Enhancing Image Colorization Using Deep Learning Method

Kevin Robles
Winona State University
175 W Mark St, Winona, MN 55987
kevin.robles@go.winona.edu

Bradley Budach
Winona State University
175 W Mark St, Winona, MN 55987
bradley.budach@go.winona.edu

Abstract

This paper outlines a project aimed at improving methods for automatic black-and-white image colorization using deep learning techniques. Building upon prior automatic methods, we propose an approach that will hopefully improve on issues related to color contrast and variety in the final images. Our inspiration is drawn from the work of Zhang et al., who successfully employed a CNN to colorize images and keep a greater degree of color depth.

Our method seeks to improve on previous models by introducing a multi-model approach. With an initial model to categorize the predicted color profile of an image, and the subsidiary models to colorize the image weighted towards the colors seen in the profile. We plan to train these subsidiary models in a similar way to Zhang et al. who introduced changes in the loss function calculation step of the CNN to emphasize rarer colors, and therefore increase contrast. We plan to take this idea further by training models on common color profiles to increase variety and accuracy of the final images.

1. Introduction

For our project, we plan on using deep learning to automatically colorize grayscale images. There have been many attempts to solve this problem in the past. Some of them are semi-automatic, and require user input to direct the colorization, which is not easy for inexperienced users. Others are automatic, and don't require any user input to colorize the image (Shanshan).

We plan to work on an automatic method of image colorization using deep learning, and solve some of the issues that come with prior automatic methods. If we are successful, it should be difficult for a person to tell the difference between a colorized photograph created by our model and a photograph that has been taken in full color.

Our proposed method is based on past work on the problem, and will hopefully be able to improve on the issue of predicted colors being too washed out.

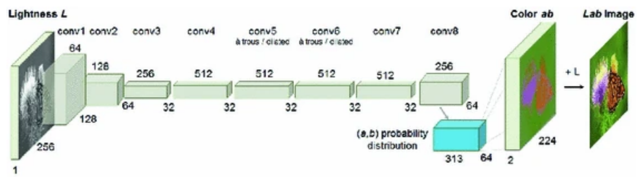
AI image colorization has many applications. Most notably it can be used for restoring old photographs or colorizing black-and-white movies, as well as other more niche applications like colorizing satellite maps.

Creating a colorization method that is adequately believable could greatly improve the work-flow of editors and people who work with black-and-white photos, and, if sufficiently simple, could be used by all sorts of individuals who want to see what their old photos would look like in full color.

2. Related Work

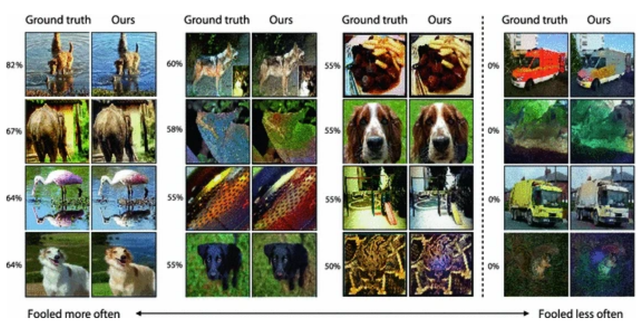
The research paper we based our work on was the paper Colorful Image Colorization by Zhang, R., Isola, P. Similarly, they used a deep learning method to automatically colorize images. Their method uses a CNN to take a grayscale image in the

Lab color space and predict the missing a and b values for the image.



The most interesting thing about their method is how they trained their model to improve color variety and brightness in the final image by adjusting the process in the backpropagation step. As they explain, “The distribution of ab values in natural images is strongly biased towards values with low ab values, due to the appearance of backgrounds such as clouds, pavement, dirt, and walls.” They continue, “ We account for the class-imbalance problem by reweighting the loss of each pixel at train time based on the pixel color rarity.”

The results of using this method are impressive. With many results looking indistinguishable from real color photos.



According to their paper, “Our full algorithm fooled participants on 32% of trials…this number is significantly higher than all compared algorithms…except for Larsson et al., against which the difference was not significant.”

Considering the impressive results of this method, we plan to use similar ideas as a basis for our proposed method and hopefully manage to improve on its success. For doing so, we have a couple of preliminary ideas on possible improvements.

Another relevant paper is Color Based Image Classification and Description by Verdaguer. This paper gave us a lot of insight into possibilities for categorizing images based on their colors and actually creating the profiles.

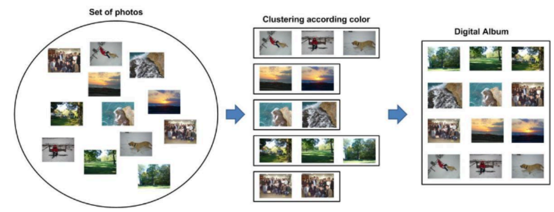


Fig. 0.2 Clustering and ordering photos

Verdaguer talks about a number of methods to categorize photos based on their colors and the applications of these methods. Colors were compared using color descriptors. The descriptor that was most relevant to our project was the Dominant Color Descriptor (DCD). According to Verdaguer, “The Dominant Color Descriptor is a compact descriptor and will let us efficiently represent the dominant colors on an image.”

The DCD is defined to be:

$$F = \{(\boldsymbol{c}_i, p_i, v_i), s\}, \quad (i = 1, 2, \ldots, N) \qquad \textbf{(4.1)}$$

where N is the number of dominant colors. Each dominant color value $\boldsymbol{c}_i$, also called centroid, is a vector of the corresponding color space component values (such as a 3-D vector in the RGB color space). The percentage p_i (normalized to a value between 0 and 1) is the fraction of pixels in the image or image region corresponding to color $\boldsymbol{c}_i$, and $\sum_i p_i = 1$.

