Automatic Recolorization of Movie Posters (Generative Modeling)

Alex Meistrenko
Student Center for Professional Development
Stanford University
alexmeis@stanford.edu

Abstract

Colorization of images is a broadly motivated and challenging task with application for instance in preparation of images for feature extraction, classification, restoration and colorization of historical images. However, manual colorization and recolorization is a time consuming and costly task, which can be automated via deep learning approaches, including CNNs with encoder-decoder structures. For this task a CNN encoder-decoder architecture is implemented, where the high-level feature extraction for the image content is realized via coupling to an Inception-ResNet-v2 model.

1 Introduction

The motivation for this work arises from my observation of an increasing interest in screening events of old movies. The focus lies on recoloring of movie posters between old and modern color schemes. Refreshed colors could be a promising approach for attracting new audience to screening events of old classics in movie theaters. Following that, the input is a colored RGB image, more specifically a movie poster, which is then transformed to a grey-scaled image and is recolored by a CNN encoder-decoder architecture within the CIELAB color space[1]. The output image is then obtained by transforming back to RGB channels.
The training of the network is studied with respect to two different loss functions. The objective function of the first model is minimized with respect to the L_2 loss between the output pixels and the ground truth pixels per color channel of the CIELAB color space. The objective function of the second model is minimized with respect to the multinomial cross-entropy loss between the output layer with a depth of 394 color classes and the corresponding soft-encoding scheme of the ground truth representation. Here, the final result is obtained by transforming the 394 color classes to the corresponding a, b outputs in CIELAB space and adding them to the L channel, containing the grey-scaled information of the image.

2 Related work

To my knowledge, the specific topic of automatic recolorization of movie posters using an AI approach was not published yet. However, colorization of grey-scaled images, sketches and comics is closely connected to the study of this work and there is a large number of references dealing with image colorization (examples from previous Stanford projects: [1, 2, 3]). In the following I give a

[1] The CIELAB color space is designed to account for visually perceived changes in luminance and human perception of distances between colors.

short overview from the very first approaches to the state of the art models.
Initially, image colorization approaches were based mainly on human intervention for specifying colors in relevant regions [4, 5]. As a first automatic and innovative approach Ref. [6] should be pointed out. Here, the authors introduced an automatic colorization technique, which relies on computing a multimodal probability distribution of all possible colors per pixel and maximizes the probability of the whole colored image at the global level by using machine learning techniques for data extraction. The idea of learning a multimodal color distribution in the context of a deep learning approach with an encoder-docoder CNN architecture was studied in Ref. [7], where the last output volume has a resolution of 64×64 pixels with a depth of 313 color channels. The model was then trained by minimizing the multinomial cross-entropy loss w.r.t. the 313 color classes. This approach drastically improves the colorization effect in comparison to regression like models with L_2 norm as a loss function, which tend to produce rather desaturated colors schemes. In parallel the idea of semantic feature extraction became popular to support the colorization effect by focusing on the semantic composition of the scene and object detection via global-level and mid-level features (see Refs. [8, 9, 10]). This motivated the authors of Ref. [11] to develop an encoder-decoder CNN architecture, where the output of the encoder part is coupled via a fusion layer to a pretrained Inception-ResNet-v2 model, acting as high-level feature extractor for the image content. In the current work, the last model is applied to the recolorization problem of movie posters, both in the original regression mode and extended to multimodal color distributions as described in Ref. [7]. The latter extension is a promising tool to obtain a saturated colorization effect, being crucial for movie posters.

