Generation and Extraction of Color Palettes with Adversarial Variational Auto Encoders

Ahmad Moussa and Hiroshi Watanabe

Graduate School of Fundamental Science and Eng., Waseda University, Tokyo, Japan,
ahmad.moussa@fuji.waseda.jp, hiroshi.watanabe@waseda.jp,

Abstract. The process of creating a meaningful and perceptually pleasing color palette is an incredibly difficult task for the inexperienced practitioner. In this paper we show that the Variational Auto Encoder can be a powerful creative tool for the generation of novel color palettes as well as their extraction from visual mediums. Our proposed model is capable of extracting meaningful color palettes from images, and simultaneously learns an internal representation which allows for the sampling of novel color palettes without any additional input.

Keywords: Variational Auto Encoder, Color Palettes, Generative Adversarial Networks

1 Introduction

Color palettes are an important part of every designer's workflow. The interaction between different colors in a palette generally dictates the overall mood of a digital artwork. Practice allows one to build an intuition for selecting colors that "go well together" in a perceptually pleasing manner, yet it is incredibly difficult to explain why certain colors fit well together, the more so formalizing this intuition into algorithmic rules. To this end, we examine the possibilities of implicitly learning these rules with a neural network.

Even though classifying a palette as 'visually pleasing' is highly subjective, the human eye is naturally drawn to certain patterns and logically arranged sets of colors. Color palettes assembled by humans usually have a very distinct structure according to the rules of color theory.

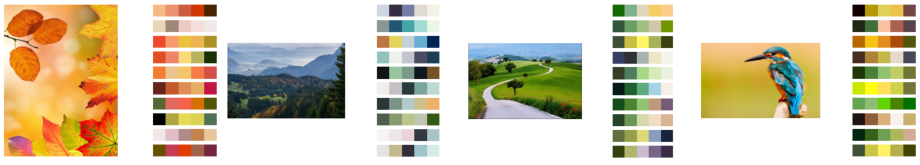

Fig. 1. Given an input image our model can generate multiple coherent color palettes.

Even though this imposes preliminary guidelines on the selection of colors that make up a good color palette, this task becomes more difficult when it has to be matched to certain types of visual media such as images, photographs,

digital artworks, posters, etc. Not only do the colors have to be meaningful with regard to each other, but also with regard to the source medium.

Even though convenient tools exist to make these color selections, it is unlikely that an inexperienced practitioner will be able to make a good selection. In the next section we briefly discuss some algorithms that are capable of creating a color palette from a digital image

2 Related Work

The high difficulty of extracting a nice color palette from a digital medium gave rise to many websites and online services to algorithmically generate such a color palette from an image. Arguably, the most popular algorithms utilized are Median Cut [8] and Modified Median Cut Quantization (MMCQ) [1].

Closely related to MC are clustering algorithms. This broad family of algorithms is widely used for the color quantization of digital images, which attempts at reducing the number of different colors in an image, and represent the original image with a reduced palette, subsequently enabling efficient compression. However, this reduced palette is usually optimal for quantization but not very attractive as a stand-alone palette. Normally, the majority of 3 dimensional clustering algorithms can be used for this purpose. Popular examples are: k-means clustering [13] and Oct-tree quantization [5].

Other deep learning methods that have been tangentially applied in this regard are, most notably: Colormind, which attempts at performing color infill with Conditional Adversarial Networks, PaletteNet [4] which attempts to recolorize an image with a given palette, as well as Text2Color [3] which attempts to generate color palettes from a given sentence using deep learning. None of which attempt the generation of a color palette from a given image.

3 Proposed Model Architecture

3.1 Variational Auto Encoder

At it's core, our proposed model consists of a Variational Auto Encoder [11]. VAEs have been proven to be powerful generative deep learning models in multiple different domains (insert citations here) as well as the construction of powerful interpolatable latent spaces. VAEs learn the parameters of a probability distribution over the input data space. VAEs are trained not only via a reconstruction term but additionally with a regularization term, which essentially enforces structure in the latent space, such that similar data samples are encoded as neighbouring points in the latent space. Conventional auto-encoders give no guarantee on the interpretability and structure of the latent space, whereas VAEs guarantee that any randomly sampled latent vector will produce a meaningful data sample.