The optional values are the color variance v_i, which describes the variation of the color values of the pixels in a cluster around the corresponding representative color, and the spatial coherency $\boldsymbol{s}$, which is a single number that represents the overall spatial homogeneity of the dominant colors in the image.

Using the DCD, images are able to be grouped into categories based on how similar they are in terms of dominant colors. We think that using a similar method, we should be able to create groups in our images with significantly different dominant colors, and train separate models for each of those groups.

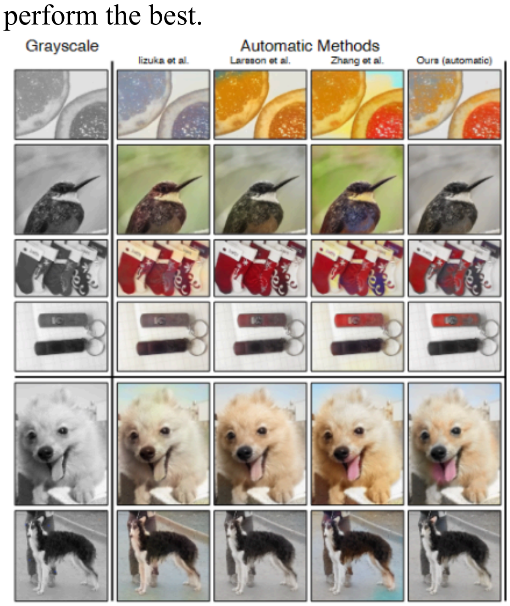
3. Methods

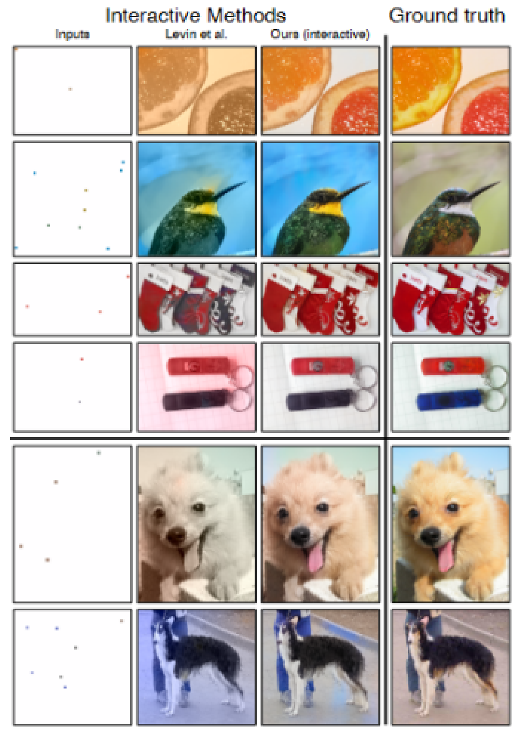
1.1. Real-Time User-Guided Image Colorization with Learned Deep Priors

By Zhang et al. in 2017. This method of image colorization is a semi-automatic method of colorizing grayscale images using a small amount of user input. Many grayscale images have features that can have various colors that could be correct, giving the user the option to choose what color specific elements of an image should be can be a good option to reproduce the most convincing colorization of an image. With this method, the user can plot which specific points of an image should be which color and the model would use those colors to fully colorize the image.

Fig. 1. Our proposed method colorizes a grayscale image (left), guided by sparse user inputs (second), in real-time, providing the capability for quickly generating multiple plausible colorizations (middle to right). Photograph of *Migrant Mother* by Dorothea Lange, 1936 (Public Domain).

This model also has the ability to colorize images automatically but when tested against other existing colorizing methods it is unable to produce accurate results in comparison to the other automatic methods. When comparing the colorizations with user input to the automatic methods however this method yielded better and more accurate results in comparison to automatic methods. When compared to other interactive methods this one was also able to perform the best.

