Final Project Report

Virtual Reality

Shichao Xu
5140309569, ACM class
Zhiyuan college
Shanghai Jiaotong university
Shanghai, China
Email: 906476903@sjtu.edu.cn

Greyscale image & video colorization

Submission Date:2017/6/6

1. Introduction

Due to the hardware devices limitation, the photos and the movies which have a long history are always in grey scale. We can only know the shape of the object in them, but we cannot imagine what these objects are actually look like. Can we make a retrieve on these grey scale photos/videos? At first glance, so much information has been lost when images are converted to grey, comparing with the information in the images with color. However, with the help with the big training datasets, we can get the color information, surface texture, illumination, ect according to the semantic ways. For example, we can learn that the grass is green, the sky is blue. Then we can recover the information like that.

1.1. Related work

Previous works have solved the grey scale image colorization task using the idea of deep learning as well as some optimization methods. Zezhou Cheng et al. inspired by the success in deep learning techniques which provide amazing modeling of large-scale data [1], and they became the first team who re-formulates the colorization problem so that deep learning techniques can be directly employed. Gustav Larsson et al. develop a fully automatic image colorization system [5], exploiting both low-level and semantic representations to enhance the quality of the automatic colorization tasks result. Aditya Deshpande et al. exploits a LEARCH framework to train a quadratic objective function in the chromaticity maps, and then colorized the grey images. Richard Zhang et al. also use the CNN model, and they even evaluate their algorithm using a colorization Turing test [7].Because they all have use their own evaluation metrics, we cannot give a great comparation for these methods' performance.

1.2. Proposed work in assigned research paper

For the latest paper [Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification] [4], the author use deep Convolutional Neural Networks that have been proven able to learn complex mappings from large amounts of training data to learn the feature of multiple graphs. The whole model consists of four main components: a low-level features network, a mid-level features network, a global features network, and a colorization network. The model will be trained together(end to end learning) in order to achieve high accuracy. The output of the model is the chrominance of the grey scale image which is fused with the luminance to form the output image.

1.3. Limitations of Proposed work

1.3.1. Critical Review. In order to get the information of color and the object class in the images, it has provided a two stream with two objective functions to do the training. However, some obvious deficiencies have block the performance of the original ideas. First, there are too many trainable parameters in the model, so that the time for training well be pretty long. Second, although the author has using the shared weights method to reduce the complicity, the two stream still has a weight balance problem in his framework, which reduce the accuracy of the result. Finally, the total loss definition is too nave, it is just a linear combination. And in this project, we proposed an advanced model based on the original framework. Both the training speed and the accuracy is increased. The most importance is that we extend the model into the grey scale videos, and the temporal information is used to guide the results.

2. Model

2.1. Encoder with convolutional layers

The first step is to encode the initial predictions. Because the initial inputs are one-channel greymaps, in this paper,

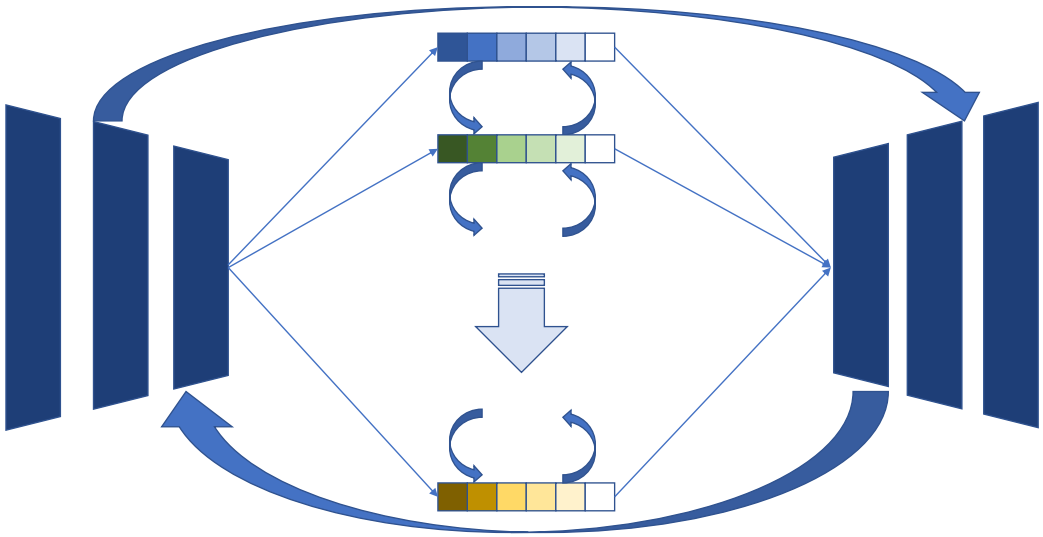

Figure 1. our model

we treat them as images, and utilize several convolutional layers (followed by pooling layers) to extract the feature representations.

