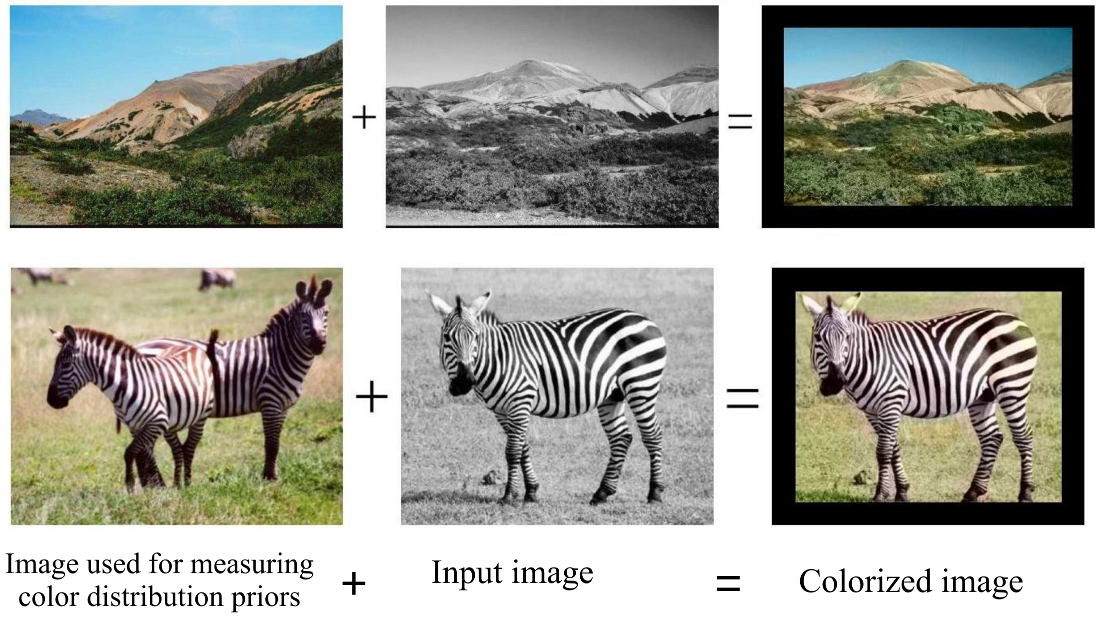


Figure 4: Demonstrations of how colorization is performed in [6]. Color distribution prior is estimated from a given image (or a set of training images); then, color projection based on this distribution is performed to predict colors at each pixel

3.2.1 Pros and Cons:

This algorithm is fast and robust to texture noises. However, its colorization performance heavily depends on the color distribution prior; consequently, it often produces desaturated coloring if the given reference images are drastically different compared to the test image(s).

3.3 Colorful Image Colorization (2016)

Colorful image colorization [1] is considered as a major breakthrough on this problem. Published on 2016, it has managed to set a new benchmark for performance. It is an automatic approach that produces realistic colorizations based on a CNN-based model. Previous approaches have either relied on significant user interaction or resulted in desaturated colorizations. To address these issues, it designed a novel *loss function*, which seem to work very well with its underlying CNN-based architecture.

Furthermore, this paper introduced a notion of evaluating quality of colorization through a *colorization Turing test*, in which it can fool a human judge 32% of the time. However, one limitation is that it is not end-to-end trainable (the loss function parameters are approximated separately). As a benchmark, I investigated this model and performed thorough experiments over several datasets, which is discussed next.

