Colorization as a Segmentation problem and Multitask Learning

Salil Dabholkar
UBID: 50321748
University at Buffalo
salilsan@buffalo.edu

Abstract

Colorization is a popular image-to-image translation problem. Given a grayscale photograph as input, the aim is to hallucinate a plausible color version of it. Image colorization assigns a color to each pixel of a target grayscale image. This problem is clearly under-constrained, so previous approaches have either relied on significant user interaction or resulted in desaturated colorization.

Segmentation is a computer vision task involving pixel level assignment of class label based on certain predefined categories. Early methods relied on low-level vision cues like color (channel) information for doing a clustering based segmentation. These have been superseded by popular machine (deep) learning algorithms.

There exists an inherent correlation between segmentation and colorization. But as of now, there has been no significant research on how segmentation can be used to help train a network for automatic colorization.

To that end, I had two main purposes in this project:
a) Attempt to formulate colorization in a way similar to how segmentation problems are formulated and attempt to solve it using a already proven (segmentation) architecture.
b) Extend the architecture for multi-task learning, simultaneously learning to segment and colorize a given image.

1 Introduction and Background

1.1 Automatic Colorization

Image colorization assigns a color to each pixel of a target grayscale image. A very difficult, unconstrained image-to-image translation problem which normally requires manual adjustment to achieve artifact-free quality. Previously (in early 2000s), it was done using a lot of human intervention. The traditional approach splits an image into shading and albedo components [1]. Good strategies should have three properties: correct prediction, avoid bad spatial patterns, predict multiple channels.

[2] introduced a general technique for "colorizing" grayscale images by transferring color between a source color image and a destination grayscale image. [3] formalized the premise that neighboring pixels in space-time that have similar intensities should have similar colors and then used optimization to solve it.

Most current approaches make use of the advances in Deep Learning, particularly in CNNs. In [4], the authors describe an automated method for image colorization that learns to colorize from (large number of) examples. Their method exploits a LEARCH framework to train a quadratic objective like a Gaussian random field. The coefficients are conditioned on image features using a random forest. Arguably, the currently most popular method, `pix2pix` [5] makes use of a conditional

adversarial network to not only learn the mapping from input to output image but also a loss function to train this mapping.

The semantics of the scene provides cues for many regions in each image: the grass is typically green, the sky is typically blue. These kinds of semantic priors do not work for everything, but it is certainly possible to model enough of these dependencies in order to produce visually compelling results. This is similar to the work done by [6] and provides inspiration for this project.

1.2 Segmentation

Segmentation assigns a class to each pixel of a target image. A relatively simpler and much more constrained problem than colorization. Before the arrival of deep networks, the best methods mostly relied on hand engineered features classifying pixels independently, typically using Random Forest [7] to predict the class probabilities of the center pixel.

Most current methods use an encoder-decoder architecture. The encoder network is something like the VGG16 classification network. The encoder weights are typically pre-trained on the large ImageNet dataset. The decoder network varies and is responsible for producing multi-dimensional features for each pixel for further tasks.

1.2.1 U-Net

CNNs are typically used for classification tasks, where is a single class label. However, in segmentation, the desired output should include localization, i.e., a class label is supposed to be assigned to each pixel. Hence, Ciresan [8] trained a network in a sliding-window setup to predict the class label of each pixel by providing a local patch around the pixel. This was slow, and lacked use of context which motivated the creation of U-Net [9].

U-Net modifies and extends this such that it works with very few training images and yields more precise segmentation. The main idea is to supplement a usual contracting network by successive layers, where pooling operators are replaced by upsampling operators. Hence, these layers increase the resolution of the output. In order to localize, high resolution features from the contracting path are combined with the upsampled output. A successive convolution layer can then learn to assemble a more precise output based on this information. The network does not have any fully connected layers and only uses the valid part of each convolution, i.e., the segmentation map only contains the pixels, for which the full context is available in the input.

1.2.2 MobileNetV2

MobileNetV2 [10] improves the state of the art performance of mobile models on multiple tasks and benchmarks as well as across a spectrum of different model sizes.

The MobileNetV2 architecture is based on an inverted residual structure where the input and output of the residual block are thin bottleneck layers. MobileNetV2 uses lightweight depthwise convolutions to filter features in the intermediate expansion layer. Additionally, it is important to remove non-linearities in the narrow layers to maintain representational power.