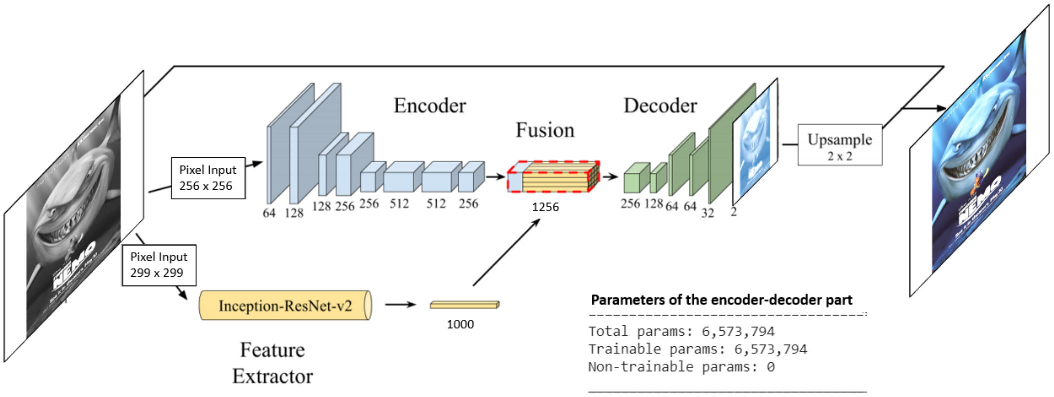
3 Datasets

For this work a high-resolution $\sim 1500 \times 1000$ data set of modern 10.8K movie posters was collected by downloading them mainly from IMDb/OMDb and IMP Awards. This set consists of posters from the time period of $2000 - 2020$. Additionally, I collected a set of 500 older movie posters from the time period of $1940 - 1990$.[2]
The larger 10.8K data set was split in three parts: 9K for training, 1K for validation and 0.8K for test tasks. Furthermore, I also tried data augmentation methods (zooming, flipping and rotating) prior the training time to increase the training set. However, this was rather less effective for the colorization task and I decided to increase the data set of modern movie posters to 40K for training tasks (without validation and test sets). However, this larger set of movie posters was used only for the regression model to check how good it improves with an increasing number of training examples (see Sec. 5). Finally, all trained models were applied to the data set of 500 older movie posters to obtain recolored movie posters (being the motivation for this work).
The input of the network accepts images in a 256×256 resolution. Therefore, all images from the collected data sets are preprocessed by downsampling to this resolution and fixing the original aspect ratio (via centering and adding white padding at the edges).

4 Methods

In this work a Convolutional Neural Network encoder-decoder architecture was re-implemented and extended to include color classification within a Tensorflow/Keras framework. The CNN is coupled via a fusion layer to an Inception-ResNet-v2 model [12], being trained on 1.2 million images from ImageNet and acting as a high-level feature extractor for the image content (see Fig. 1). Compared to the original work [11], in this work the last layer of the Inception-ResNet-v2 model with softmax activation is included in the fusion part, resulting in a depth of $1 \times 1 \times 1000$ for the inception embedding, which is replicated and added via the fusion layer to the depth axis of the last feature volume from the encoder part (see Ref. [11] for more details). Following the fusion layer 256 convolutional kernels with size 1×1 are applied to merge the feature information in a $32 \times 32 \times 256$ volume. This coupled model was studied with implementations in Refs. [11, 13] for colorization of grey-scaled images.

[2] The proof of concept and several test runs were done on a data set from unsplash photos (9K for training, 0.8K for validation and test tasks), see Fig. 5 in appendix.



```
Parameters of the encoder-decoder part
------------------------------------
Total params: 6,573,794
Trainable params: 6,573,794
Non-trainable params: 0
____________________________________
```

Figure 1: CNN encoder-decoder architecture coupled via a fusion layer to an Inception-ResNet-v2 model with the number of trainable parameters and application to movie posters, (modified figure from Ref. [11]).

Optimization with regression loss

For the first model (denoted as regression model), the learning procedure relies on minimizing the L_2 norm between the output pixels and the ground truth pixels per color channel of the CIELAB color space. Let $X \in \mathbb{R}^{H\times W\times 3}$ denote the ground truth image and $\hat{X}$ its network approximation. For $k \in a, b$ being the color channel of the CIELAB color space, the L_2 loss function per image becomes:

$$L_{2,ab}\left(X,\hat{X}\right) = \frac{1}{2HW}\sum_{k\in\{a,b\}}\sum_{h=1}^{H}\sum_{w=1}^{W}\left(X_{h,w,k}-\hat{X}_{h,w,k}\right)^2 \ , \tag{1}$$

where h, w denote the pixel positions in height and width.

In total two regression models were trained, one on the movie poster set with 9K training images and the second one on the movie poster set with 40K training images. In both cases the training was stopped after 180 epochs (see Fig. 4 in appendix for the training loss). With a Tesla V100 graphics card the algorithm takes 1 hour per 10 epochs in case of 9K training images and a batch size of 50 images.

