Learning to Color from Language

Varun Manjunatha*
University of Maryland, College Park
varunm@cs.umd.edu

Mohit Iyyer*
Allen Institute for Artificial Intelligence
mohiti@allenai.org

Jordan Boyd-Graber
University of Maryland, College Park
jbg@umiacs.umd.edu

Larry Davis
University of Maryland, College Park
lsd@umiacs.umd.edu

1 Introduction

Automatic image colorization [12, 1, 19, 2, 9]—the process of adding color to a greyscale image—is inherently underspecified. Unlike background scenery such as sky or grass, many common foreground objects could plausibly be of any color, such as a person's clothing, a bird's feathers, or the exterior of a car. Interactive colorization seeks human input, usually in the form of clicks or strokes on the image with a selected color, to reduce these ambiguities [20, 3, 13, 8]. In this paper, we introduce the task of colorization from natural language, a previously unexplored source of color specifications.

Many creative use cases for automatic colorization involve images paired with language. For example, comic book artwork is normally first sketched in black-and-white by a penciller; afterwards, a colorist selects a palette that thematically reinforces the written script to produce the final colorized art. Similarly, older black-and-white films are often colorized for modern audiences based on cues from dialogue and narration [18].

Language is a weaker source of supervision for colorization than user clicks. In particular, language gives us no ground-truth information about the colored image (e.g., the exact color of a pixel or region). Given a description like *a blue motorcycle parked next to a fleet of sedans*, an automatic colorization system must first localize the motorcycle within the image before deciding on a context-appropriate shade of blue to color it with. The challenge grows as language becomes more abstract: a red color palette likely suits an artistic rendering of *the boy threw down his toy in a rage* better than it does *the boy lovingly hugged his toy*.

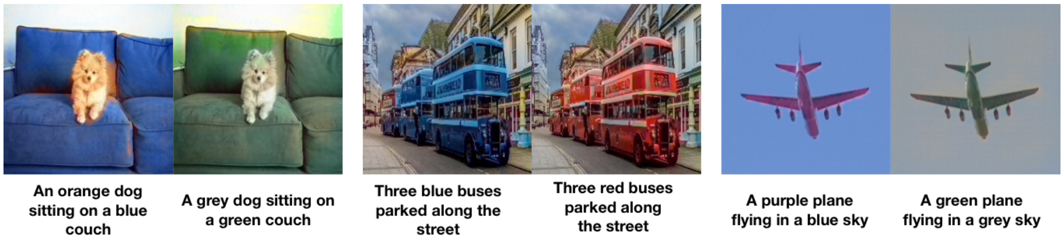

An orange dog sitting on a blue couch
A grey dog sitting on a green couch
Three blue buses parked along the street
Three red buses parked along the street
A purple plane flying in a blue sky
A green plane flying in a grey sky

Figure 1: Three pairs of images whose colorizations are conditioned on corresponding captions by our FILM architecture. These examples demonstrate that our model can localize various objects mentioned by the captions and properly color them.

We present two neural architectures for language-based colorization that augment an existing fully-convolutional model [19] with representations learned from image captions. As a sanity check, both architectures outperform a language-agnostic model on an accuracy-based colorization metric. However, we are more interested in whether modifications to the caption properly manifest themselves

*Authors contributed equally.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

in output colorizations (e.g., switching one color with another); crowdsourced evaluations confirm that our models are indeed able to properly localize and color objects based on captions without reducing colorization quality (Figure 1).

2 Model

In this section, we describe our baseline colorization network and explain how to condition its output on representations learned from language.

2.1 Fully-convolutional networks for colorization

Following Zhang et al. [19], we treat colorization as a classification problem in CIE *Lab* space: given only the lightness channel *L* of an image, a fully-convolutional network predicts values for the two color channels *a* and *b*. For efficiency, we quantize the color channels into a 25×25 grid, resulting in 625 labels for classification; the contribution of each label to the loss is downweighted by a factor inversely proportional to its frequency in the training set, which lessens the impact of desaturated *ab* values. Our baseline network architecture (FCNN) consists of eight convolutional blocks, each of which contains multiple convolutional layers followed by batch normalization [10].[2]

2.2 Colorization conditioned on language

Given an image *I* paired with a unit of text *T*, we first encode *T* into a continuous representation $\boldsymbol{h}$ using the last hidden state of a bi-directional LSTM [6]. We integrate $\boldsymbol{h}$ into every convolutional block of the FCNN, allowing language to influence the computation of all intermediate feature maps. Specifically, denote $\mathbf{Z}_n$ as the feature map computed by the nth convolutional block. A conceptually simple way to incorporate language into this feature map [16, 4] is to concatenate $\boldsymbol{h}$ to the channels at each spatial location i, j in $\mathbf{Z}_n$, forming a new feature map

$$\mathbf{Z}'_{n_{i,j}} = [\mathbf{Z}_{n_{i,j}}; \boldsymbol{h}]. \tag{1}$$

This method (CONCAT) requires considerably more parameters than the FCNN due to the additional language channels. Inspired by recent work on visual question answering, we also experiment with a less parameter-hungry approach, feature-wise linear modulation [15, FILM], to fuse the language and visual representations. Since the activations of FILM layers demonstrate attention-like properties when trained on VQA tasks, we also might expect FILM to be better at localizing objects from language than CONCAT.