1.3 Multi-task learning

Multi-task learning (MTL) has been popular in the machine learning community before the advent of deep learning. This includes designing networks that learn multiple tasks at the same time by exploiting the commonalities and differences between them. We obtain an efficient learner and generalization is improved because we leverage domain-specific information. Because the network uses shared information for all the tasks, what is learnt for one of the tasks benefits all the other tasks. Given correlated tasks that we want the network to learn, multitask networks provides us with better feature representations.

[11] takes an interesting approach in this field. They exploit the class labels of the dataset to more efficiently and discriminatively learn the global priors which are then used to jointly optimize the colorization training.

2 Proposed Work

The work in this project is mainly inspired by work in [6] which showed that colorization can be a powerful pretext task for self-supervised feature learning. The success of [11] wherein they jointly trained classification and colorization, made is seem plausible to do the same with segmentation. I decided to pursue in same direction, using some of the latest advances in deep learning, particularly in the field of segmentation.

2.1 Problem Formulation

I have looked at the colorization problem from the eyes of segmentation. Instead of assigning class values to each pixel, we can use a similar intuition and network architecture to assign color values.

In order to define the problem, I represent images in the YUV color space. In this color space, Y represents luminance while U and V represent chrominance. The input to our algorithm is a rgb image, and the output is the prediction in the YUV space. The loss function is also defined in the YUV space by converting the grayscale image to YUV.

An RGB image is used as input so it is easier to work with a pretrained backbone. YUV is chosen as it has least correlation among the components and allows coloration based on segments. A simple MSE loss is used between the two YUV representations.

Thus, our problem essentially becomes assigning the appropriate YUV values to the given image (instead of a class).

The second proposal requires an additional output of a segmentation mask from the same network.

2.2 Proposed Architecture

The model proposed is a modified U-Net. It works with very few training images and yields precise results. There are a large number of feature channels, which allow the network to propagate context information to higher resolution layers. Successive convolution layers then learn to colorize more precisely due to the combination of high resolution features and upsampled output.

A U-Net consists of an encoder (downsampler) and decoder (upsampler). In-order to learn robust features, and reduce the number of trainable parameters, a pretrained model is used as the encoder, which in my case in the MobileNetV2 model. Intermediate outputs from it are used and the decoder is an upsample block predicting the 3 channel YUV image. The upsample block is simply four blocks consisting of Conv2DTranspose, followed by BatchNormalization and Relu each. Skip connections are made from the MobileNetV2 to the upsample block.

The proposed architecture for colorization can be seen in Figure 4. The input and the output shape is (128, 128, 3) as colorization is a pixel-level task.

The second proposed architecture for multi-task learning is similar to the above with an additional branch for segmentation mask output. This architecture for this multi-task formulation can be seen in Figure 5

Mean-squared error is used as the loss for colorization as it has been shown to work well in recent research, particularly with LAB and YUV spaces. SparseCategoricalCrossentropy loss from tensorflow was used to train the segmentation part in the second network as it requires assignment of only a class / category

3 Results

3.1 Dataset

The dataset used for this project was the Oxford-IIIT Pets dataset [12]. The dataset is a collection of 7,349 images of cats and dogs of 37 different breeds, of which 25 are dogs and 12 are cats. It has images of pets against a variety of backgrounds giving me the opportunity to test my model on broad areas (like grass and snow) as well as minute details on the animals. The format of this dataset

is very similar to the Pascal VOC dataset, but this was much more manageable within my current resources.

3.2 Quantitative

Both the models were trained for 150 epochs. The colorization model achieved a training accuracy of 97.5% and a validation accuracy of 97.23%. Some random images from the test set can be seen in Figure 3.3 These results are impressive as most baseline models considered find it hard to cross the 95% mark. Most of the comparisons in this task use the loss metric for comparison. It is compared with [4] in the table below:

Method	Error
Welsh et al.	0.35
Deshpande, Scene Ind.	0.28
Deshpande, Scene Dep. + Histogram	0.25
Mine	**0.19**

Table 1: Comparison of errors for colorization

The accuracy graph for the colorization model was as follows:

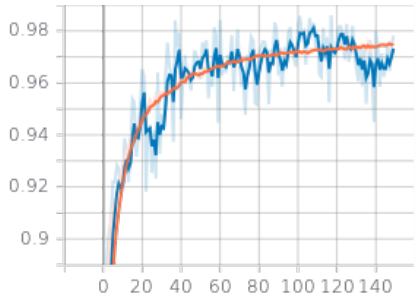

Figure 1: Colorization model training [orange] and validation [blue] accuracy

The second model for outputting a segmentation mask along with a color also worked well. The model trained simultaneously trained with colorization, easily outperformed the baselines in the paper [12]. The results are as compared below:

Method	Error
All foreground	45%
Parkhi et al.	61%
Cats and Dogs	65%
Mine	**74%**

Table 2: Comparison of mIOU for segmentation

The muti-tasking model also performed slightly better at colorization task. It reached the 95% accuracy mark faster (in just 15 epochs vs 25 epochs) and the final accuracy on test data was also slightly better (98% vs 97.23%).

