LaTeX Leveraging Captions for Context-Aware Image Colorization

Sheena Lai
sheenal@stanford.edu

Haoming Song
song4016@stanford.edu

Abstract

The existing literature on image colorization focuses on optimizing the accuracy of color pixels of an image from solely the input of the grayscale version of the original image. In this paper, we approach the task of image colorization by investigating the effects of the inclusion captions in the input of an image colorization model. Using a simple autoencoder architecture in combination with captions processed as text embeddings (specifically using CLIP embeddings), we found that the incorporation of captions as input was able to significantly improve the performance of our image colorization models compared to a plain image colorization model without captions. We observed that when captions were integrated with the original grayscale input, the autoencoder both showed a lower pixel-wise MSE loss from the ground truth and was qualitatively more bold with including color, while the absence of captions in the input resulted in more conservative guesses for pixel color.

1. Introduction

Image colorization is a well-known deep learning task to automate the process of applying color to a grayscale image. Image colorization is often associated with the practical application of coloring historical black and white images. Furthermore, many applicable black and white images from history include captions with them that can aid in the accuracy of the colorization of input images by providing additional context to the image.

In this paper, we survey the performance of an image colorization model, with and without the inclusion of text embeddings of an associated caption with the grayscale image input, exploring the effectiveness of BERT and CLIP text embeddings. We investigate the impact of captioning on the performance of image colorization models, along with a survey of captioning in symphony with well-performing image colorization techniques, such as using the CIELAB color space to define images during training and U-net architecture for optimizing the semantic analysis of the grayscale input.

2. Related Work

Much of the literature surrounding image colorization optimizes for inference the color of pixels, just from the gray scale context of the image. However, we have not found any significant literature for using captions that might come with an image to help with inferring the image color. This turns the classic task of image colorization from purely computer vision to a multimodal task.

Autoencoders Autoencoders are a popular, effective architecture of image colorization. Singh et al. introduce the strategy of using an autoencoder as an effective way to colorize images[5]. The autoencoder is three-part, consisting of an encoder, bottleneck, and decoder, where the encoder is responsible for recognizing different regions and segmentations of the input image, the bottleneck is responsible for further processing of the downsampled features, and the decoder is responsible for reconstructing a colorized version of the image from the extracted features. Hu et al. follows up on the autoencoder with a simple architecture, consisting mainly of convolutions, normalization, and max pooling layers on the encoder side and convolutions, normalization, and upsampling layers (we use bilinear here) on the decoder side, optimizing performance with a limited training[2].

Text embeddings Text embeddings are an efficient way to represent meaningful semantics within a segment of text. Devlin et al. introduces BERT (Bidirectional Encoder Representations from Transformers), which is a commonly used word embedding to define the semantics of a piece of text[1]. Radford et al. introduces CLIP, contrastive language–image pretraining, which are embeddings that are learned from image-text pairs with a contrastive loss[4].

Image representation Traditionally, images are represented as RBG, which are the three channels that represent the intensities of red, green and blue components. Huang et. al., specify several empirically well-performing methods in the field of image colorization, such as defining images in the CIELAB color space[3]. Instead of breaking down each pixel into red, green, and blue channels, CIELAB color

space deconstructs color into a lightness channel (L) and two other channels (a, b) which represents the warm to cool hues of the color, where a represents the red-green axis of a color and b represents the blue-yellow axis of a color. These channels have been shown to capture the perception of human visualization better than RGB channels.

3. Data

We trained our models on the Flickr30k Images dataset, which contains 31,784 unique images from Flickr. Each image also contains five different captions written by annotators, which totals to over 150,000 unique image-caption pairings.