In technical terms, the input data to the encoder is designated by x, the intermediate latent vector by z and it's weights and biases by θ. Thus, the encoding neural network can then be represented by $q_\theta(z|x)$. Equivalently, the decoder can be represented by $p_\phi(x|z)$, where it's input is the latent vector z

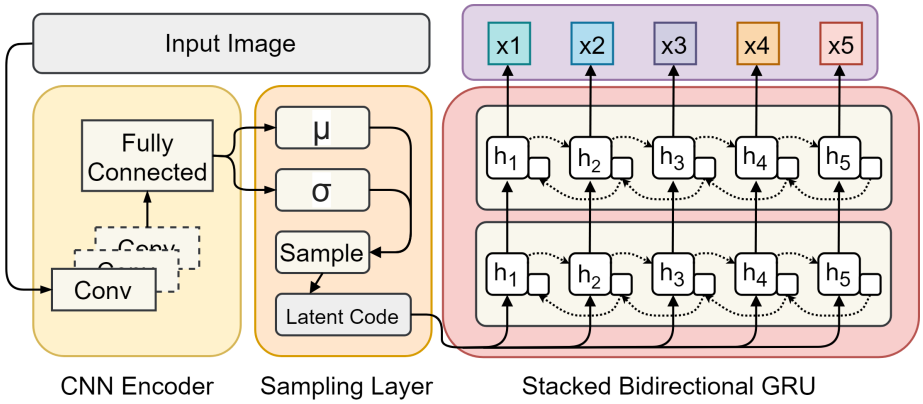

Fig. 2. The overall architecture of the proposed variational autoencoder consists mainly of three parts. A convolutional encoder, an intermediary sampling layer that allows us to generate the latent codes and a bidirectional LSTM decoder which will utilize the latent code to generate a color palette.

with weights and biases ϕ. To measure how accurately the decoder has reconstructed the expected output we utilize the log-likelihood $logp\phi$ additionally the loss function contains a regularizing term that specifies how much information was lost between $q_\theta(z|x)$ and p(z). In variational inference this amounts to the evidence lower bound function (ELBO):

$$-E_{z\sim q_\theta(z|x)}\left[\log p_\phi\left(x \mid z\right)\right] + KL\left(q_\theta\left(z \mid x\right) \| p(z)\right) \tag{1}$$

Due to sampling the latent vector, obtaining the gradient through the ELBO is not possible. This problem can be avoided by the re-parametrization trick with the specification that p(z) is a standard normal distribution with mean zero and variance one:

$$\epsilon \sim \mathcal{N}(0, I), z = \mu + \sigma \odot \epsilon \tag{2}$$

3.2 Proposed Model Architecture

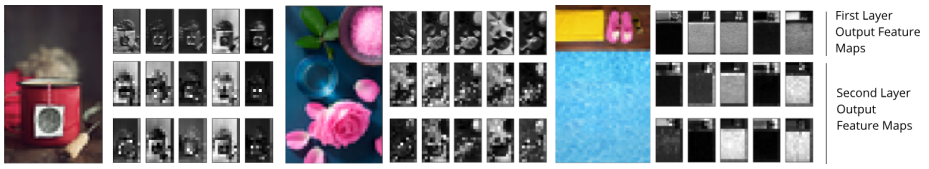

Fig. 3. We inspect the output of the first two convolutional feature maps when running an input image through the encoder (5 and 10 filters respectively). The feature map activations show that differently colored areas are recognized.

Utilizing a convolutional encoder makes strong assumptions about the shape of the input to our model. Even though the detection and localization of specific

objects is not the main goal in our case, we care for the color of pixels as well as the amount and number of different colors that could occur. Extracted high level features in our case might be blobs of color of different sizes and color gradients running throughout different parts of the image. Figure 3 shows the feature maps extracted from the first and second convolutional layers in our encoder.

We also make a strong assumption that a digital color palette is a sequence of numbers that have a strong correlation between each other. To this end, we utilize a bidirectional GRU [2] to generate the output sequence of colors. We choose the GRU over an LSTM [10] for a number of reasons: the GRU exposes the entire hidden state, we find that this is beneficial, is suitable for modelling short sequences, and thirdly, is overall computationally more efficient. We observe the success of the GRU in [3]. Additionally, we opt for a bidirectional unit, as the colors in a palette have a correlation in both forward and backward parsing order. In this manner the encoder will generate a hidden state z from the input image which will be passed to the RNN decoder. The decoder consumes the latent code to generate a sequence of colors x = {x1, x2, x3, x4, x5} where each item in this sequence is a 3 dimensional vector which specifies the color component values of the color. A complete overview of the architecture can be observed in figure 1.