The time taken by the multi-tasking model was obviously more than the single colorization model (due to more parameters and 2 minimization criterion). But it was still much faster than training both models separately.

3.3 Qualitative

The qualitative results provide evidence for the quantitative results. Here are some examples of the colorization outputs:

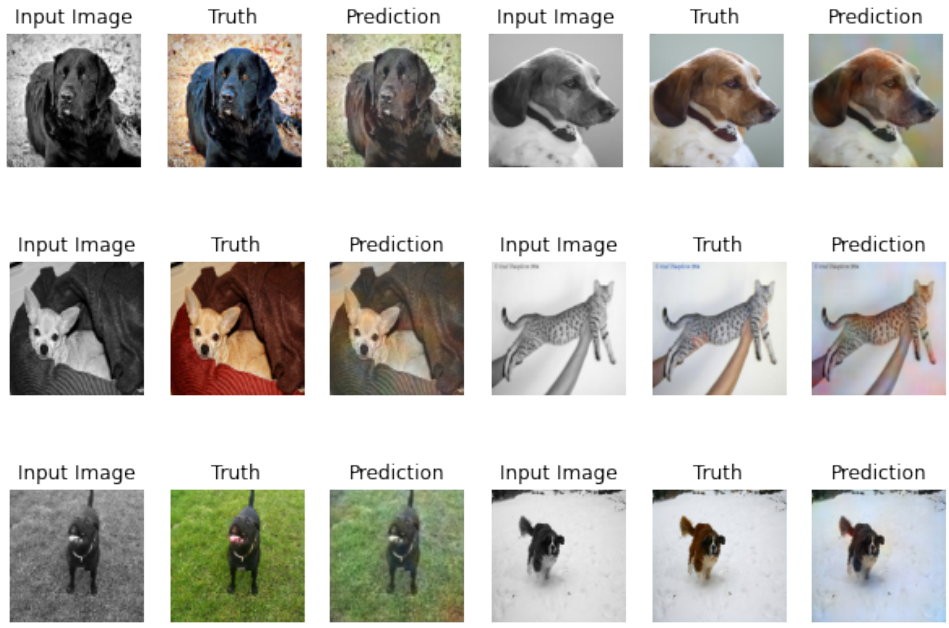

Figure 2: Some random colorization examples from the test dataset

A few things I found interesting in this are its ability to perfectly distinguish grass and snow as can be seen in the last row. The top-left and middle right images also show a (subjectively) much more natural and likely tone. Its ability to hallucinate a more ”scenic” greener background behind the dog in the top right image and deepen the contrast on the dog’s face is also interesting.

The segmentation outputs were also good as expected:

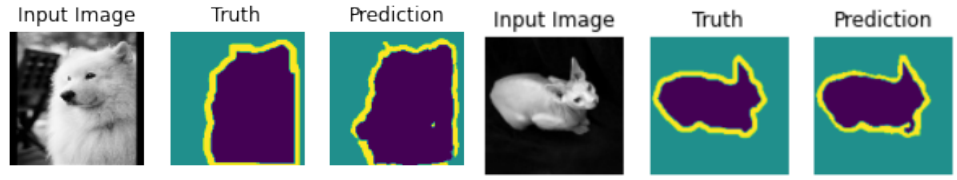

Figure 3: Some random segmentation examples from the test dataset

4 Conclusion and Future Work

This project proved that formulating and solving the automatic colorization problem in a way similar to the segmentation problem is possible and can achieve great results. The success of the second model also proves the correlation between the features used and shared by colorization and segmentation, thus enabling us to achieve excellent results on both the tasks.

The project was carried out on a comparatively smaller dataset due to the limited availability of resources and time. A clear next step would be to thoroughly analyze these models on a large dataset like the PASCAL VOC. Multi-task learning opens up a plethora of other possibilities. One such possibility is to multitask classification along with segmentation and colorization and see what interesting results arise.

References

[1] Edwin H Land and John J McCann. "Lightness and retinex theory". In: *Josa* 61.1 (1971), pp. 1–11.

[2] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. "Transferring color to greyscale images". In: *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. 2002, pp. 277–280.