As illustrated in Figure 1, instead of mimicking the standard convolutional encoders, we make use of an additional highway layer, which directly resizes the output of second convolutional layer, and concatenates it with the final feature representation. This highway layer can help the encoder to capture both local details and global context.

In formal, the encoder takes the initial inputs $\boldsymbol{p} = (\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_t)$ as input, and gives the feature representations $\boldsymbol{f} = (\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_t)$ as output. Note that, in our implementation, we also stack the initial predictions with the corresponding video frames, which means

$$\boldsymbol{f}_i = \text{encoder}([\boldsymbol{x}_i; \boldsymbol{p}_i]), \quad i \in \{1, 2, \cdots, t\} \tag{1}$$

2.2. Temporal fusion layers

The main purpose of temporal fusion layers is to extract temporal information between continuous video frames, and refine the temporal consistency of feature representations. With the extracted encodings , we would like to refine the predictions from both time directions.

In this paper, as illustrated in Figure 1, we make use of a single layer bi-directional recurrent neural networks with residual connections to generate the refined encodings $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_t)$:

$$\begin{aligned} \boldsymbol{h}_i^L &= \boldsymbol{h}_{i+1}^L + g(W_f^L \boldsymbol{f}_i + W_h^L \boldsymbol{h}_{i+1}^L + b^L) \\ \boldsymbol{h}_i^R &= \boldsymbol{h}_{i-1}^R + g(W_f^R \boldsymbol{f}_i + W_h^R \boldsymbol{h}_{i-1}^R + b^R) \\ \boldsymbol{e}_i &= h(W[\boldsymbol{h}_i^L; \boldsymbol{h}_i^R] + b) \end{aligned} \tag{2}$$

where $\boldsymbol{h}_i^L$ and $\boldsymbol{h}_i^R$ are the hidden states, $g(\cdot)$ is the transition function corresponding to the memory cell, and $h(\cdot)$ is an activation function ($\tanh$ in our implementation). In order to extract long-term relationships and keep higher temporal consistency, in our experiments, we use both GRU [2] and LSTM [3] as the memory cells of recurrent neural networks.

2.3. Decoder with deconvolutional layers

After the refined encodings $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_t)$ have been generated from temporal fusion layers, our final target is to produce the refined dense predictions $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_t)$ using a decoder:

$$\boldsymbol{y}_i = \text{decoder}(\boldsymbol{e}_i), \quad i \in \{1, 2, \cdots, t\} \tag{3}$$

Inspired by [6], in this paper, we choose to utilize several deconvolutional layers as our decoder. Similar to our convolutional encoder, we also make use of several highway layers as illustrated in Figure 1. To be more specific, we insert a single 1×1 convolutional layer to the 4^{th} pooling layer to obtain an additional output, which will be added to the output of the 2^{th} deconvolutional layer. Another highway

layer has also been added in a similar way, but we will not discuss more details here.

As our entire system is totally differentiable, we optimize the whole network in an end-to-end manner using a very straightforward frame-wise pixel-wise cross-entropy objective function.

3. Experiment

Dataset. For the segmentation task, our experiments is based on the DAVIS dataset, which contains 50 high quality video with many difficult challenges like occlusion, out-of-view, deformation and etc. We only use its original images to be the input of the networks.We also add cval1k of ImageNet to be 1-frame videos, in order to increase the varity of the objects.

Training. The encoder part of our network is based on the VGG-16 network. At the start of the training, the original VGG network s weight will be loaded. And in the real experiment, the RNN with LSTM cell will take longer time than RNN with GRU cell to converge. So we choose GRU cell in our experiment . We use the Adam optimizer (has implemented in Tensorflow) with 1e-3 initial learning rate, 0.9 beta1, 0.999 beta2. The batch size of the training is 10. We have also tried other numbers to be the batch size, the small batch size converges slower but may lead to a better result, while the larger batch size will converge faster but takes more GPU memory. This experiment is done on the server with 2 Nvidia K80 GPUs. And the whole network converges for about 12 hours.

Result

We propose our model as a standard benchmark with the following metrics:

RMSE: root mean square error in averaged over all pixels.
PSNR: peak signal-to-noise ratio in RGB calculated per image.

The result of the experiment is shown in Table 1, while part of the result images are shown in figure 2. We can see that the figures we colorized have a very good looking.

But some problems still happened in the results' picture, like the color bleeding, not colored enough, etc. These are coursed by the network not recognize the object properly.

Ablation Study. This Ablation study is to show and quantify the importance of the dynamic layers in our network, Table 1 shows the evaluation of dynamic fusion network compared to ablated versions without the dynamic layers. We try the ablation study on the video dataset. We can see that the dynamic truly pass the useful information and plays an important role. And If we just remove the whole layer, the mIOU will drop at least 1%.

4. Conclusion

As we try to solve the video colorization problem, we show that our model is powerful on improving colorization tasks performance. By extracting the images with channels, the dynamic layer can exchange the useful information to predict the next frame and correct the previous. And this model can be applied to both the video tasks(can improving the performance) and the single image tasks(extend to video).