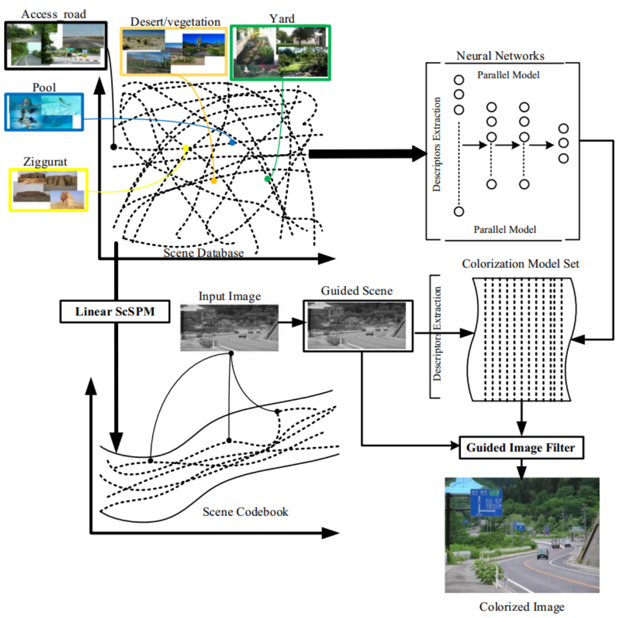




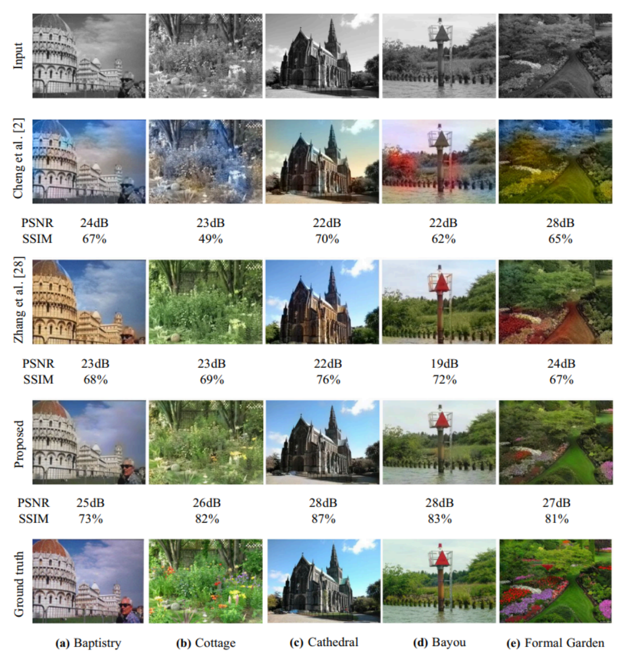
The advantage of this method is that it produces very convincing results, they may not be very accurate to the original colored images as the user may not choose the exact correct colors that match the original. A disadvantage however is that this process will slow down a lot if a user needs to colorize many images at a time due to having to input plots of color for every single image before being able to run the model.

1.2. Scene-guided colorization using neural networks

In 2022 Xia et al. proposed a fully automatic method of image colorization that uses different classes of scenes to help speed up colorization and produce more accurate results. All inputs get categorized into different scenes which are then used in the colorization process to refine the image after the initial colorization model is completed. This is very similar to other automatic colorization methods but only with scene-guided aid to help improve the accuracy of the final result.

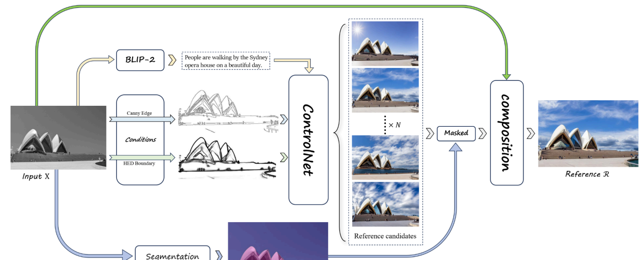


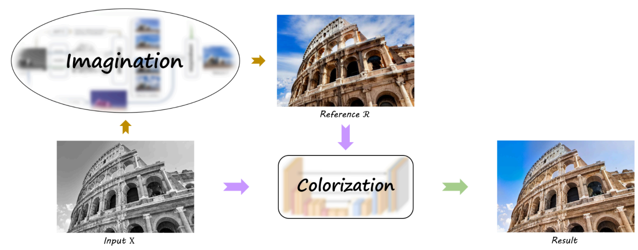
When compared with other methods this has been shown to have closer results to the ground truth in comparison to other state-of-the-art automatic colorization methods even in cases where the colorization was not very accurate. Other studies however have sometimes used the evaluation metric of people observing the image and seeing if it can fool the viewer into thinking it is the ground truth. In this study, only the error was used to measure which method had the best performance.



1.3. Automatic Controllable Colorization via Imagination

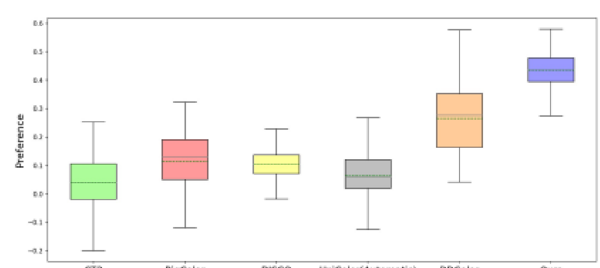
In 2024 Cong Et al. proposed a method of image colorization that utilizes an imagination module along with other pre-trained models to produce state-of-the-art image colorizations. Based on a grayscale image the model will produce various fully colored reference images that will be used in the final colorization of the original image.





When compared to other existing models this model had the highest color vividness (CF) out of all existing models that were tested. Along with those results when tested with users this model was preferred most overall.

Method	COCO-Stuff					ImageNet					In-the-wild
	FID ↓	SSIM ↑	PSNR ↑	LPIPS ↓	CF ↑	FID ↓	SSIM ↑	PSNR ↑	LPIPS ↓	CF ↑	CF ↑
Coltran [39]	13.1	0.359	13.3	0.513	38.5	15.5	0.264	8.65	0.723	**55.9**	38.0
GCP [78]	7.35	0.900	22.4	0.194	37.9	3.36	0.931	23.5	0.238	33.9	37.0
CT2 [76]	22.9	0.358	13.5	0.518	46.2	8.44	0.354	14.5	0.480	41.6	44.5
ColorFormer [31]	8.69	0.756	21.3	0.216	41.0	3.83	0.834	22.5	0.196	36.0	39.9
BigColor [35]	8.53	0.832	20.8	0.217	43.4	3.59	0.893	21.5	0.212	40.4	37.8
UniColor [27]	7.90	0.855	22.4	0.195	36.4	6.22	0.909	22.0	0.238	35.7	33.9
DISCO [79]	11.2	0.738	19.4	0.236	46.2	9.21	0.783	21.0	0.265	44.5	43.5
DDColor [32]	6.19	0.904	23.1	0.169	44.7	3.16	0.885	23.4	0.186	42.2	41.9
Ours	7.21	0.859	23.3	0.180	**47.2**	3.62	0.884	23.8	0.207	48.8	**48.9**