[3] Anat Levin, Dani Lischinski, and Yair Weiss. "Colorization using optimization". In: *ACM SIGGRAPH 2004 Papers*. 2004, pp. 689–694.

[4] Aditya Deshpande, Jason Rock, and David Forsyth. "Learning large-scale automatic image colorization". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 567–575.

[5] Phillip Isola et al. "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[6] Richard Zhang, Phillip Isola, and Alexei A Efros. "Colorful image colorization". In: *European conference on computer vision*. Springer. 2016, pp. 649–666.

[7] Jamie Shotton, Matthew Johnson, and Roberto Cipolla. "Semantic texton forests for image categorization and segmentation". In: *2008 IEEE conference on computer vision and pattern recognition*. IEEE. 2008, pp. 1–8.

[8] Dan Ciresan et al. "Deep neural networks segment neuronal membranes in electron microscopy images". In: *Advances in neural information processing systems*. 2012, pp. 2843–2851.

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[10] Mark Sandler et al. "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.

[11] 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". In: *ACM Trans. Graph.* 35.4 (July 2016). ISSN: 0730-0301. DOI: 10.1145/2897824.2925974. URL: https://doi.org/10.1145/2897824.2925974.

[12] Omkar M. Parkhi et al. "Cats and Dogs". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2012.

