Classification and Reproduction of Famous Artists' work

Chih-Ching Chang, Junwei Su

1. Introduction

The research of visual perception of object and face recognition in computer vision area has successful progress recently. Several successful products about face recognition are widely used now. There are algorithms that can generate the content we want based on the feature learned from different objects. On the other hand, besides the content of an image, we could also modify the style of an image. In graphics, non-photorealistic rendering is used to make the image with styles of digital arts.

However, there isn't a successful improvement on producing content with arbitrary kind of artistic style while humans or artists have mastered the skill. To solve this problem, we would like to utilize the learning algorithm, Deep Neural Networks, to extract and classify different artistic styles; we would like to use it to separate and recombine different images and generate a new image with selected artistic style as well.

2. Related Work

Convolutional Neural Networks is a type of artificial neural networks where it consists of layers of small computational units that process visual information hierarchically in a feed-forward manner. Recent studies have shown that this machine learning technique can produce promising results in image and video recognition applications (Krizhevsky et al., 2012; Simonyan et al., 2014; Sermanet et al., 2014).

In addition to object recognition, (Gatys et al., 2015) had demonstrated that the feature of style and content of an image are separable in convolutional neural network. They developed a scalable CNN by training over 1 million images. Basically, given images A and B, their algorithm was able to produce image C which matches the content of A and the style of B.

Another popular implementations of this neural style algorithm can be found in Justin Johnson's open-source Torch implementation. The program was written in Lua, and it achieves similar results to the paper. Meanwhile it also allows users to blend multiple styles into one image.

3. Technical Part

We will use VGG convolutional neural network to extract content and style in different images. This will be implemented based on caffe-network. To synthesize the style and content after retrieving them from the input images, we will use and modify the loss function in Gatys' paper to make a smooth image. Another approach for new thing in this project is trying to use some other kind of function to make the synthesizing better.

4. Milestone

- Original plan

  We planned to build our own convolutional neural network (CNN) based on the online source library, caffe-network. Caffe-network is a tool that helps train convolutional neural network. However, after reading through the online source and trying to use the caffe-network module, we figured out that it is unlikely to train our own CNN without better hardware in this short amount of time. Thus, we decided to use the pre-trained models in 'Caffe Model Zoo' from the official caffe-network website, which are also stated in the paper from Gatsy et al.

  The current stage is trying to set up the environment to run caffe-network, which is not trivial as we expected due to different OS or hardware. Moreover, we decided to use Torch to help us accelerate the process to run the algorithm.

- Remaining Milestone

  a. We are still trying to figure out a better way to do the synthesizing part. Better way could be one of the following: (1) Increase the performance of the current algorithm without making the result too worse. (2) Try to make the result look better (e.g., smoother). (3) Making both (1) and (2) happen.

  b. Since we cannot build our own CNN model in this short amount of time, we might want to compare different online models to see which one will give us a better result. E.g., maybe 19-layer models are always better than 16-layer models.
  (p.s. It is mentioned in the online source that they spent 2 weeks for

training the model even if they used the best GPU while training it on images from imageNet.)

c. The recognition part: it is easy to just use the current pre-trained models to recognize the content of the image. But it might not be trivial to make CNN recognize the style. It is worth trying whether we can extract the style feature from the figure and recognize the style at the same time.

d. The user interface: Our original thought is to build this as an online application, like filters in Instagram. However, it seems that it is unlikely to run the whole algorithm in a short amount of time. Thus, we might just want to build a simple version of user interface rather than make this to online application (either mobile application or webpage application).

- Remaining Timeline

Date	Goal
4/4	Build a working application.
4/11	Build a working application, add more features
4/18	User interface
4/25	Final presentation preparing
5/6	Final report writing

5. References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, “A Neural Algorithm of Artistic Style”

[2] Y. Shih, S. Paris, C. Barnes, W. T. Freeman, and F. Durand, “Style Transfer for Headshot Portraits”

[3] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”

[4] K. Siegel, “Picasso’s Marilyn Monroe and Other Blends: Neural Style in TensorFlow”

[5] A. Blessing, and K. Wen, “Using Machine Learning for Identification of Art Paintings”.

[6] A. Vedaldi, and A. Zisserman, “VGG Convolutional Neural Networks Practical”

[7] Caffe-network: http://caffe.berkeleyvision.org/

[8] ImageNet: http://www.image-net.org/

[9] Torch: http://torch.ch/

[10] J. Johnson. neural-style. https://github.com/ jcjohnson/neural-style, 2015.

[11] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton (2012). ImageNet Classification with Deep Convolutional Neural Networks. *In Advances in neural information processing systems, 1097–1105*

[12] Karen Simonyan, Andrea Vedaldi, Andrew Zisserman (2014). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv preprint arXiv:1312.6034.*

[13] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, Yann LeCun (2014). OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *In ICLR, 2014.*