Grayscale Images Colorization

Instructor: Prof. Sundeep

Jiahao Chen, Zhenlan Hu, Zhuoya Shi

05/19/2019

Table of content

- Starting point--available frameworks
- Dataset and computer specs
- Test result of framework 1
- Tested parameters for framework 2
- Prediction results for framework 2
- Result comparison and conclusion
- Reference

Starting point--available frameworks

- Framework 1: Convolutional neural network (VGG16)
  - 4 layers, no max pooling
  - Boundary condition: same mode
- Framework 2: Convolutional neural network & ResNet V2
  - Convolution neural network
    - Encoder: 8 convolution layers
    - Fusion: concatenate with ResNet output, 1 convolution layer
    - Decoder: 5 convolution layers, 3 upsampling
  - Pre-trained ResNet

```
#Load weights
inception = InceptionResNetV2(weights='imagenet', include_top=True)
inception.graph = tf.get_default_graph()
```

Dataset and computer specs

- Dataset:
  - Colored image from FlickrAPI (keywords: “portraits”)
  - Single class of images are easier for the model training, but may result in a lack of generalization ability.

```
In [2]: api_key = u'3f2550987c3d0ea5d486835a149e1779'
        api_secret = u'97cda0275103d370'
        flickr = flickrapi.FlickrAPI(api_key, api_secret)
```

```
In [8]: keyword = 'portraits'
        dir_name = 'portraits'
        photos = flickr.walk(text=keyword, tag_mode='all', tags=keyword,extras='url_c',\
                             sort='relevance',per_page=100)
```

- Computer specs:
  - Google Cloud Platform (GPU: 1*Nvidia Tesla P100; 2vCPUs, 13GB Memory)

Test result of framework 1 (dataset size = 8)








Tested parameters for framework 2

- Changed parameters: training dataset size, batch size, epoch

Id	Training dataset size	Batch size	Epoch
1	10	10	1000
2	128	32	1000
3	1024	32	50