Optimization with classification loss

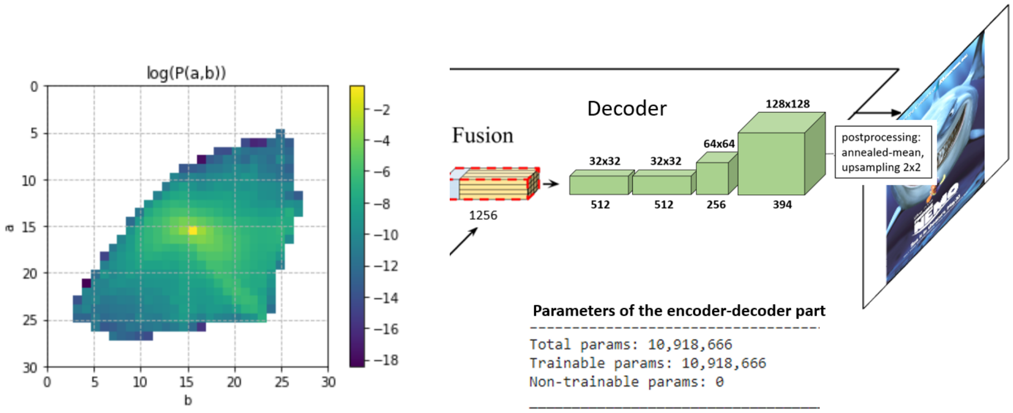
For the second model (denoted as classification model), the learning procedure relies on minimizing the multinomial cross-entropy loss between the predicted and the ground truth color class per pixel. Therefore, prior the simulation an initial distribution function of color classes in CIELAB space f_q is computed by discretizing the space of possible a, b values in 32×32 grid cells (ranging from -128 to 128). Consequently, each grid cell corresponds to 8×8 combinations of a, b values. The class distribution function of different a, b values is then calculated over the full set of movie posters. From Fig. 2 (left side) one observes that only $Q = 394$ classes are populated and only those classes are taken into account. Now, the model is required to learn an output of the form $\hat{Z} \in \mathbb{R}^{H\times W\times Q}$, being the color class representation of a certain pixel at position h, w. Let $Z_{h,w,q}$ denote the ground truth representation of color classes, which is given by converting the ground truth a, b colors via a soft-encoding scheme over the 5-nearest neighbors and weighting them w.r.t. their distances from the ground truth a, b values using a Gaussian kernel (s. Ref. [7]). The cross-entropy loss is then given by:

$$L_{CE\,,q}\left(Z,\hat{Z}\right) = -\sum_{h,w} v\left(Z_{h,w}\right)\sum_{q} Z_{h,w,q}\log\hat{Z}_{h,w,q} \ , \tag{2}$$

where $v\left(Z_{h,w}\right)$ accounts for the class imbalance problem by reweighting the loss w.r.t. rare pixel colors. This imbalance arises in the presence of large background areas with more or less the same color. In analogy to Ref. [7] the reweighting factor is given by:

$$v\left(Z_{h,w}\right) := w_{q^*} \ , \ q^* = \arg\max_q Z_{h,w,q} \ , \ w \sim \left(f_q + \frac{1}{Q}\right)^{-1} \ , \ \mathbb{E}\left[w\right] = \sum_q f_q w_q = 1\,. \tag{3}$$

The decoder part from Fig. 1 was replaced by the decoder structure shown in Fig. 2, which outputs a volume $128 \times 128 \times 394$ with the required number of color channels in the last layer and their softmax activation. The final output follows then from a postprocessing step, including the computation of the annealed mean with a temperature dependent softmax distribution over the color channels, reducing so the spatial inconsistencies (see



```
Total params: 10,918,666
Trainable params: 10,918,666
Non-trainable params: 0
```

Figure 2: Left: log-probability distribution of color classes for movie posters w.r.t. a,b channels in CIELAB color space. Right: decoder part of the classification model including the postprocessing step and the total number of parameters (encoder/fusion parts are unchanged).

Ref. [7] for a detailed discussion):

$$\hat{X}_{h,w,(a,b)} = \mathcal{H}(\mathbb{E}\left[f_T\left(Z_{h,w}\right)\right]), \quad f_T(z) := \frac{\exp(\log(z)/T)}{\sum_q \exp(\log(z_q)/T)}, \quad T = 0.38, \tag{4}$$

where $\mathcal{H}$ denotes a mapping from the annealed value over q to a, b channels of a pixel at position h, w. The resulting layers for a, b channels are then upsampled to 256×256 and joined with the input image of the L channel, followed by a conversion to the RGB color space.
The classification model was trained on the movie poster set with 9K training images for 260 epochs. With a Tesla V100 graphics card the algorithm takes 1 hour per 3-4 epochs with a batch size of 20 images. The reduction of the batch size in comparison to the regression model was necessary to account for the higher memory requirement of the larger output volume.