Because of the immense size of the dataset and limited compute resources and training time, we focus on evaluating our models on subsets of the dataset. During the experimentation phase, we trained and tested our model over 10% of the dataset. After tuning all hyperparameters, we trained our final models over 30% of the dataset. We have split the data into 80% training, 10% validation, and 10% testing sets. The approximate size of each different iteration is given below:

% of dataset used	No. Train	No. Val	No. Test
10%	12713	1589	1589
30%	38139	4768	4767

Figure 1. Dataset Sizes

4. Methods

We seek to analyze the impact of image captioning in the effectiveness of image colorization models by training three different models, each with varying degrees of image captions incorporated during training, and evaluated their effectiveness by comparing their final losses to each other during testing.

As as baseline, we finetuned an image colorization model over just a grayscale image. We compared this baseline to two other models, each trained with the incorporation of a different text embedding derived from the image's captions, Bidirectional Encoder Representations with Transformers (BERT) and Contrastive Language-Image Pre-training (CLIP).

All models will be trained on the exact same preprocessed dataset (Flickr30k). They will be trained by being given an input of the grayscale version of an image, as well as its associated caption (for all models except the baseline). The colorized components of the image are given as the output. They will be evaluated by calculating each pixel-wise Mean Squared Error loss on the model's predicted image against the actual colorized image (we expand on the specific procedures between processing the RGB versus CIELAB color spaces in section 4.1).

We will then compare the MSE loss of the captioned models to the baseline MSE loss of the plain model to determine how much of a difference image captioning makes on image colorization models, along with qualitative observations.

For experimentation purposes, we trained our different fine-tuning strategies on just 10% of the dataset.

4.1. CIELAB and RGB Color Spaces

While finetuning our models, we investigated the effectiveness of the RGB versus the CIELAB color spaces.

For the RGB color space, as input, we calculate the grayscale input image as a weighted average of the three color channels (1 channel). Then, we use the original RGB image as the output (3 channels).

On the other hand, for the CIELAB color space, we convert the original RGB image into the CIELAB channels (L, a, and b)[3]. Since the L channel represents lightness, we use the L channel as the input image (1 channel). Since we are given the L channel, the model just needs to figure out the the a and b channels, so our output is the image's a and b channels (2 channels).

In practice, we have found that predictions within the CIELAB color space gave more evenly colored results compared to the RGB colorspace. In the figure comparison below, these two images are both generated with the BERT caption embeddings included, but the RGB coloring seems to have uneven splotches of green and red among the background region of trees and a house. On the other hand, the CIELAB colorspace output seems to have more contained, controlled coloring, as seen with the green color contained pretty accurately within the shapes of grass.

RGB CIELAB

Figure 2. RGB vs. CIELAB colorspaces

We can extrapolate this to the meanings of each of the channels that each colorspace needs to predict. While for the RBG color space, the brightness of an image is not distinctly separated from the hues, the model has to both maintain the brightness values of the image and figure out the colors to overlay on the pixels within the constraints of the 3 channels that it has to predict. In contrast, for the CIELAB colorspace, the model only needs to predict the hues channels given the brightness channel, because the semantic contribution of brightness is distinct from the hues. Therefore, the CIELAB colorspace was able to consistently able to output less splotchy, more precise color predictions.

After these initial color space experimentations, we settled on using the CIELAB colorspace for further finetuning and survey of our models.

4.2. BERT annd CLIP Text Embeddings

To evaluate the role of semantic understanding in improving colorization, we experimented with two well-known forms of textual embeddings derived from the captions: BERT and CLIP. We leverage embeddings because they are a dense, semantic representation of text.

BERT, embeddings generated from a transformer based language model, which subsequently has strengths in strong contextual understanding of the embedded text[1]. We used the pretrained bert-base-uncased model from the HuggingFace Transformers library to generate sentence-level embeddings. Each caption is tokenized and passed through BERT, from which we extract the final BERT embedding to be used. This embedding is then projected and concatenated as an additional channel to the grayscale image.