The core idea of FILM is to apply a feature-wise affine transformation to the output of each convolutional block, where the transformation weights are conditioned on language. Given $\mathbf{Z}_n$ and $\boldsymbol{h}$, we first compute two vectors $\boldsymbol{\gamma}_n$ and $\boldsymbol{\beta}_n$ through linear projection,

$$\boldsymbol{\gamma}_n = \mathbf{W}_{n_\gamma}\boldsymbol{h} \qquad \boldsymbol{\beta}_n = \mathbf{W}_{n_\beta}\boldsymbol{h}, \tag{2}$$

where $\mathbf{W}_{n_\gamma}$ and $\mathbf{W}_{n_\beta}$ are learned weight matrices. The modulated feature map then becomes

$$\mathbf{Z}'_{n_{i,j}} = (1 + \boldsymbol{\gamma}_n) * \mathbf{Z}_{n_{i,j}} + \boldsymbol{\beta}_n, \tag{3}$$

where $*$ denotes the element-wise product. Compared to CONCAT, FILM is parameter-efficient, requiring just two additional weight matrices per feature map.

3 Experiments

We evaluate FCNN, CONCAT, and FILM using accuracy (a poor substitute for plausibility [19]) and with crowdsourced experiments that ask workers to judge colorization *plausibility*, *quality*, and generalization ability to language *manipulations*. Table 1 summarizes our results; while there is no clear winner between FILM and CONCAT, both of these architectures rely on information present in language to produce higher-quality colorizations than those generated by FCNN.

[2] See Zhang et al. [19] for complete details; we modify their architecture slightly for faster training by using a one-hot encoding for the *ab* channels instead of soft targets.

	ab Accuracy		Human Experiments		
Model	Acc@1	Acc@5	plausibility	quality	manipulation
FCNN	15.4	45.8	20.4	32.6	N/A
CONCAT	**17.9**	**50.3**	39.0	**34.1**	77.4
FILM	16.3	46.8	**40.6**	32.1	**81.2**

Table 1: While all models achieve similar accuracy in *ab* space, **CONCAT** and **FILM** are far more contextually plausible as measured by our *plausibility* task, which asks workers to choose which model's output best depicts a given caption. This additional plausibility does not degrade the output, as shown by our *quality* task, which asks workers to distinguish an automatically-colorized image from a real one. Finally, our caption *manipulation* experiment, in which workers are guided by a caption to select one of three outputs generated with varying color words, shows that modifying the caption has a significant effect on the output images of **CONCAT** and **FILM**.

3.1 Experimental setup

We train all of our models on the 82,783 images in the MSCOCO [14] training set, each of which is paired with five crowdsourced captions. Training from scratch on MSCOCO results in poor quality colorizations due to a combination of not enough data and increased image complexity compared to ImageNet [17]. Thus, for our final models, we initialize all convolutional layers with a **FCNN** pretrained on ImageNet; we fix **FILM**'s convolutional weights during training but finetune **CONCAT** because there are no pretrained weights for the additional language features. To automatically evaluate the models, we compute top-1 and top-5 accuracy in our quantized *ab* output space[3] on the MSCOCO validation set. While the accuracies do not significantly differ across the three architectures, we show next that there are large perceptual differences between the models.

3.2 Human experiments

We run three human evaluations of our models to evaluate their plausibility, overall quality, and how well they are able to condition their output on language.[4] Each evaluation is run using a random subset of 100 caption/image pairs from the MSCOCO validation set[5], and we obtain five judgments per pair.

***Plausibility* given caption:** We show workers a caption along with three images generated by **FCNN**, **CONCAT**, and **FILM**, respectively. They are asked to choose the image that best depicts the caption; if multiple images accurately depict the caption, we ask them to choose the one that looks most realistic. **FCNN** does not receive the caption as input, so it makes sense that its output is only chosen 20% of the time; there is no significant difference between **CONCAT** and **FILM** in plausibility given the caption.