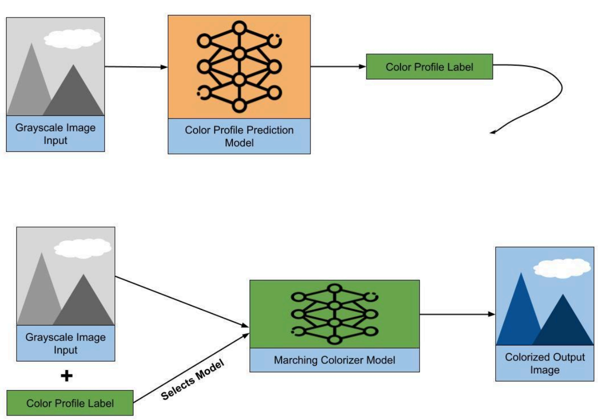


1.4. Color profile guided image colorization (Our method)

Our method to colorize images will be mostly similar to the method proposed and created by Xia et al. The model will function fully automatically because there will be a large amount of images that will be colorized with the model and it would not be viable to create plots of color for thousands of images. Our model will also expand on ideas from existing colorization models like the model proposed in 2017.

We would first create profile categories from our training data by comparing the color information in different types of pictures. We would create categories like outdoor vs indoor. Outdoor pictures tend to have a different look from indoor pictures when it comes to the color of the light and the general hues. Other categories could be more specific things like warm, cold, nature, city, etc. Anything that has a distinct difference in the types of colors present. We would then take these color weights and train models to handle those specific types of images by applying those color weights in the loss function similar to Zhang et. al.

Alongside this, we would train a model to predict the color profile of a black-and-white image. The final method would involve passing an image through this profile predictor and then passing the image to the correct model to color it with the most realistic colors possible. Hopefully, this will lead to better colorization that is adaptive to the features within the image, and prevent images from being washed out.



In the figure above is the pictured model where a grayscale image is imputed into the Color profile prediction model and that output is then used to determine which colorizer to use which will again use the grayscale image as input to produce the final colored image.

4. **Experiments**

4.1. Datasets

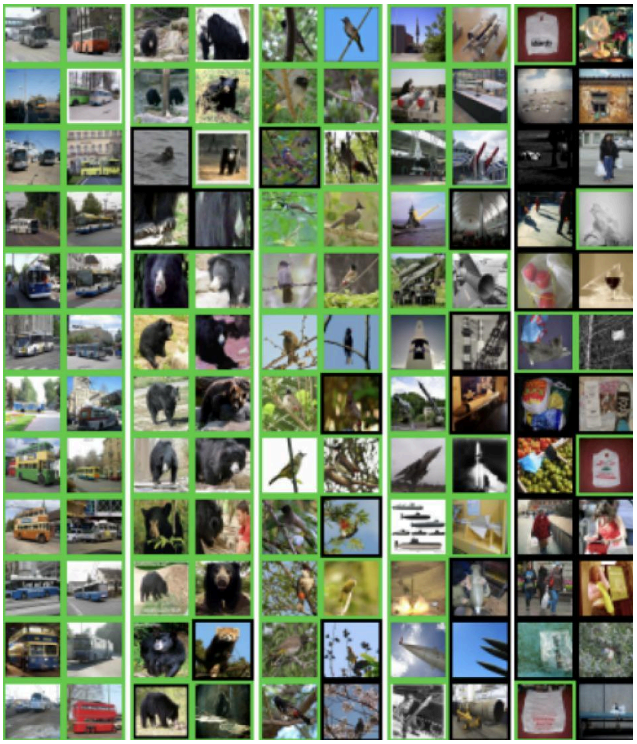
The main data set we are currently using is ImageNet. Most photos can be used for training our model and validating by simply grayscaling.

https://www.image-net.org/

In terms of pre-processing, we are converting all the images in the dataset into LAB colorspace because this is the color space we will be using to train the model. LAB colorspace also allows us to easily get grayscale images for input into the model because the L channel contains all the lightness information. We extract the L channel as our input and use the a and b channels as the ground truth labels for training. Another preprocessing step we perform is converting the image to a square to simplify the process. Since any image can be used there is a large amount of data on Imagenet that can all be used so there is no need to artificially create more data by transforming, skewing, or otherwise.

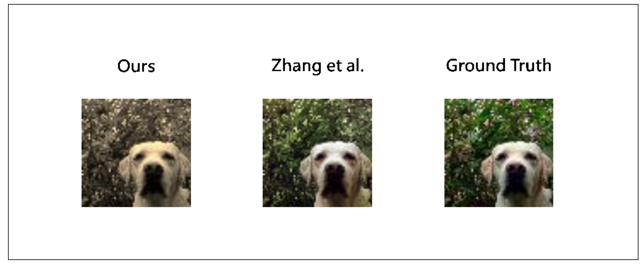
For initial testing we are using a subset of ImageNet called Tiny ImageNet which contains 100000 images of 200 classes (500 for each class) downsized to 64×64 colored images. Each class has 500 training images, 50 validation images, and 50 test images.

The following is an example of some of the images in Tiny ImageNet:



4.2. Evaluation metrics

To evaluate the effectiveness of our model in comparison to others we could not primarily rely on only error in comparison to the ground truth because a good colorization of an image could have some objects as many different kinds of colors which would be incorrect from that metric. So a better evaluation metric to determine the effectiveness of our model would be human perception. By collecting random images and showing them to a subject after they are put through our model and others to compare we can ask the subject which colorization of an image they find most accurate. By finding out which colorization method subjects chose the most we can find out the most effective method and compare it to ours.