3.3 VAE GAN

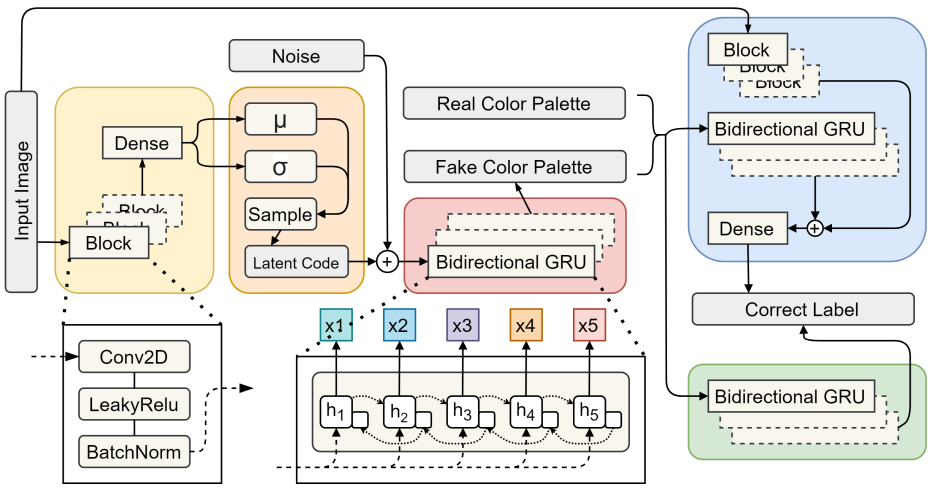

Fig. 4. We extend our initial architecture and append two discriminators. One that learns how well the color palette fits to the input image, and another that learns to evaluate the palette itself. An auxiliary noise vector allows us to generate differing color palettes for the same latent vector when interpolated.

We propose a variation of our model that is trained adversarially. This type of architecture was previously introduced in [12]. Instead of utilizing a distance metric for the reconstruction term we use a learned similarity metric.

Generative Adversarial Networks [6] consist of two neural networks that are pitted against each other. A generator that attempts to generate novel data samples by up-sampling a latent vector $x = Gen(z)$, and a discriminator that attempts to classify generated samples as real or fake, such that $y = Dis(x)\epsilon[0, 1]$ and formally this objective can be stated as follows:

$$\mathcal{L}_{\text{GAN}} = \log(Dis(x)) + \log(1 - Dis(Gen(z))) \tag{3}$$

Similarly to discriminating between real and generated images, our discriminator has to learn a meaningful similarity metric that is capable of setting apart real color palettes from fake ones. We also propose a secondary discriminator that not only receives a color palette but also the input image that was fed to the encoder. It's purpose is to fuse image and palette and classify this combined representation as real or fake. Only discriminating the generated palette as real or fake is not informative enough to train a strong generator capable of generating palettes that have a meaningful relationship towards the input image.

In a similar manner to [12], to train this adversarial model we replace the VAE reconstruction loss term with a Gaussian Observation Model, such that:

$$p\left(Dis_l(x) \mid z\right) = \mathcal{N}\left(Dis_l(x) \mid Dis_l(\tilde{x}), \mathbf{I}\right) \tag{4}$$

Where Dis_l is the hidden representation of the l-th layer of the Discriminator, $\tilde{x}$ is the generated data sample, and I the identity covariance. The loss term is thus:

$$\mathcal{L}_{Ilike}^{Dis_l} = -E_{q(z|x)}\left[\log p\left(Dis_l(x) \mid z\right)\right] \tag{5}$$

And the final criterion becomes:

$$\mathcal{L} = \mathcal{L}_{prior} + \mathcal{L}_{llike1}^{Dis} + \mathcal{L}_{GAN1} + \mathcal{L}_{llike2}^{Dis} + \mathcal{L}_{GAN2} \tag{6}$$

Where L_{GAN1} and L_{GAN2} are the respective terms for each of the two discriminators.