The other embedding mode we surveyed was CLIP, which, in contrast, is pretrained to align text and image representations in a shared embedding space[4]. Because CLIP embeddings are trained on contrastive loss between text-image pairs, it is effective at containing multi-modal semantics between image and text. For our models, we used OpenAI's CLIP implementation, generating a text embedding by passing each caption through the CLIP text encoder, and similarly concatenate it with the grayscale image as input to the colorization model.

Unlike BERT, CLIP embeddings are specifically optimized for alignment with visual content, which may give the model more contextual understanding of the connection between the image components and the caption.

4.3. Preprocessing Steps

To properly fit the prompt of image colorization, we have taken each image in the dataset and produced a grayscale version of it. We then associated each grayscale image as an input and associated its colorized version as the expected output. Each image is resized to 256 by 256 pixels.

In our caption models, we derive a text embedding (BERT or CLIP) from the caption and concatenate the embedding to the grayscale image as a second channel.

Finally, because the dataset is structured so that image-caption pairs are grouped together by a common image, we also shuffled the dataset so that we don't just get multiple groups of the same image paired together during training, validation, and testing, and train on a diverse variety of image caption pairings. We generated a random seed and saved it in order to preserve consistency.

4.4. Architecture

Autoencoders are a well-established architecture for the image colorization task, found to be efficient and effective[5]. Autoencoders follow the structure of downsampling from an original input image to extract an efficient, denser representation of the image, a bottleneck to further process and extract meaning from these representations, and then upsample back to an output image of the expected format.

For our base architecture, we used a simplified version of the architecture from the autoencoder architecture that Hu et al. introduced.

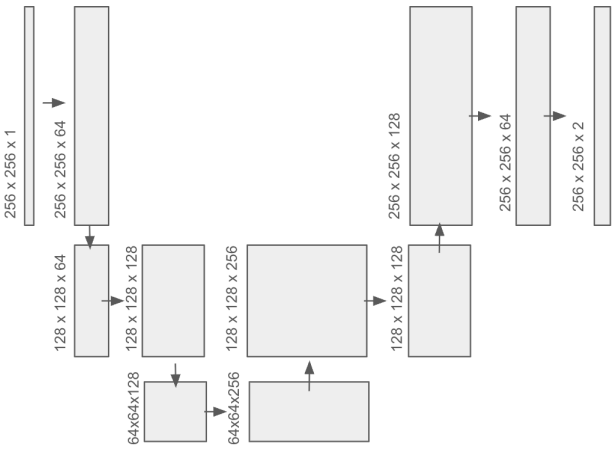


Figure 3. Base Autoencoder Architecture

We use this architecture for our baseline model of just colorizing the image without any caption input. More specifically, each block consists of a convolutional, ReLU, and batch normalization layer. Each downsample is a max

pooling layer and each upsample is a bilinear upsample layer. For our RGB experimentation, we predict the values of the red, green, and blue channels, given the grayscale version, while for our CIELAB experimentation, we predict the values of the a and b channels, given just the L channel.

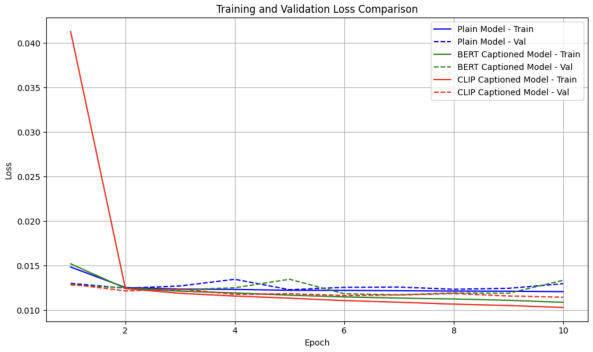
Models are trained over ten epochs, with a batch size of 32, and evaluated against MSE loss with an Adam optimizer.

5. Experiments

To save time, all experiments with the models were done using 10% of Flickr30k image dataset. The final models were testing using 30% of the Flickr30k image dataset. During experimentation, we encountered overfitting issues with both captioned models, and tried a variety of methods to reduce the validation loss for both captioned models.