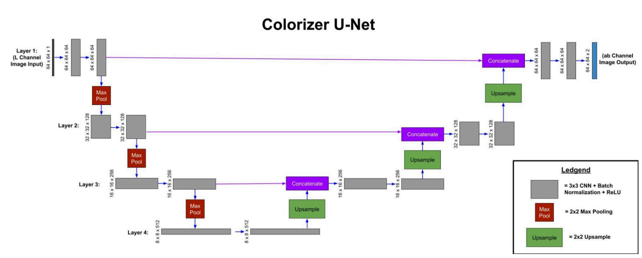


Another method of evaluation could be to show subjects a colorization with our model and the ground truth of an image. Then we would ask the subject what they think the real colored image is and the ideal results would be 50% which would mean the subject cannot tell our colorization apart from the ground truth and it would be considered accurate. Since our model has not produced high-quality accuracy it would fail this for the most part so we stuck with the other evaluation metric for our testing to directly compare with other models.

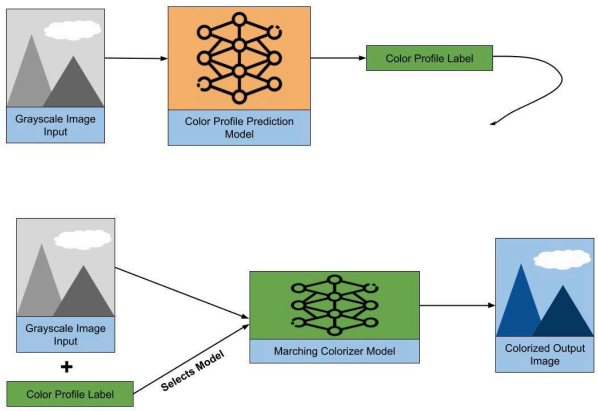
4.3. Implementation details

For this project we are using Keras to create the model. We found that this was the most simple tool to use as it exposes the functionality of machine learning straightforwardly. Naturally, we are using Python for this reason. Code is being run remotely using Google Collab servers to allow for outsourcing computing power.

Our current model is based on a U-Net for image processing. This allows the network to learn low-level features in the encoder and then pass them through a decoder to achieve the final result. It has around a 70% accuracy metric, but the results have been more consistent than our previous models.



- U-net architecture for colorization



- Combined model architecture

We did manage to implement a loss function similar to Zhang et al. that helped to improve the vividness of colors by weighting towards rarer colors. This was done by extracting the a and b values from all our images and creating a histogram with 64 x 64 bins to store colors. We then used this histogram to generate weights emphasizing colors with this formula

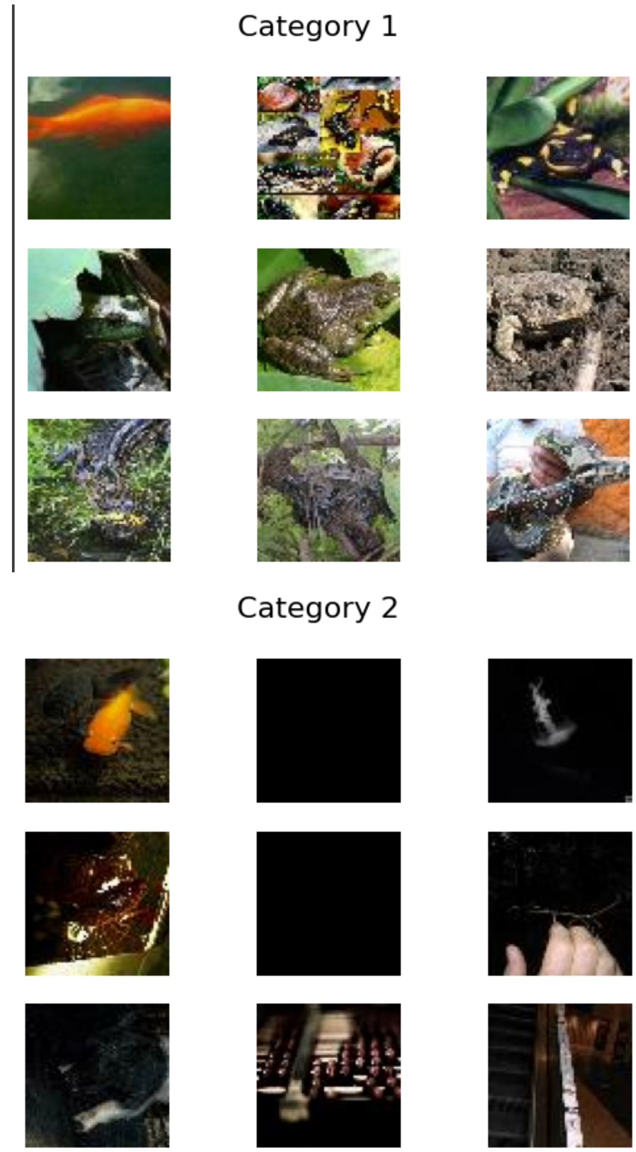
1 / sqrt(a_weights+0.0000001)

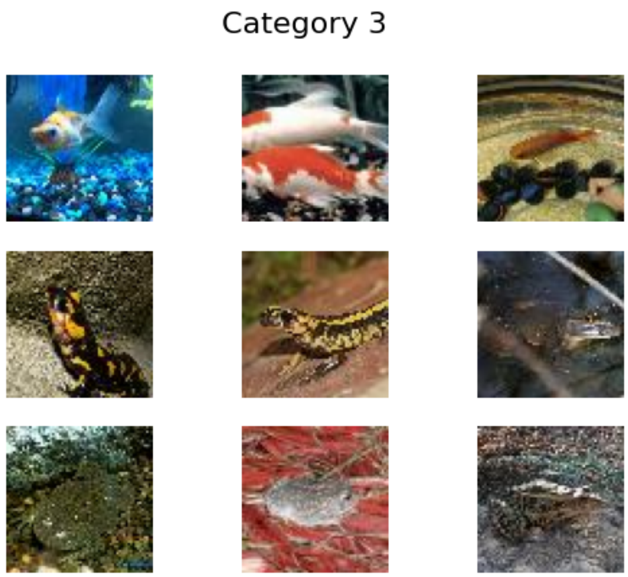
1 / sqrt(b_weights+0.0000001)