According to my simulations with Adam Optimizer, the best choice for the learning rate with both loss functions was identified in the range between $10^{-6} - 10^{-4}$. Therefore, the simulation starts with the initial value of 10^{-4} for the learning rate, which then decays by a factor of 2 every 10 epochs, when no loss reduction is observed.

5 Experiments and Results

In the following I show and discuss the final results for the models of Sec. 4, trained on data sets of movie posters from Sec. 3.
There is a big issue with defining an objective/reasonable metric for the quality of the outputs. Exact pixel accuracy appears to be a very restrictive metric, especially for a task, which does not require exact color reproduction. Instead of considering exact pixel accuracy, I focus on the averaged $L_{2,ab}$ norm (see Eq. (1)) over the full test set of movie posters (800 images) as well as on a similar color class reproduction. Therefore, I count the number of color classes in recolored images and compare this quantity to the actual number of color classes. Additionally, I consider deviations in color classes from the ground truth, focusing on accuracy values for deviations by at most 0 color classes (including white padding), by at most 1 color class (w/o white padding) and by at most 2 color classes (w/o white padding). The results are shown in Tab. 1. The regression performs slightly better in the sense of $L_{2,ab}$ norm, leading to an averaged pixel color shift of $\sqrt{239.99} \approx 15.5$ units for a, b channels. However, the classification model outperforms the regression model in the number of color classes, being much closer to the ground truth value of 384 color classes in the test set. Furthermore, also the accuracy for 0-class, 1-class and 2-class deviations are higher for the classification model as it was initially expected. Finally, in Fig. 3 I present the recoloring effect on a selection of old movie posters[3]. Both regression and classification models show a decent performance in recoloring the old movie posters. Especially, in case of the classification model the results are very colorful and brighter than the original images. However, the regression model trained on 40K images shows also a colorful and more consistent colorization, being comparable in training time to the classification model.

[3] More results can be found in the appendix, including unsplash photos, modern movie posters and more images of old movie posters.

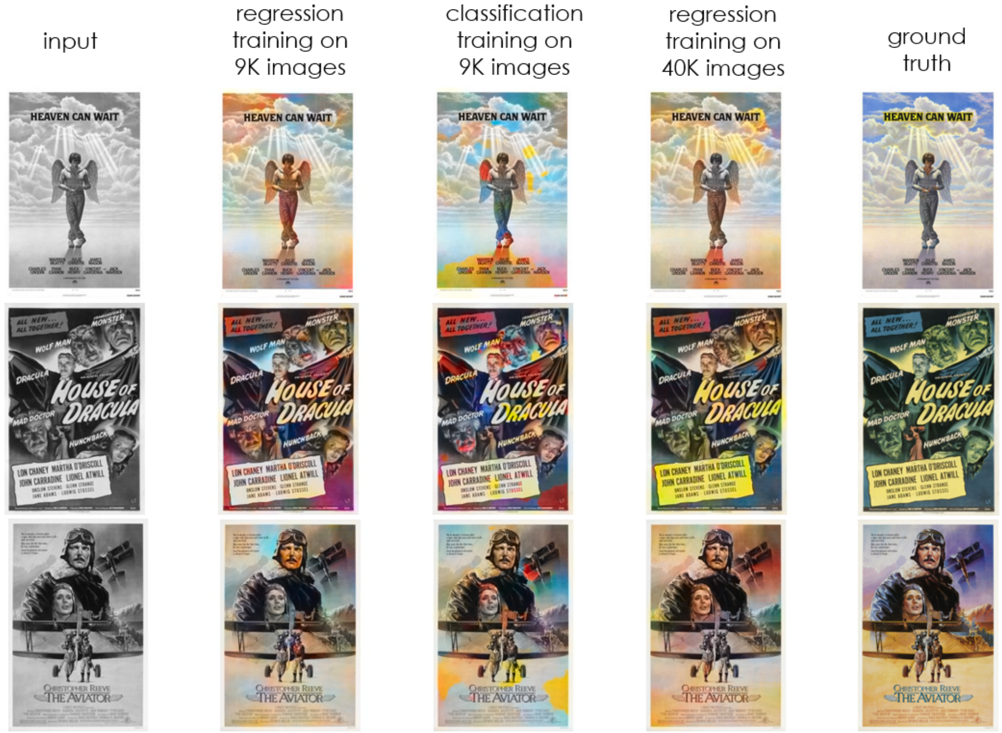

input
regression training on 9K images
classification training on 9K images
regression training on 40K images
ground truth