4 Adopted Model, Implementation and Results

Unlike other approaches, colorful image colorization [1] poses the colorization problem as a multi-modal classification problem. The paper presents couple of CNN-based models (one is shown in Fig. 2, another in Fig. 5), with a novel objective function. The objective function is carefully designed to map the image-to-image translation problem to a classification problem. First, it takes advantage of the fact that A, B color components of LAB colorspace for natural images are concentrated in a small region, which can be discretized into finite number of bins (Q), as illustrated in the top row of Fig 5. Given the lightness channel (L_p) of a pixel p, its A, B pair corresponds to a particular bin (out of 313 bins in total), which is mapped to a 1-hot vector (Z_p). Consequently, task of the classification model, is to predict which bin each pixel corresponds to. That is, the output is a 313-mode probability distribution ($\hat{Z}_p$) for each pixel p. The objective function is modelled as a cross entropy loss between Z and $\hat{Z}$, expressed as follows:

$$L_{col}(Z, \hat{Z}) = -\sum_{p} Z_p \sum_{q \in Q} Z_p[q] log(\hat{Z}_p[q])$$

This cross-entropy loss can be further augmented with class rebalancing, to encourage rare colors. The detailed model specification, as shown in the bottom-row of Fig. 5, is an 8-layer CNN architecture where each conv layer refers to a block of 2 or 3 repeated conv and ReLU layers [19], followed by a BatchNorm layer [20]. The network has no pool layers; all changes in resolution are achieved through spatial down-sampling or up-sampling between conv blocks.