We follow the exact training regime detailed in the [12], except that we train the two discriminators with Mean Squared Error instead of Binary Crossentropy as proposed in Least Squares GANs [14]. Additionally, we perform one sided label smoothing, train both generator and discriminators with the Adam optimizer, using a learning rate of 0.0005 for the generator and 0.002 for the discriminators, according to the Two timescale update rule [9], with a beta1 value of 0.5. These changes incur a stable training procedure.

4 DATASET

To train our model we require pairs of images and their corresponding palettes. It is possible to create the dataset by taking an arbitrary number of images and computing the corresponding palettes with the quantization algorithms mentioned in section 2. Even though this would allow us to create an extensive dataset, the computed palettes do not satisfy the properties that we discussed

earlier. An alternative would be utilizing websites that allow the creation of a palette from a given image, but it is not transparent what type of algorithm is used to obtain the palette, and thus refrain from utilizing them. For our purposes we collect a dataset where the colors of each palette have been selected and arranged manually by a person. Two websites that offer such palettes are ColorPalettes [1] and DesignSeeds [2] (used in [4]).

In a similar manner to PaletteNet [4] we scrape Design Seeds for a total of 3200 images, and ColorPalettes for a total of 4000 images. The caveat is that images on ColorPalettes are accompanied by palettes that contain 5 colors rather than 6 in DesignSeeds. This causes a problem when we try to combine both datasets. We randomly drop one of the 4 intermediate colors in the DesignSeeds Dataset. We also convert images and corresponding palettes to the CIE LAB color space[7] and normalize the pixel values to a [0,1] range.

5 RESULTS

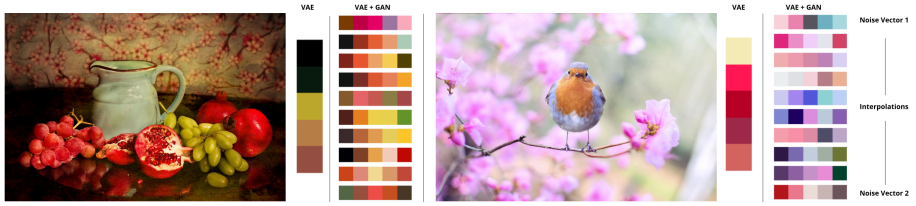

Fig. 5. The regular VAE generates plausible palettes but does not generalize well to new data. Sampling random noise for the auxiliary vector in the adversarial model allows for the generation of multiple plausible palettes for the same image.

We conduct a number of experiments to test our model qualitatively by encoding new images, and generate color palettes from them. Additionally, we explore the learned latent space of each of the two models. Examples of palettes extracted form images can be seen in fig. 5. The VAE produces a plausible palette but does not generalize well to new data. The adversarial model leverages this by being able to generate a slew of palettes from which one can choose.

Secondly, to test the learned latent space we perform interpolation on some latent vectors for both models. Figure 6 shows the results. We generate two random latent codes from a normal distribution with mean zero and variance one, and interpolate them linearly [**?**]. Interpolations show a smooth transition from one palette to another, with several meaningful intermediary palettes and exemplify that a meaningful latent space has been learned. The adversarial variant also allows the interpolation along two axes which are the latent code and the auxiliary noise vector. We observe that the auxiliary noise vector generally controls the brightness of the decoded palette.

A nice property which allows us to control the colorfulness of sampled colors is simply by increasing the sampling interval of the latent code. Samples drawn

[1] https://colorpalettes.net/
[2] https://www.design-seeds.com/

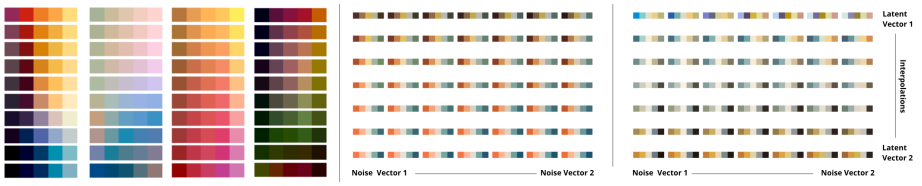

Fig. 6. Our model presents a continuous latent space that is interpolat-able. We sample random latent vectors and decode them and their interpolations. The addition of an auxiliary noise vector alongside the sampled/encoded latent vector in the adversarial model allows us to interpolate between the two of them to generate fine variations of the same palette. We found that interpolating the noise vector while keeping the latent vector fixed resulted in change of variation of the decoded color palette.

from an interval that is tighter around the mean have duller colors and are more likely to make meaningful palette, whereas samples drawn from a gradually larger interval are more colorful.