Figure 3: Comparison between the outputs of different models (regression and classification, see Sec. 4) and the ground truth images.

metric	regression (trained on 9K images)	classification (trained on 9K images)
$L_{2,ab}$ (averaged)	239.99	282.50
number of color classes	250	319
1-class accuracy	21.0%	23.0%
2-class accuracy	42.8%	44.8%
0-class accuracy with white padding	51.5%	52.4%

Table 1: Regression vs. classification model. Shown are averaged values for the $L_{2,ab}$ norm as well as the total number of classes (ground truth value: 386) and different accuracy quantities with deviations of 0, 1 and 2 color classes from the ground truth.

6 Conclusion and Outlook

In this work I applied a CNN encoder-decoder architecture to a recoloring task on old movie posters. I implemented two different models, focusing on a regression like loss and on cross-entropy loss over a multinomial distribution function. The regression as well as the classification models (trained on 9K images) show a decent performance on the recoloring task. Nevertheless, both suffer from graphic artifacts / color splotches, which are more visible for the classification model because of vibrant colors. The regression model trained on 40K images shows, that this effect can be drastically reduced, resulting in much more realistic and consistent images. For a future work, it would be interesting to run the classification model on this larger set of training images, promising most probably colorful and realistic looking images. For this work, the expected very long training times for such a model were out of scope.

More modern approaches for colorization rely on generative adversarial networks (GANs), this would be an interesting direction for the development of the recolorization model, allowing for an artistic style transfer in the same context.

Appendix: Further Results

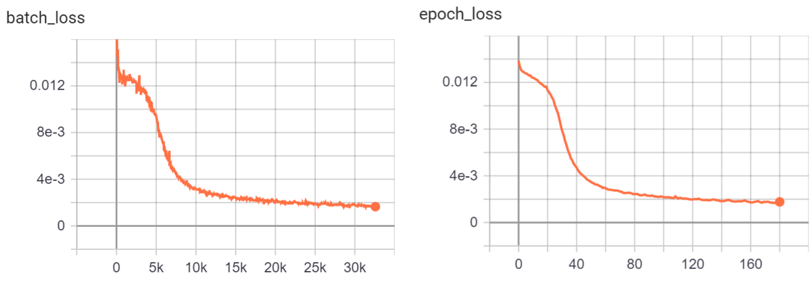


Figure 4: Training loss w.r.t. the number of mini-batches (for a mini-batch size of 50) and w.r.t. the number of epochs (approximately 1 hour per 10 epochs on a Tesla V100 graphics card).