Result for group 1 with framework 2



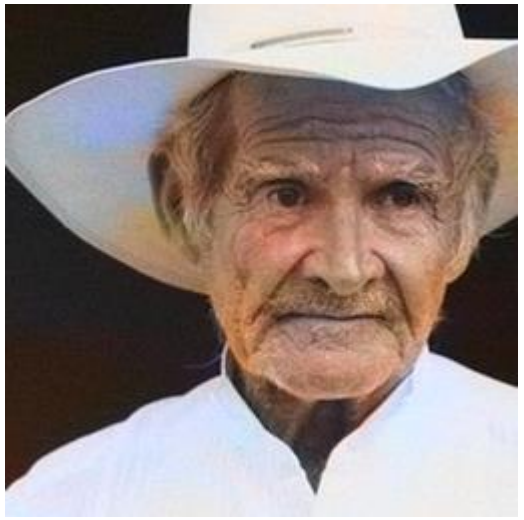





Result for group 2 with framework 2





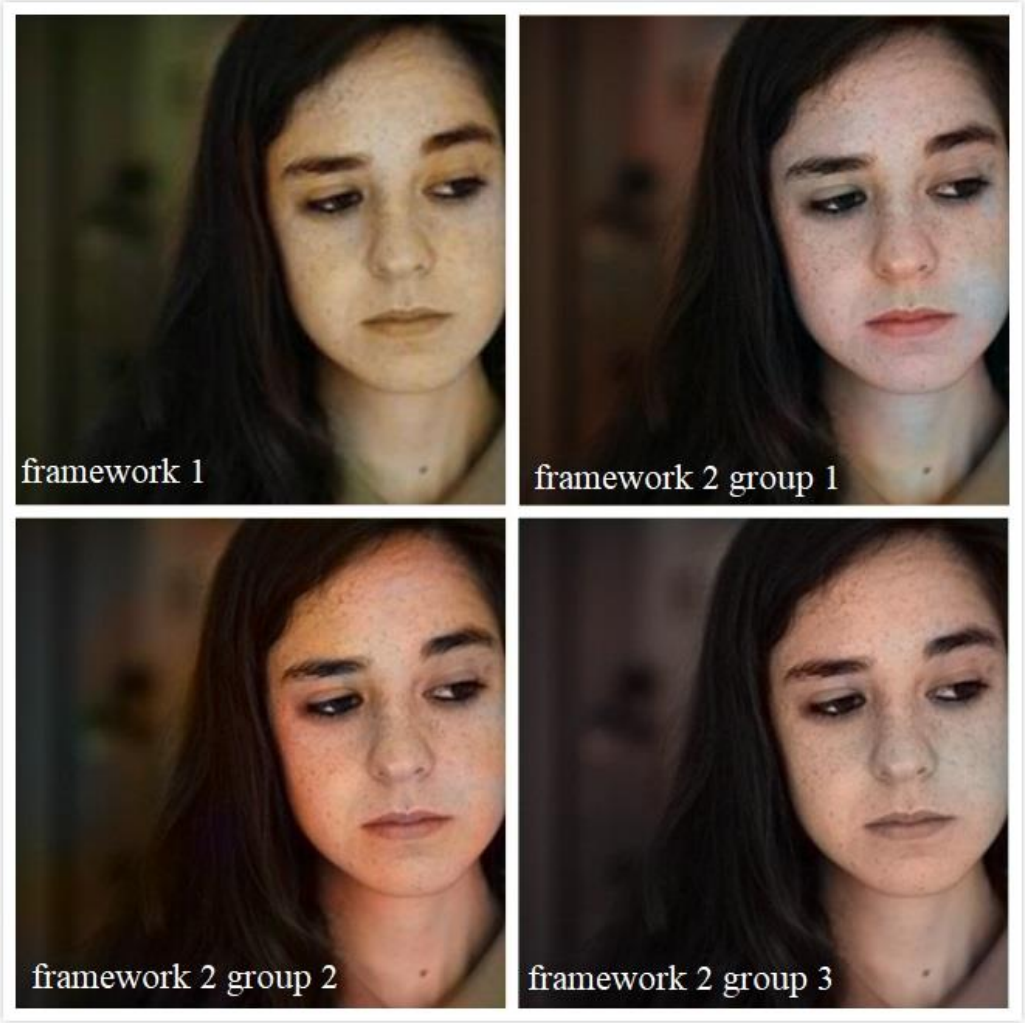
Result for group 3 with framework 2

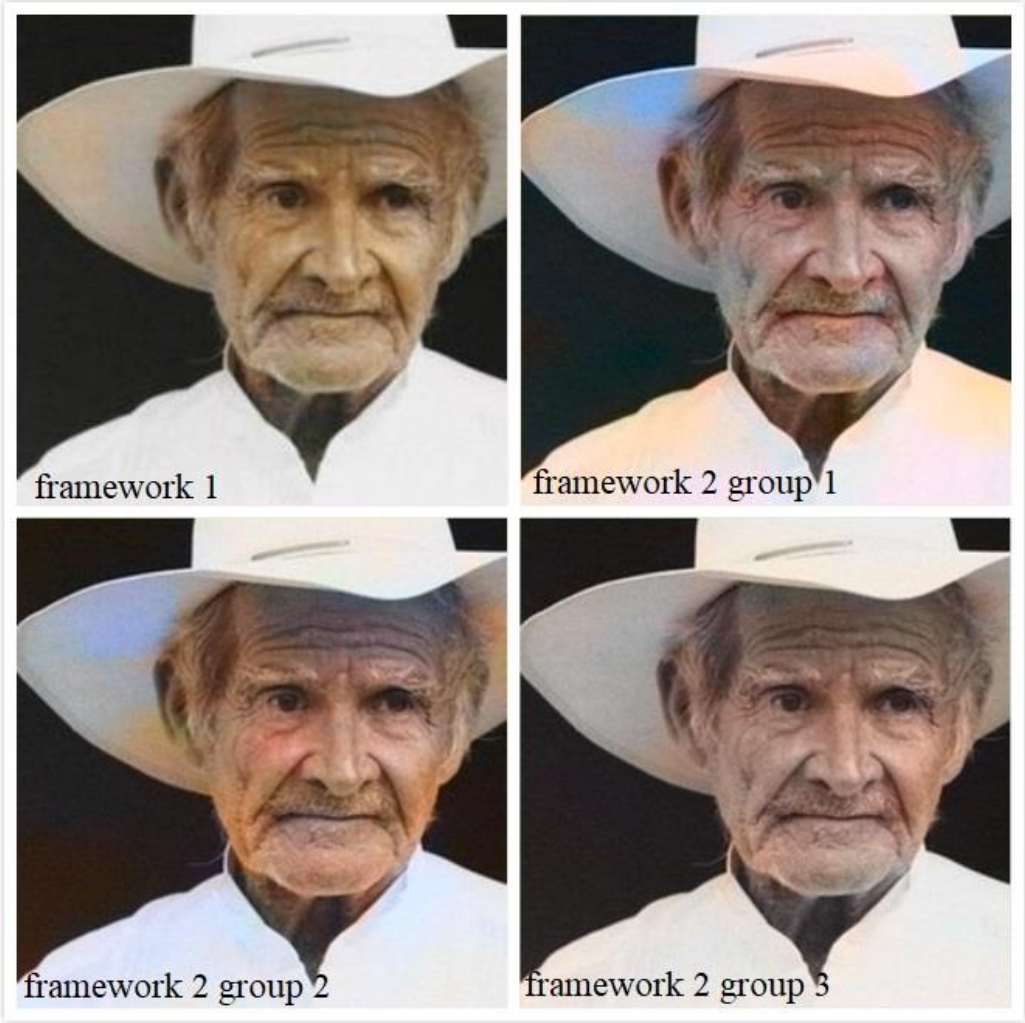







Result comparison





Conclusion

- Framework 1 tends to apply a yellowish tone to the B&W images because the pre-trained VGG 16 structure is not complicated enough (only 4 conv. layers added and no max pooling), may not sufficient for features recognition.
- The group 1 result is mostly unsaturated due to the small training dataset.
- The group 2 result has a relatively correct colorization, but not all the regions are colorized evenly (Eg, for the right side result samples, there are some small blue and green regions on the hat where they should not exist). It may because the batch size has not been properly set, which may influence the process of finding the global optimal.
- The group 3 has better result compared with the previous two because of the larger dataset size and proper batch size. The training epochs for group 3 is only 50 due to the time limitation. We believe if the epoch value increases, the colorization result will be better.

Reference

- Baraldi, L., 2016: VGG 19 model for Keras, created Jan 16, 2019, https://gist.github.com/baraldilorenzo/07d7802847aaad0a35d3
- Baraldi, L., 2016: VGG 16 model for Keras, accessed April 13, 2019, https://gist.github.com/baraldilorenzo/8d096f48a1be4a2d660d
- Shetty, S., 2018: Image Colorization, accessed 12 April 2019, https://www.kaggle.com/shravankumar9892/image-colorization
- Simonyan, K., and Zisserman, A., 2015: Very Deep Convolutional Networks for Large-Scale Image Recognition, Computer Vision and Pattern Recognition, https://arxiv.org/abs/1409.1556
- Zhang, R., et al., 2016: Colorful image Colorization, Computer Vision and Pattern Recognition, https://arxiv.org/abs/1603.08511
- Automatic Colorization, created in January 2016, https://tinyclouds.org/colorize/
- ILSVRC-2014 model (VGG team) with 16 weight layers, https://gist.github.com/ksimonyan/211839e770f7b538e2d8
- Colorizing B&W Photos with Neural Networks, 2017-10-13 https://blog.floydhub.com/colorizing-b-w-photos-with-neural-networks/