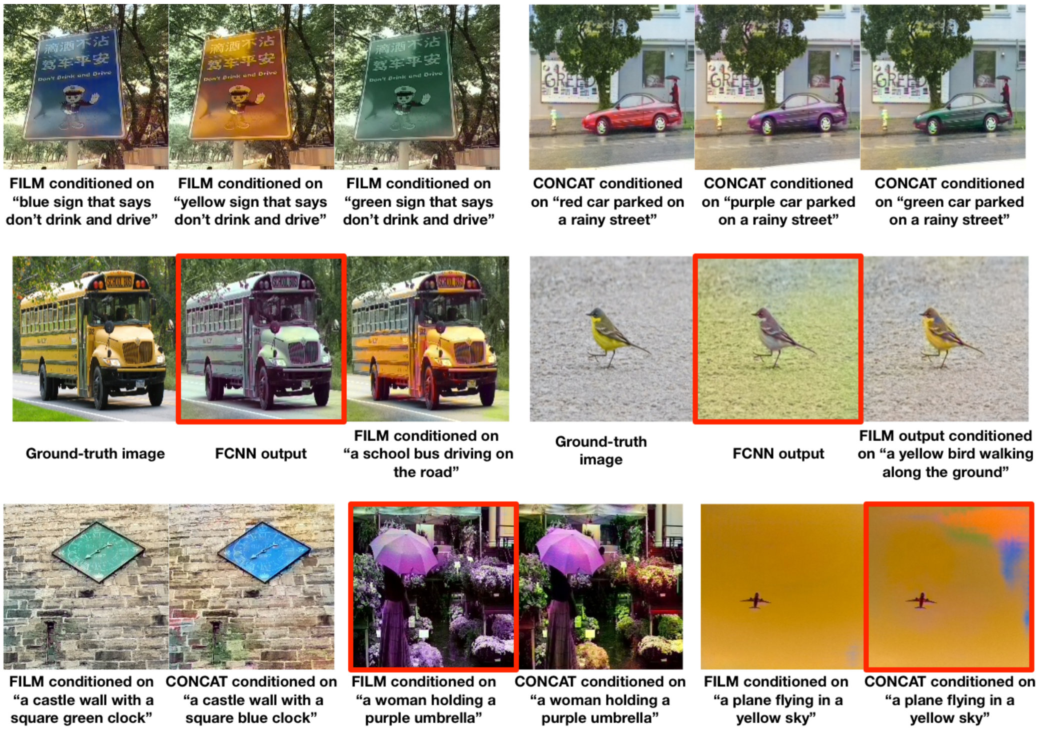
Colorization *quality*: Workers receive a pair of images, a ground-truth MSCOCO image and a generated output from one of our three architectures, and are asked to choose the image that was *not* colored by a computer. The goal here is to fool the workers into selecting the generated images; the "fooling rates" for all three architectures are comparable, which indicates that we do not reduce colorization quality by conditioning on language.

Caption *manipulation*: Our last evaluation measures how much influence the caption actually has on the **CONCAT** and **FILM** models. We generate three different colorizations of a single image by swapping out different colors in the caption (e.g., *blue car*, *red car*, *green car*). Then, we provide workers with a single caption (e.g., *green car*) and ask them to choose which image best depicts the caption. If our models cannot localize and color the appropriate object, workers will be unable to perform this task. Fortunately, Table 1 shows that **CONCAT** and **FILM** are both capable of handling caption manipulations.

[3] We evaluate accuracy at the downsampled 56×56 resolution at which our network predicts colorizations. For human experiments, the prediction is upsampled to 224×224.

[4] Our human evaluations are conducted using the Crowdflower platform.

[5] We only evaluate on captions that contain one of ten "color" words (e.g., red, blue, purple).


FILM conditioned on "blue sign that says don't drink and drive"

FILM conditioned on "yellow sign that says don't drink and drive"

FILM conditioned on "green sign that says don't drink and drive"

CONCAT conditioned on "red car parked on a rainy street"

CONCAT conditioned on "purple car parked on a rainy street"

CONCAT conditioned on "green car parked on a rainy street"

Ground-truth image

FCNN output

FILM conditioned on "a school bus driving on the road"

Ground-truth image

FCNN output

FILM output conditioned on "a yellow bird walking along the ground"

FILM conditioned on "a castle wall with a square green clock"

CONCAT conditioned on "a castle wall with a square blue clock"

FILM conditioned on "a woman holding a purple umbrella"

CONCAT conditioned on "a woman holding a purple umbrella"

FILM conditioned on "a plane flying in a yellow sky"

CONCAT conditioned on "a plane flying in a yellow sky"

Figure 2: The top row contains successes from our caption manipulation task generated by **FILM** and **CONCAT**, respectively. The second row shows examples of how captions guide **FILM** to produce more accurate colorizations than **FCNN** (failure cases outlined in red). The final row contains, from left to right, particularly eye-catching colorizations from both **CONCAT** and **FILM**, a case where **FILM** fails to localize properly, and an image whose unnatural caption causes artifacts in **CONCAT**.