5.1. Initial Experimentation

We initially began our study by training all three models using 10 epochs, with a learning rate of $\alpha = 10^{-3}$ and no dropout. Below are our training and validation results for all three models:



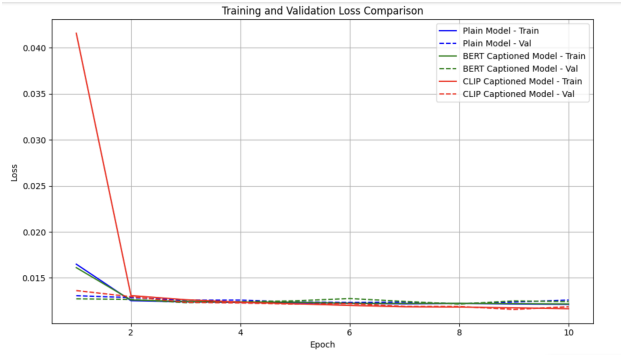
Model	Train Loss	Val Loss	Test Loss
Plain	0.0121	0.0130	0.0130
BERT	0.0109	0.0134	0.0134
CLIP	0.0103	0.0115	0.0115

Figure 4. Initial Results

From the initial results, it seemed like CLIP seemed to outperform both BERT and the baseline model. However, all three models also seemed to show various degrees of overfitting. The overfitting was much more prominent in both of the captioned models, especially BERT. These preliminary results seemed to suggest that adding captions did have an effect on image colorization performance during training, but also made the models more prone to overfitting and as a result demonstrated mediocre final results on the validation and test data.

5.2. Adding Weight Decay

In order to reduce overfitting on the model, we decided to try a variety of methods and finetuned a variety of hyperparameters. The first method we tried to introducing weight decay into the learning rate. After experimenting, we found the best weight decay parameter to be 10^{-5}. The results are shown below:



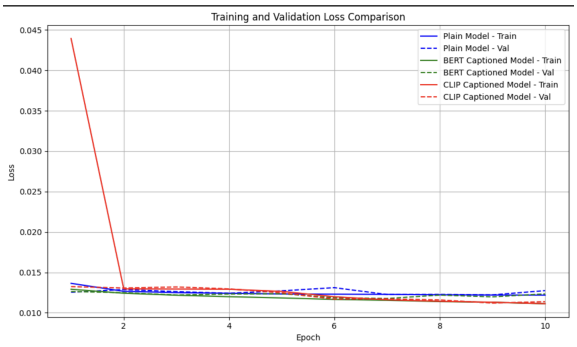
Model	Train Loss	Val Loss	Test Loss
Plain	0.0121	0.0126	0.0126
BERT	0.0122	0.0124	0.0124
CLIP	0.0117	0.0119	0.0120

Figure 5. Results with Weight Decay

From the results, we see that both the plain and the BERT captioned models seemed to perform better with weight decay. The most significant difference is reduction in the difference between the training and validation loss. The BERT captioned model had around a 0.0023 unit difference between training and validation losses, but the difference was reduced to 0.0002 with weight decay. On the other hand, the CLIP model seemed to perform worse on the test data, although the amount of overfitting on the model also seemed to reduce significantly.

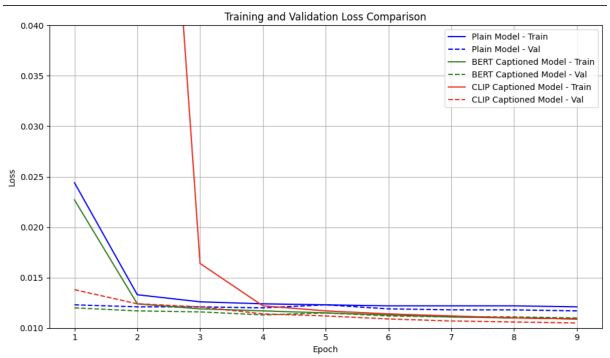
5.3. Dropout

Another method we implemented to reduce overfitting was to add dropout layers. We did this by inserting dropout layers in the decoder blocks of each model, adding a dropout layer after each ReLU layer. After experimenting with finetuning hyperparameters, we found $p = 0.5$ to yield the best results. Attached are the results after applying dropout:



Model	Train Loss	Val Loss	Test Loss
Plain	0.0122	0.0127	0.0127
BERT	0.0111	0.0123	0.0124
CLIP	0.0111	0.0114	0.0116

Figure 6. Results with Dropout



Model	Train Loss	Val Loss	Test Loss
Plain	0.0121	0.0117	0.0118
BERT	0.0109	0.0110	0.0110
CLIP	0.0109	0.0105	0.0106

Figure 7. Final Results

Dropout also seemed to help reduce overfitting for both captioned models. Additionally, dropout seemed to be especially useful for the CLIP captioned model, yielding a record low validation loss value of 0.114. Dropout seemed to be the best addition in terms of both reducing overfitting and maintaining a low validation and test loss score.

5.4. Learning Rate

In addition to introducing parameters, we also finetuned the learning rate of the models to test which one gave the best performance. We ultimately found using a learning rate of $\alpha = 10^{-4}$ yielded the best results.

5.5. Ending Training Early

Besides finetuning model hyperparameters, we also noticed a common issue throughout experimentation: On the tenth epoch, the validation losses for all three models ended up increasing, even though the training losses continued to decrease. This phenomenon suggests that we were training the models for too long, resulting in them overfitting on the training data. To adjust this, we also made the decision to decrease the number of epochs to 9 during training.

6. Final Results

We ultimately applied each of the adjustments that we introduced during experimentation and tripled the dataset size by using 30% of the data from the Flickr30k image dataset instead of 10%. We used a learning rate of $\alpha = 10^{-4}$ with a weight decay value of 10^{-5} and a dropout value of $p = 0.5$, and trained the data over 9 epochs. Our final results are shown below:

From the results above, we see that all three of our final models outperform all of our experimentation results. It is evident that the expansion of the dataset size, in addition to the finetuning of the variety of different hyperparameters such as the learning rate and number of epochs, had a positive impact on the accuracy of all three models. Additionally, we see that the models no longer suffer from overfitting issues, with the validation loss even being sometimes lower than the training loss.

6.1. Text Embedding Performance

The results show that both of the captioned models slightly but noticeably outperformed the original model without captions. The BERT captioned model had around 6.77% less loss compared to the original model, and the CLIP model had around 10.17% less loss. These values suggest that incorporating captions during training during image colorization significantly improves its performance, with CLIP text embeddings performing slightly better than BERT embeddings.

6.2. Image Behavior

To further visualize the impact of captions in image colorization performance, we have displayed a sample image that was evaluated by the baseline model and one of our captioned models, BERT. The caption associated with the image was "Two brown and white dogs racing across a lawn".

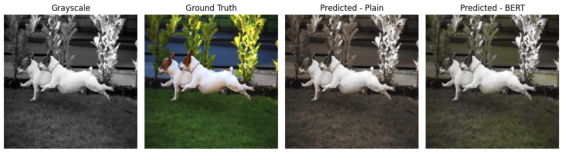


Figure 8. Qualitative Example (Dogs in Grass)

From the visualization, we see that the baseline model simply attempts the shade the image in a different color, while the BERT model not only performs shading on the image, but also attempts to color certain elements of the image, such as the grass and the plants in the background. However, it fails to capture more subtle elements of the image such as the road and the color of the fur on the dogs' heads.

Below is another image with the sample caption: "A person on a bmx bike, running a course."