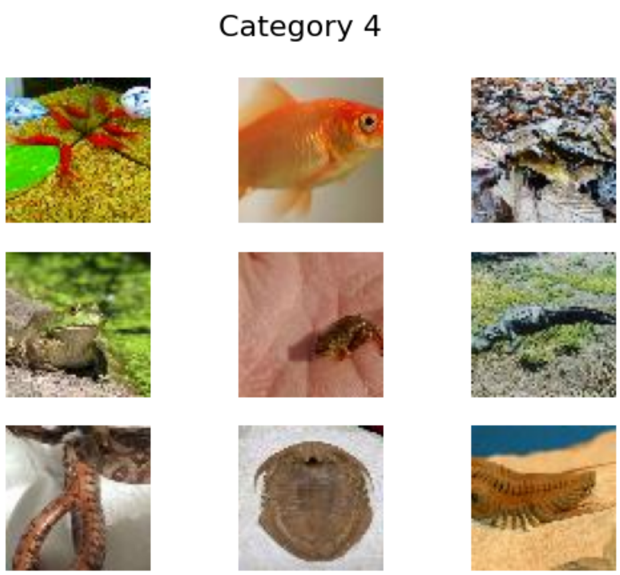
Adding a tiny value to the end of each of the weights to prevent division by 0 errors. We then combined the weights for both a and b together and normalized the whole array to add up to 1, so weights wouldn't be extremely high values.

For the custom loss function, we use a standard Mean Squared Error loss blended at 65% with a custom mean square error that applies the a and b weights to every color within the predicted result from the model.

We also attempted to create color categories using the K-Means algorithm to sort images based on similarities between their colors. We started by sorting the images into 4 categories shown below.





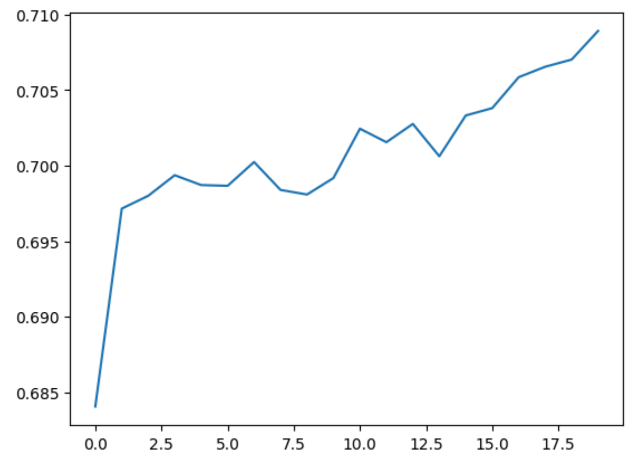


We attempted to train colorizer models on these categories, but ultimately decided not to do that for now as our dataset split up was too small and was causing overfitting issues. However, the steps could be reproduced with a much larger dataset for potentially better results.

The profile predictor model (part one of our multi-model system), could simply be a convolution network with some flattened and dense layers at the end and categorical_crossentropy to match images to their models. We made a simple model like this when testing that worked with 96% accuracy, showing the possibility of color profile prediction from grayscale images.