5 Figures

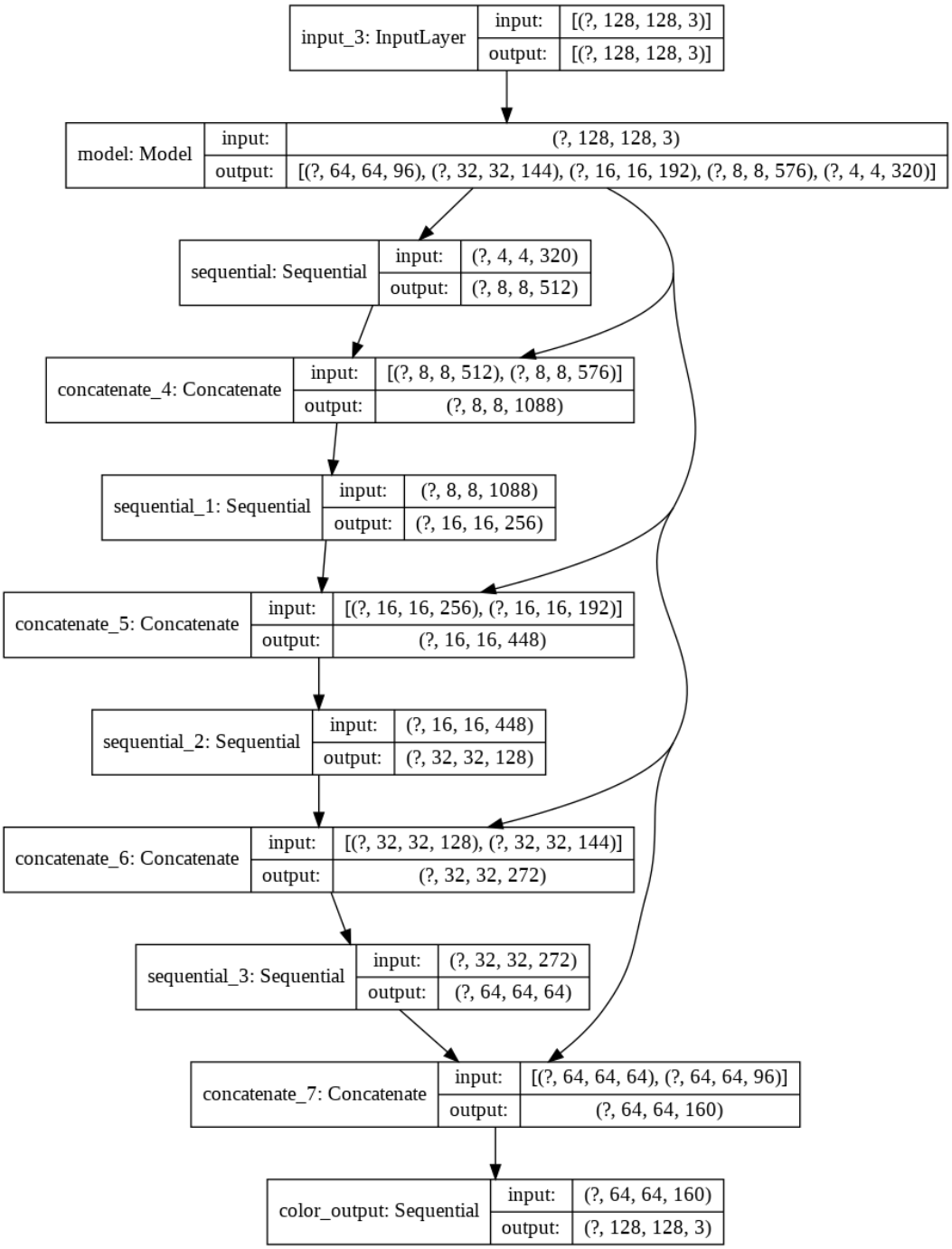

Figure 4: Proposed architecture for automatic colorization

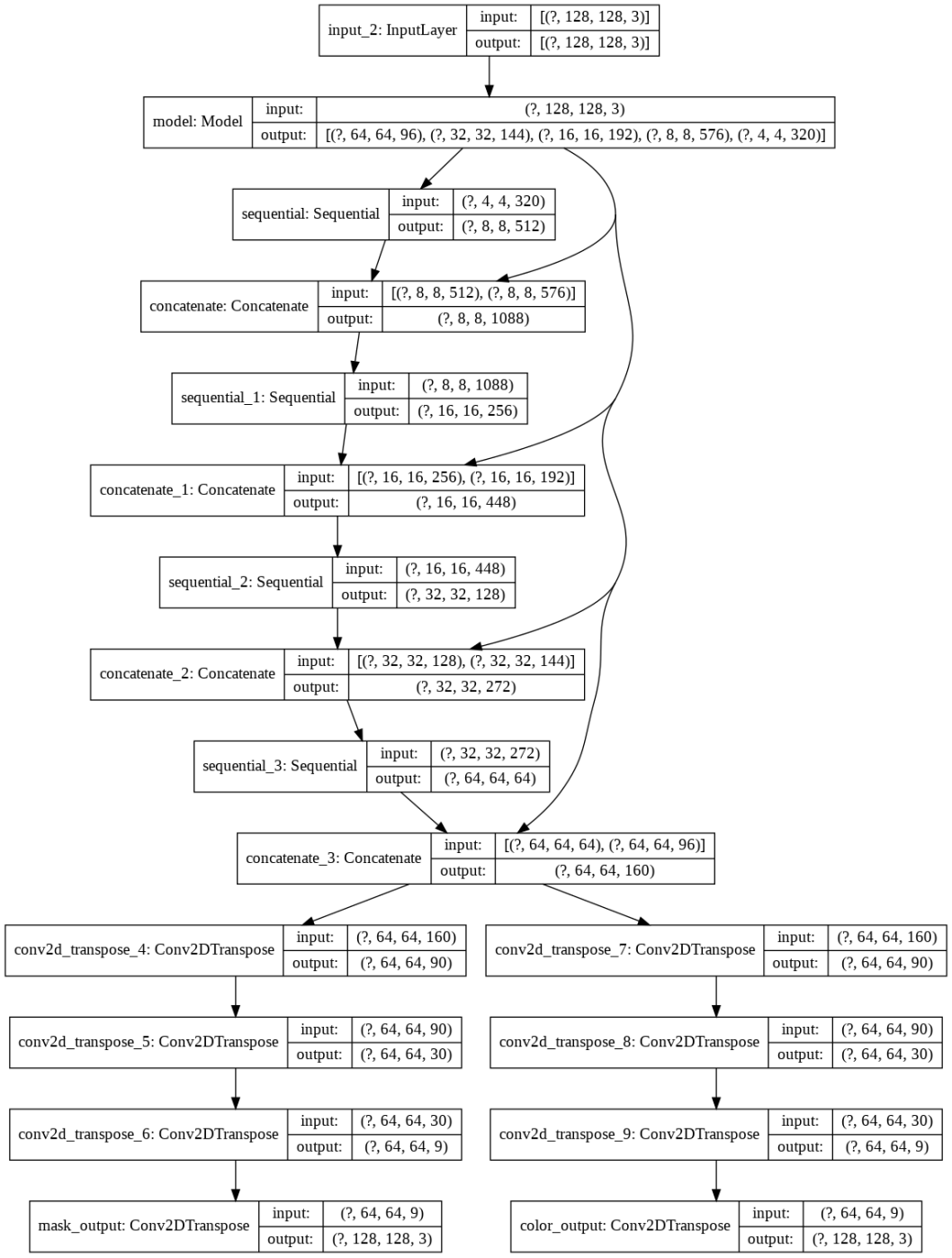

Figure 5: Proposed architecture for multi-tasking colorization and segmentation