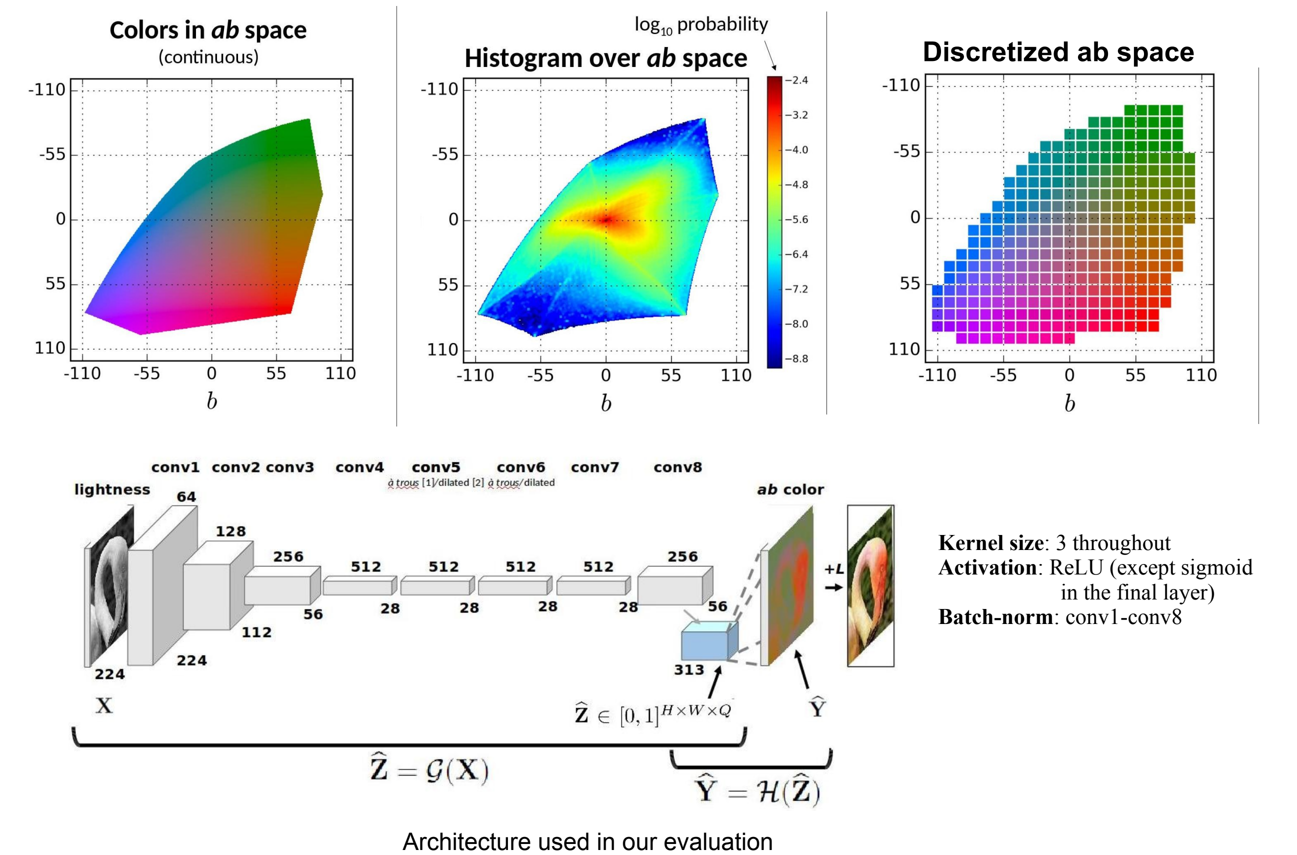


Figure 5: The top row shows how the distribution of A, B color components is discretized to finite number of bins; bottom row shows the model architecture adopted in our evaluation. (figures are taken from the paper [1]; code is available at [21])

4.1 Implementation Details and Results

While working on designing a generator for our GAN-based model, I investigated this model with different objective functions (L_1 loss, L_2 loss, least-squared loss, etc.) instead of their cross-entropy based loss function. This is due to the fact that in a GAN-based model, *discriminator* expects an image from the *generator*, and tries to discriminate it as real or fake in order to force the generator to get better. Therefore, rather than adopting their classification model directly, I implemented their architecture using objective functions based on L_1, L_2, least-squared loss (so that it outputs an image, not classification probabilities).

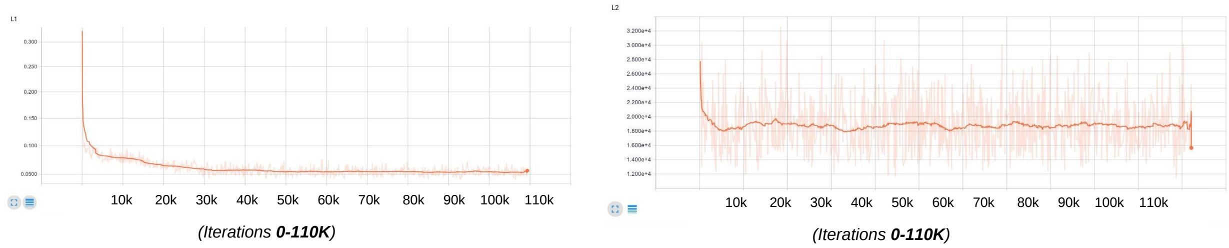
Given an input image I, the network is fed with its L channel (I_L); the output layer of the network is adjusted to predict $\hat{I}_{AB}$. We have found L_1 and L_2 loss functions perform quite well with this model. These loss functions between true I_{AB} and $\hat{I}_{AB}$ can be expressed as follows:

$$L_1(I_{AB}, \hat{I}_{AB}) = \lambda \sum_p |I_{AB}[p] - \hat{I}_{AB}[p]|$$

$$L_2(I_{AB}, \hat{I}_{AB}) = \lambda \sum_p (I_{AB}[p] - \hat{I}_{AB}[p])^2$$

Here, λ is a normalization constant. CelebA datasets [22] is used for training primarily, that has $195K$ training examples (a total of $202K$ images); In a separate training, a set of $200K$ images from Places2 dataset [23] were also used for training[1]. Training was performed using two 1080 gpus (in a core-i7 machine having 64GB RAM); training time for $100K$ iterations with a batch size 32 was about 2-3 days for each trial. The implementation is done using tensoflow [25] in python.

[1]Larger datasets like ImageNet [24] and full Places2 challenge dataset were not used for the project due to time constraint; however, we are planning to train our final GAN-based model over these datasets during the summer.



(a) Function value od L_1 loss over iterations (b) Function value od L_2 loss over iterations

Figure 6: Convergence of different cost functions with the adopted colorization model (as seen on tensor-board)

Figure 7: Results for the colorful colorization model [1] with L1 (top) and L2 loss (top); First 4 columns show few examples from the test set of CelebA [22] dataset, while the rest (3) are chosen from test set of Places2 [23] dataset

We found that this model performs better with L_1 loss function compared to L_2 loss. Additionally, training is much smoother for L_1 loss as well, as evident from Fig 6. This might be because of the averaging effect of 2-norm which causes a blurry colorization (similar phenomena is discussed in [1]). The results on few test cases are shown in Fig. 7.