4.4. Results

Model Accuracy Metric



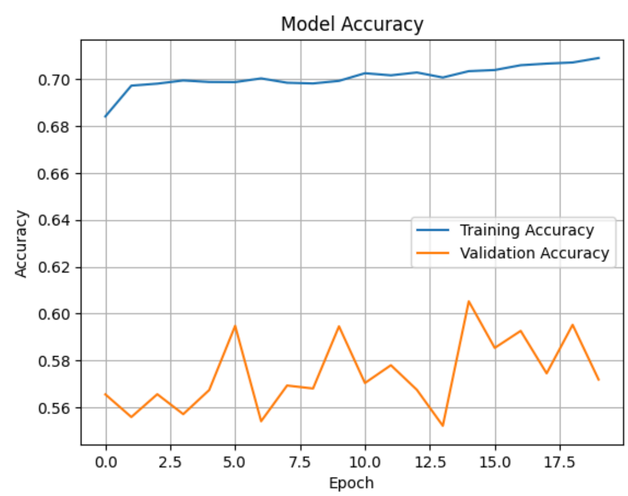
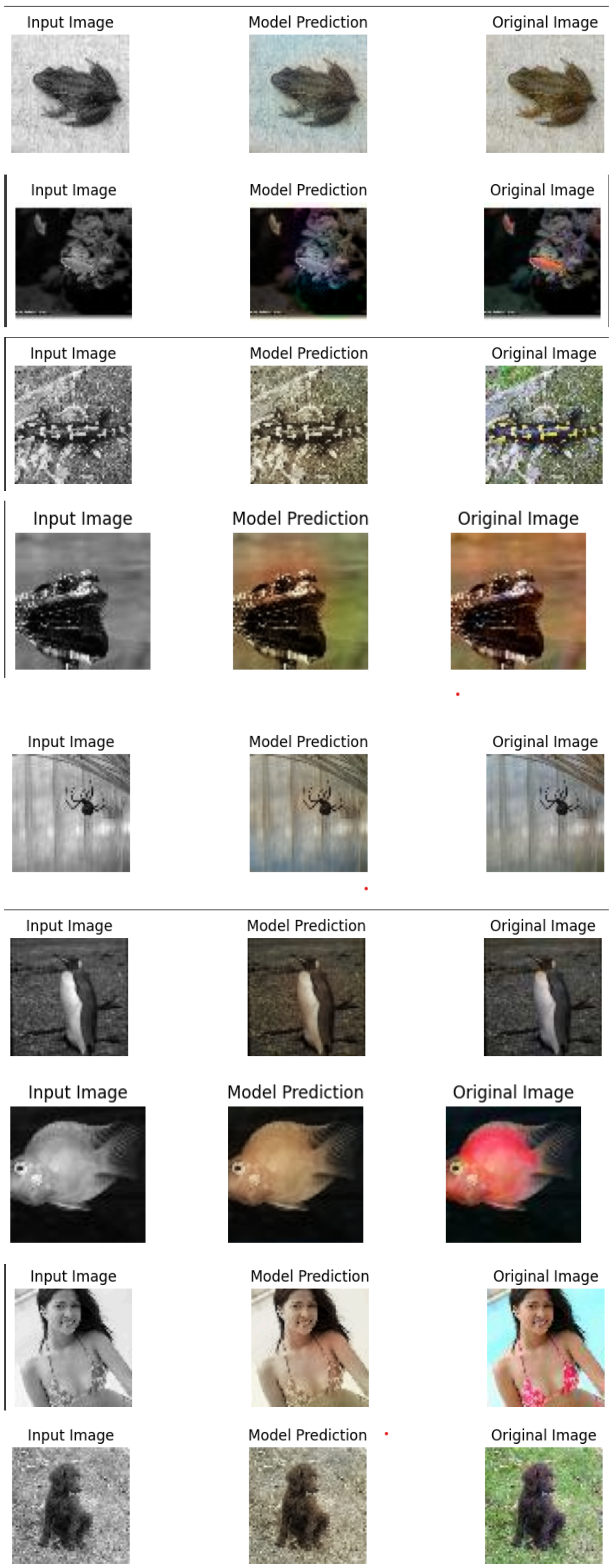
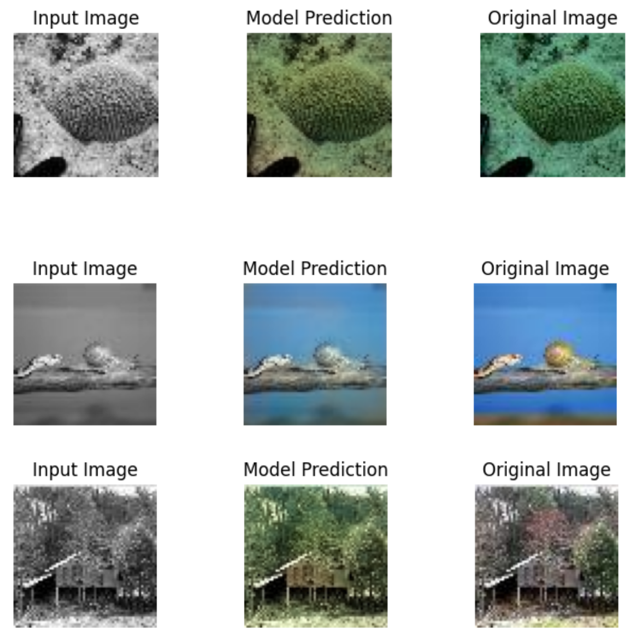
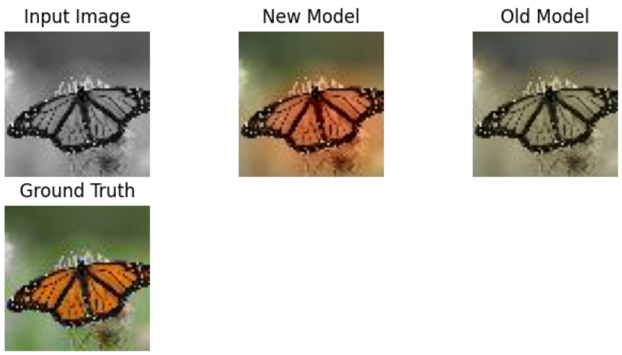


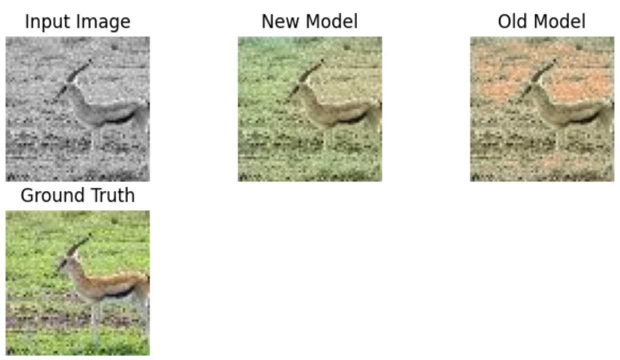
Image Results:

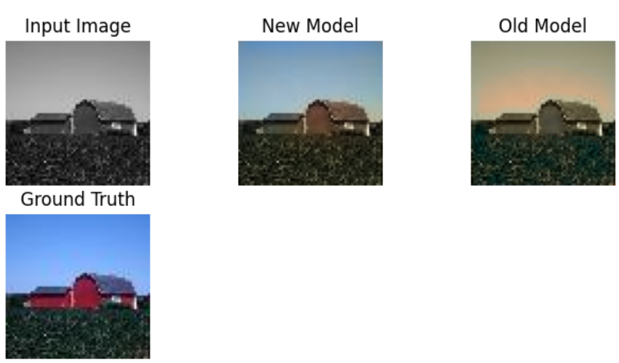


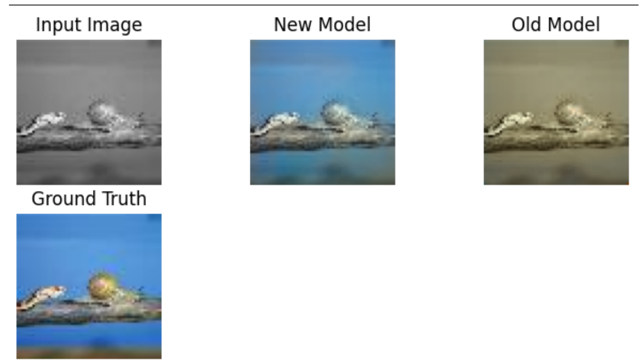


The model overall is a lot better than when we started out. The changing of the architecture to a U-net and the custom color loss function to emphasize rare colors definitely helped the model pick better and more vibrant colors for the images. Below show some comparisons between the new model with the updated architecture and custom loss function and our old model without any of those things. It shows a large improvement in terms of vibrancy and color variety.





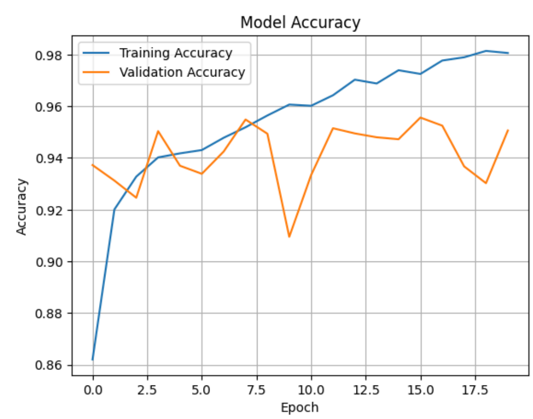




Despite our improvements, when compared to Zhang and others models, ours is clearly still lacking. However the results show promise if we were to get a much larger dataset and fully implement the color profile system we initially planned.

In terms of color prediction and profiling, we do have some partial results on that front. One is the splitting of images into categories using K-Means which appears to be a good way to create the categories without needing manual sorting. Those categories can be seen under the specification section.

We did manage to train a profile predictor model on the profiles that were created that worked with high accuracy. The results are shown below.



In the future this is where we would pick up to continue improving our model.

5. **Conclusion**

We accomplished and learned a lot while working on this project. While our model is not the best, it shows promise if the idea were to be expanded on further.

Some of the things we could do to improve the model are as follows. One big thing would be training the model on a much larger dataset of 256x256 images instead of 64 x 64. This would make the model much more versatile. Also training on a larger dataset in terms of quantity would help with the overfitting issues, make the model more adaptive, and open the possibility for the color profiling. Creating the color profiles would be an experiment to see what quantity works best to keep the amount of profiles low and easy to predict, while also maximizing the differences between them. We tested with 4, but more profiles would likely be optimal in a finished model. Color weights could be generated for each profile and models trained to conform to each.

In the end, we didn't end up creating a model better than the competition. Our model is worse in many ways, partially due to the restrictions on hardware and how often we could use it, restrictions on the size of the dataset we were able to use, and restrictions on the time we had to do this project. But we did lay the groundwork for a lot of possibilities going forward. We learned a lot of things about how image colorization is possible with AI and created a model that works decently well. Besides that, we learned a ton of new things. So all in all I would call this project a success despite the shortcomings and challenges along the way.

References

[1] Shanshan Huang, Xin Jin, Qian Jiang, Li Liu, Deep learning for image colorization: Current and future prospects, Engineering Applications of Artificial Intelligence, Volume 114, 2022, https://doi.org/10.1016/j.engappai.2022.105006.

[2] Zhang, R., Isola, P., Efros, A.A. (2016). Colorful Image Colorization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science(), vol 9907. Springer, Cham. https://doi.org/10.1007/978-3-319-46487-9_40

[3] Dhir, R. Ashok, M. Gite, S., An Overview of Advances in Image Colorization Using Computer Vision and Deep Learning Techniques: Review of Computer Engineering Research, 2020 Vol. 7, No. 2, pp. 86-95, https://doi.org/10.18488/journal.76.2020.72.86.95

[4] Zhang, R., Zhu, J.-Y., Isola, P., Geng, X., Lin, A. S., Yu, T., & Efros, A. A. (2017). Real-time user-guided image colorization with learned Deep Priors. ACM Transactions on Graphics, 36(4), 1–11. https://doi.org/10.1145/3072959.3073703

[5] Iizuka, S., Simo-Serra, E., & Ishikawa, H. (2016). Let there be color! ACM Transactions on Graphics, 35(4), 1–11. https://doi.org/10.1145/2897824.2925974

[6] Liu, H., Fu, Z., Han, J., Shao, L., & Liu, H. (2018). Single satellite imagery simultaneous super-resolution and colorization using multi-task deep neural networks. Journal of Visual Communication and Image Representation, 53, 20–30. https://doi.org/10.1016/j.jvcir.2018.02.016

[7] Xia, Y., Qu, S., & Wan, S. (2018). Scene guided colorization using neural networks. Neural Computing and Applications, 34(13), 11083–11096. https://doi.org/10.1007/s00521-018-3828-z

[8] Verdaguer, Sergi. Color Based Image Classification and Description. October 20th 2009. Color based image classification and description - CORE Reader

[9]Cong, X., Wu, Y., Chen, Q., & Lei, C. (2024, April 8). *Automatic controllable colorization via imagination*. arXiv.org. https://arxiv.org/abs/2404.05661