References

[1] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. 2015.

[2] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In *NIPS Workshop*, 2014.

[3] Sepp Hochreiter and Jurgen Schmiduber. Long Short-Term Memory. *NC*, 1997.

[4] 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 Transactions on Graphics (Proc. of SIGGRAPH 2016)*, 35(4):110:1–110:11, 2016.

[5] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. 2016.

[6] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning Deconvolution Network for Semantic Segmentation. In *ICCV*, 2015.

[7] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

Model\Metric	RMSE	PSNR
No colorization	0.412	23.76
Automatic colorization with K = 32	0.299	24.45
Our model	0.291	23,98

Figure 2. our model

Model\Metric	RMSE	PSNR
No colorization	0.412	23.76
Our model without dynamic layer	0.339	23.89
Our model	0.291	23,98

Figure 3. our model

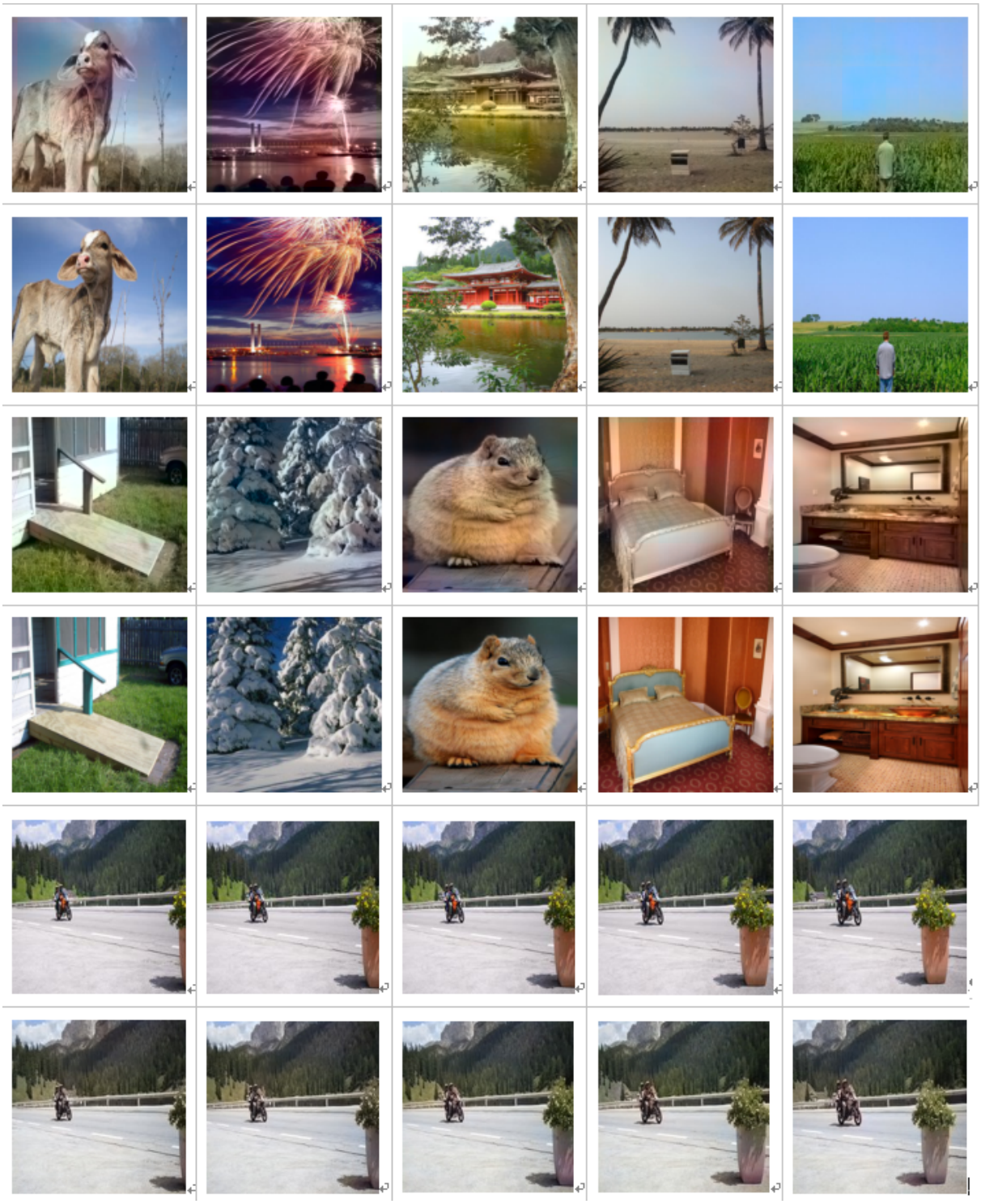

Figure 4. some of the result, where the former is the experiment result and the later is the original image.