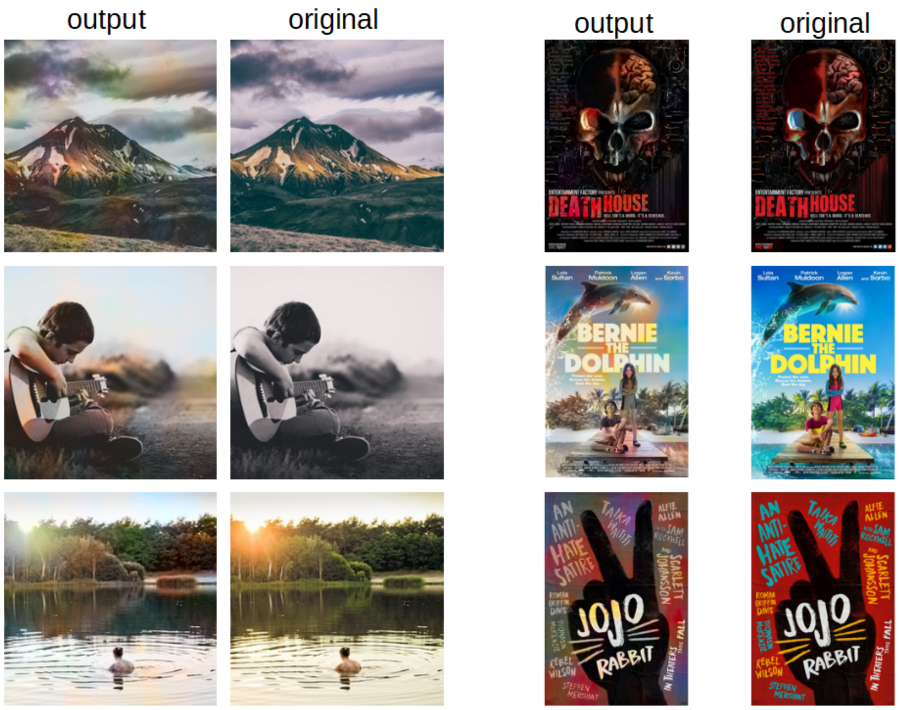


Figure 5: Comparison between ground truth images and corresponding network outputs of the regression model for different data sets (Left: unsplash photos. Right: modern movie posters).

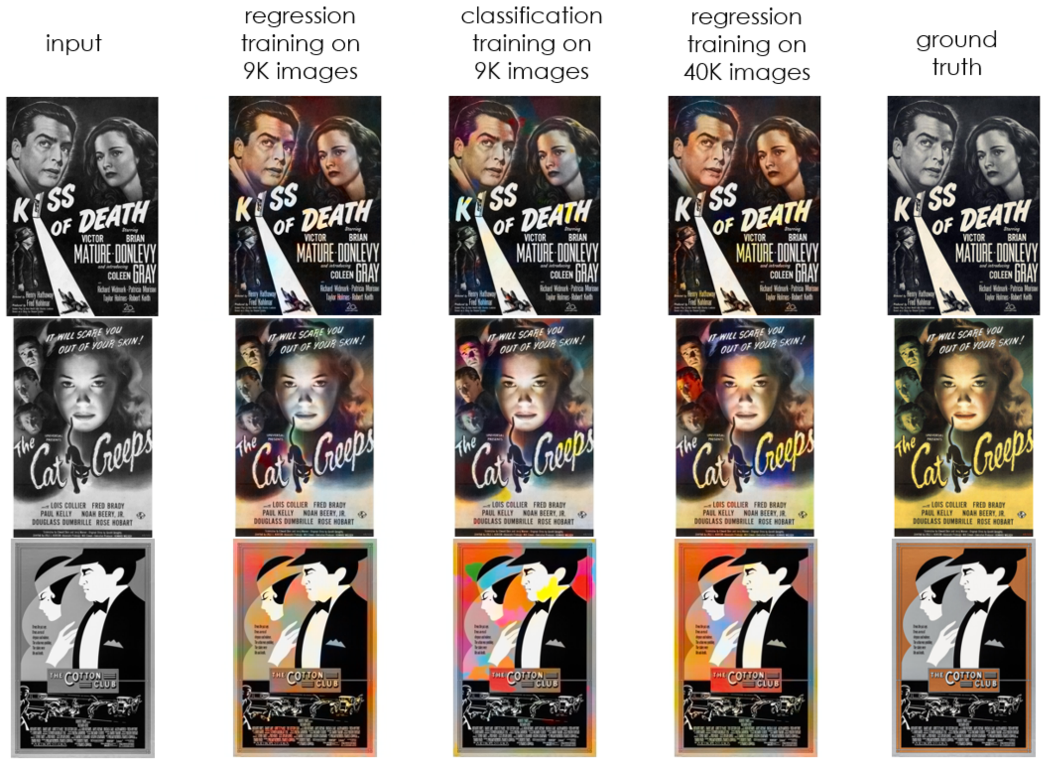

input
regression training on 9K images
classification training on 9K images
regression training on 40K images
ground truth
KISS OF DEATH
VICTOR MATURE
BRIAN DONLEVY
COLEEN GRAY
IT WILL SCARE YOU OUT OF YOUR SKIN!
The Cat Creeps
LOIS COLLIER FRED BRADY PAUL KELLY NOAH BEERY, JR. DOUGLASS DUMBRILLE ROSE HOBART
THE COTTON CLUB

Figure 6: Comparison between the outputs of different models (regression and classification, see Sec. 4) and the ground truth images.