6 EVALUATION

Qualitative evaluation of our model is difficult, but can be leveraged by subjective inspection of the generated results. We conduct two surveys with 12 participants to evaluate the quality of the generated palettes based on two criteria: how well they fit to a given input image, and how coherent they are by themselves. Table 1 shows the results.

Table 1. Average Mean Opinion Score assigned to the generated palettes based on how well the a given palette fits to the input image as well as how coherent each palette is individually.

	MOS (1 to 5)		
Score	***Human***	***VAE***	***VAE-GAN***
Fittingness	4.514	3.168	1.82
Prettiness	4.008	3.514	3.34

In the first survey we presented the surveyee with an image and three palettes. One being a palette created by a person (palette from test set) and the two other being the palettes generated by our two models. The objective was to rate each palette with a score from 1 to 5, which indicates how much the palette fits to given image. We select 10 images at random from the test set and run our models on them to generate the palettes. Additionally, the surveyee is not told which color palette is from what origin. Both survey results unsurprisingly indicate that the ground truth palettes are superior in either aspect. This is because color accuracy is non-negotiable for the discriminating human observer.

7 CONCLUSION

In this paper we presented two deep learning models that can generate novel color palettes from given images, as well as generate them at random from a

learned latent space. We also showed that this latent space is meaningful and continuous and can be sampled in a number of different ways.

For an expert it might take several minutes to create a color palette from a random image, whereas the proposed framework can suggest several tentative color palettes in an instant. Beyond this, utilizing a dataset consisting of color palettes that were handcrafted by individuals, a deep learning approach is capable of capturing aspects of the human decision process. This is impossible to mimic with a hand-engineered algorithm.

References

1. Bloomberg, D., Leptonica: Color quantization using modified median cut (2008)
2. Chung, J., Gulcehre C., Cho, K., Bengio, Y.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling (2014) 1412.3555
3. Cho, W., Bahng, H., Park, D., Yoo, S., Wu, Z., Ma, X., Choo, J.: Text2Colors: Guiding Image Colorization through Text-Driven Palette Generation (2018)
4. Cho, J., Yun, S., Lee, K., Choi, J.: PaletteNet: Image Recolorization with Given Color Palette (2017) 1058-1066. 10.1109/CVPRW.2017.143.
5. Gervautz, M., Purgathofer, W.: A Simple Method for Color Quantization: Octree Quantization (1988) Magnenat-Thalmann, N., Thalmann, D.: New Trends in Computer Graphics 219–231 Springer Berlin Heidelberg 978-3-642-83492-9
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.. (2014) Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., and Weinberger, K.Q. (eds.), Advances in Neural Information Processing Systems 27, pp. 2672–2680. Curran Associates, Inc. (2014)
7. Gowda, S.N., Yuan, C.: ColorNet: Investigating the importance of color spaces for image classification (2019) 1902.00267
8. Heckbert, P.: Color Image Quantization for Frame Buffer Display (1982) Association for Computing Machinery 297–307 10.1145/965145.801294 SIGGRAPH Comput. Graph.
9. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium (2018) 1706.08500
10. Hochreiter, S., and Schmidhuber, J.: Long Short-term Memory (1997) Neural Computation 1735-80 10.1162/neco.1997.9.8.1735
11. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. In Proceedings of the International Conference on Learning Representations (2014)
12. Larsen, A.B.L., Sønderby, S.K., Larochelle, H. Winther, O.: Autoencoding beyond pixels using a learned similarity metric (2016) Proceedings of The 33rd International Conference on Machine Learning, in PMLR 48:1558-1566
13. MacQueen, J.: Some methods for classification and analysis of multivariate observations (1967) Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics 281–297 University of California Press
14. Xudong, M., Qing, L., Haoran, Xie., Lau, R. Y. K., Wang, Z., Smolley, S.P.: Least Squares Generative Adversarial Networks (2017) 1611.04076