It is to be noted that these results were found while designing it as a generator only, we will discuss results for our GAN-based models in the project report. Currently, the loss function (L_{col}) for classification model of the original paper is being tried out, to investigate how it performs over these datasets.

4.2 Critics

As mentioned, the classification model presented in the original paper [1] was not directly applicable (as generator) for our GAN-based model. Therefore, we kept it as a image-to-image translation model and tried L_1- and L_2-loss (instead of cross-entropy based loss) function. Additionally, by doing so, we make this model end-to-end trainable (their proposed model is not end-to-end trainable). Currently, as a generator alone, this model works quite well, specially with L_1 loss function. However, further improvement is possible with other GAN-based models, which will be discussed elaborately in our project report.

5 Conclusion

This report discusses three papers that I explored while designing generator for our GAN-based model (project group: YAL). These three algorithms, namely colorization using optimization, colorization via multimodal predictions, and colorful image colorization. The later, being the current state-of-the-art and based on deep-learning, was adopted in our evaluation. The problem formulation is provided, implementation details are discussed and results are presented with necessary illustrations.

References

[1] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer, 2016.

[2] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 415–423, 2015.

[3] Aurelie Bugeau, Vinh-Thong Ta, and Nicolas Papadakis. Variational exemplar-based image colorization. *IEEE Transactions on Image Processing*, 23(1):298–307, 2014.

[4] Huimin Lu, Yujie Li, and Seiichi Serikawa. Underwater image enhancement using guided trigonometric bilateral filter and fast automatic color correction. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 3412–3416. IEEE, 2013.

[5] Luz A Torres-Méndez and Gregory Dudek. Color correction of underwater images for aquatic robot inspection. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 60–73. Springer, 2005.

[6] Guillaume Charpiat, Matthias Hofmann, and Bernhard Schölkopf. Automatic image colorization via multimodal predictions. *Computer Vision–ECCV 2008*, pages 126–139, 2008.

[7] Qing Luan, Fang Wen, Daniel Cohen-Or, Lin Liang, Ying-Qing Xu, and Heung-Yeung Shum. Natural image colorization. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 309–320. Eurographics Association, 2007.

[8] Vadim Konushin and Vladimir Vezhnevets. Interactive image colorization and recoloring based on coupled map lattices. In *Graphicon2006 conference proceedings, Novosibirsk Akademgorodok, Russia*, pages 231–234, 2006.

[9] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[10] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM Transactions on Graphics (ToG)*, volume 23, pages 689–694. ACM, 2004.

[11] Przemyslaw Lagodzinski and Bogdan Smolka. Digital image colorization based on probabilistic distance transformation. In *ELMAR, 2008. 50th International Symposium*, volume 2, pages 495–498. IEEE, 2008.

[12] Domonkos VARGA and Tamás Szirányi. Fully automatic image colorization based on convolutional neural network. *4th Winter School of PhD Students in Informatics and Mathematics*, page 36, 2016.

[13] Jie Li, Katherine A Skinner, Ryan M Eustice, and Matthew Johnson-Roberson. Watergan: Unsupervised generative network to enable real-time color correction of monocular underwater images. *arXiv preprint arXiv:1702.07392*, 2017.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[15] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.

[16] Michael J Vrhel and HJ Trussell. Color correction using principal components. *Color Research & Application*, 17(5):328–338, 1992.

[17] Aditya Deshpande, Jason Rock, and David Forsyth. Learning large-scale automatic image colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 567–575, 2015.

[18] Anat's colorization. `http://www.cs.huji.ac.il/~yweiss/Colorization/`. Accessed: 04-20-2017.

[19] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[21] Colorful colorization. `http://richzhang.github.io/colorization/`. Accessed: 04-20-2017.

[22] Celeba dataset. `http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html`. Accessed: 04-20-2017.

[23] Places2 dataset. `http://places2.csail.mit.edu/download.html`. Accessed: 04-20-2017.

[24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[25] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.