Figure 9. Qualitative Example (BMX Bike)

From these results, the baseline model once again resorts to simply shading the image, while the BERT model is able to accurately shade in the color of the background sky. However, it fails to capture other elements, like incorrectly coloring the ground as green and failing to color the clothes the biker is wearing.

Finally, we have one more image with the sample caption: "A brown dog runs in the sand."

Figure 10. Qualitative Example (Dog in Sand)

From these results, we see that the captioned BERT model is able to accurately shade in the color of the background plants. However, model also attempts to color in the dog brown, but does not finish completely coloring it.

From these three sample images, we see that the captioned models are able to color parts of the image, while the baseline model merely shades the entire image a different color. From this perspective, it seems like the image captions influence the models to be more ambitious at "coloring" the image than without them, indicating that captioning contributes contextual information that aids the model to make more confident coloring decisions.

7. Conclusion

After training two different text embeddings that incorporate image captions into the training stage of image colorization models, we can conclude that captions do have a positive effect on the performance of image colorization models, with CLIP slightly outperforming BERT. We see that when visualizing the images produced by the different models, the captioned models attempt to actually color in parts of the image, while the baseline model merely shades the image.

During the experimentation stage, we learned that adding captions to the model during training significantly reduced its training loss, but also resulted in overfitting, initially causing the captioned models to perform the same as or below the baseline model. In solving this issue, we learned that adding weight decay and dropout layers to the models significantly reduced overfitting, as well as ending training early and expanding the the size of the training and testing data. We ended up incorporating all of these adjustments to produce the best results in our final model.

Finally, we also learned about the CIELAB color space metric and how much better it was at measuring color than the RGB color space. The evaluation of models on the CIELAB color space resulted in more even shadings and color partitions on images than using RGB, which prompted us to make a switch in how we evaluated the performance of our models.

7.1. Limitations

While BERT and CLIP were able to allow the model to perform better at image colorization than the baseline model, there were still a number of issues that hindered their performance. The main issue that limited the performance of both models was that many regions of the images where still left uncolored. This absence is likely due to the provided captions summarizing the image well but also not providing enough context into the exact colors of certain elements.

For example, the sample captions of the image of the BMX biker from section 6.2 is only able to give the model general context of the image, allowing the models to paint the color of the background sky blue, but fails to provide other specific details, such as the color of the course in the ground or the color of the biker's clothes. Without those details, there is no way for the model to infer the specific colors of these objects. The performance and improvement of the captioned models largely depend on the quality and specificity of the captions, and general captions are not enough to fully capture the details for the color of each object in the image.

7.2. Next Steps

Our results and experiments have shown that adding captions significantly improve the performance of image colorization models. To further improve the accuracy of these models, there is much more data available in the Flickr30k image dataset. With a stronger GPU, it's possible to give the models more training, validation and testing data to further improve its performance.

One application for extending this project would be to address the limitations mentioned in 7.1, the lack of color-specific captions in the training data. An idea would be to test whether the incorporation of more detailed, color-specific captions would result in a significant performance increase on the models. Doing so would require manually writing out more detailed, color-specific captions for each object in the image, and then train two different versions for each captioned model, one with the current, more general captions, and one with more specific, detailed captions.

References

[1] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171-4186).

[2] Hu, Z., Shkurat, O., & Kasner, M. (2024). Grayscale image colorization method based on U-net network. Int. J. *Image Graph. Signal Process*, 16, 70-82.

[3] Huang, S., Jin, X., Jiang, Q., & Liu, L. (2022). Deep learning for image colorization: Current and future prospects. *Engineering Applications of Artificial Intelligence, 114*, 105006.

[4] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021, July). Learning transferable visual models from natural language supervision. *In International conference on machine learning* (pp. 8748-8763). PmLR.

[5] Singh, V., Arora, D., & Sharma, P. (2022). Image Colorization Using Deep Convolution Autoencoder. *In Proceedings of International Conference on Recent Trends in Computing: ICRTC 2021* (pp. 431-441). Springer Singapore.