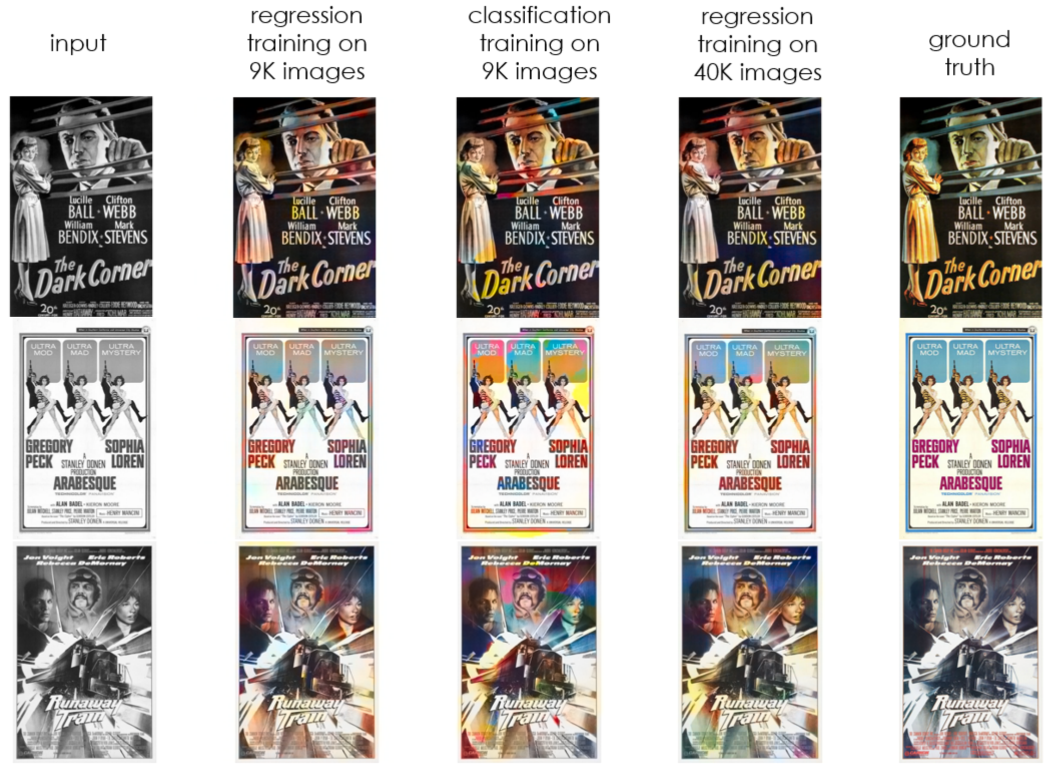


Figure 7: Comparison between the outputs of different models (regression and classification, see Sec. 4) and the ground truth images.

References

[1] Yuki Inoue. CS231n final project: Line drawing colorization. *http://cs231n.stanford.edu/reports/2017/pdfs/425.pdf*, 2017.

[2] Kushaagra Goyal, Bhuvneshwar LNU, and Yash Malviya. CS231n final project: Autocolorization of monochrome images. *http://cs231n.stanford.edu/reports/2017/pdfs/418.pdf*, 2017.

[3] Mahesh Agrawal and Kartik Sawhney. CS231n final project: Exploring convolutional neural networks for automatic image colorization. *http://cs231n.stanford.edu/reports/2017/pdfs/409.pdf*, 2017.

[4] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. *ACM Trans. Graph.*, 23(3):689–694, August 2004.

[5] Yi-Chin Huang, Yi-Shin Tung, Jun-Cheng Chen, Sung-Wen Wang, and Ja-Ling Wu. An adaptive edge detection based colorization algorithm and its applications. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, page 351–354, New York, NY, USA, 2005. Association for Computing Machinery.

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

[7] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. *Computer Vision – ECCV 2016*, pages 649–666, 2016.

[8] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *Computer Vision – ECCV 2016*, pages 577–593, Cham, 2016. Springer International Publishing.

[9] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Trans. Graph.*, 35(4), July 2016.

[10] J. Zhao, L. Liu, , C. G. M. Snoek, J. Han, and L. Shao. Pixel-level semantics guided image colorization. In *British Machine Vision Conference*, 2018.

[11] Lucas Rodes-Guirao Federico Baldassarre, Diego Gonzalez-Morin. Deep-koalarization: Image colorization using cnns and inception-resnet-v2. *ArXiv:1712.0340, https://github.com/baldassarreFe/deep-koalarization*, 2017.

[12] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

[13] Emil Wallner. *https://github.com/emilwallner/Coloring-greyscale-images*.