4 Discussion & Future Work

The previous section presents an array of experiments that show both of our language-conditioned architectures are capable of manipulating image color from captions (further supported by the top row of Figure 2). In this section, we qualitatively examine some model outputs and identify potential directions for improvement.

Language-conditioned colorization depends on correspondences between language and color statistics: *stop signs* are always red, and *school buses* are always yellow. While this extra information helps us produce more plausible colorizations compared to language-agnostic models (second row of Figure 2), it biases models trained on natural images against unnatural colorizations. For example, the yellow sky produced by **CONCAT** in the bottom right of Figure 2 contains blue artifacts because skies are usually blue in MSCOCO. Additionally, our models are limited by the lightness channel L of the greyscale image, which prevents dramatic color shifts like black-to-white. Smaller objects are also problematic; often, colors will "leak" into smaller objects from larger ones, as shown by **FILM**'s colorizations of purple plants (Figure 2, bottom-middle) and yellow tires (middle-left).

While the approach we have outlined here is a promising start, it is also clear that there are many avenues to improve language-conditioned colorization. From a vision perspective, we would like to more accurately colorize parts of objects (e.g., a person's shoes); moving towards a more complex architecture such as variational autoencoders [2] or PixelCNNs [5] might help here, as could increasing training image resolution. We also plan to explore the **FILM** architecture more thoroughly and add stronger attentive interactions between the feature map and language representation. On the language side, moving from explicitly specified colors towards more abstract or emotional language is a particularly interesting direction. To this end, we plan to train our models on dialogue/image pairs from datasets such as COMICS [11] and Visual Dialogue [7]; these models could also help learn powerful joint representations of vision and language to improve performance on downstream prediction tasks.

References

[1] Cheng, Z., Yang, Q., and Sheng, B. (2015). Deep colorization. In *International Conference on Computer Vision*.

[2] Deshpande, A., Lu, J., Yeh, M.-C., Chong, M., and Forsyth, D. (2017). Learning diverse image colorization. In *Computer Vision and Pattern Recognition*.

[3] Endo, Y., Iizuka, S., Kanamori, Y., and Mitani, J. (2016). Deepprop: Extracting deep features from a single image for edit propagation. In *Eurographics*.

[4] Feichtenhofer, C., Pinz, A., and Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition. In *Computer Vision and Pattern Recognition*.

[5] Guadarrama, S., Dahl, R., Bieber, D., Norouzi, M., Shlens, J., and Murphy, K. (2017). Pixcolor: Pixel recursive colorization. *arXiv preprint arXiv:1705.07208*.

[6] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*.

[7] Huang, T.-H. K., Ferraro, F., Mostafazadeh, N., Misra, I., Agrawal, A., Devlin, J., Girshick, R., He, X., Kohli, P., Batra, D., et al. (2016). Visual storytelling. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

[8] Huang, Y.-C., Tung, Y.-S., Chen, J.-C., Wang, S.-W., and Wu, J.-L. (2005). An adaptive edge detection based colorization algorithm and its applications. In *Proceedings Annual ACM International Conference on Multimedia*.

[9] Iizuka, S., Simo-Serra, E., and Ishikawa, H. (2016). Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. In *ACM Transactions on Graphics*.

[10] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference of Machine Learning*.

[11] Iyyer, M., Manjunatha, V., Guha, A., Vyas, Y., Boyd-Graber, J., Daumé III, H., and Davis, L. (2017). The amazing mysteries of the gutter: Drawing inferences between panels in comic book narratives. In *Computer Vision and Pattern Recognition*.

[12] Larsson, G., Maire, M., and Shakhnarovich, G. (2016). Learning representations for automatic colorization. In *Proceedings of the European Conference on Computer Vision*.

[13] Levin, A., Lischinski, D., and Weiss, Y. (2004). Colorization using optimization. In *ACM Transactions on Graphics*.

[14] Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*.

[15] Perez, E., Strub, F., de Vries, H., Dumoulin, V., and Courville, A. C. (2017). Film: Visual reasoning with a general conditioning layer. *arXiv*, abs/1709.07871.

[16] Reed, S. E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., and Lee, H. (2016). Learning what and where to draw. In *Advances in Neural Information Processing Systems*.

[17] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*.

[18] Van Camp, J. (1995). The colorization controversy. *The Journal of Value Inquiry*, 29(4).

[19] Zhang, R., Isola, P., and Efros, A. A. (2016). Colorful image colorization. In *Proceedings of the European Conference on Computer Vision*.

[20] Zhang, R., Zhu, J.-Y., Isola, P., Geng, X., Lin, A. S., Yu, T., and Efros, A. A. (2017). Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics*, 9(4).