Imitating famous artists with convolutional neural networks

We propose to develop a machine learning system modeled after an algorithm in a recent paper by Gatys et al [GEB15]. The algorithm in the paper describes a convolutional neural network algorithm that enables any given photograph to be recreated in the style of famous paintings. Specifically, the algorithm is able to take a painting and learn its stylistic composition rather than its content, then apply this stylistic information to a new image to create a new version of the image that stylistically matches the original painting. We plan on extending the original algorithm in the paper by training our system to recreate images not in the style of a single image, but in the style of an artist based on his/her representative artwork.

Implementation

We plan on creating a command-line tool in Python, interfacing with the Caffe [JSD+14] deep learning framework via pycaffe, to first recreate the algorithm demonstrated in the paper and then enable support for artist replication. Open source implementations of the algorithm using Lua and Torch do exist[1] but we'll be building an original implementation in Python to gain a better understanding of how the system works.

Once base functionality is established, we plan on expanding upon the original algorithm in the paper by attempting to recreate images not in the style of a single famous painting but by the style of an artist oeuvre. This will require either 1) learning a stylistic representation of an artist's lifetime work by training a network on not just one, but many paintings, or 2) stochastically chosing a painting or set of paintings on which to recreate a picture. Art for given artists will be obtained via automated scraping of WikiArt[2], which provides a large database of artwork categorized by artist.

Milestone 1 (10/29) Establish GitHub repository with writeups, background information, and utility scripts. Create a Linux server set up on the Caffe and pycaffe framework, from which we will do more intensive computation. Download the VGG-19 Caffe model used in [GEB15]. Write utility script to scrape artwork from WikiArt, and download art for one or two preliminary artists. (Done)

Milestone 2 (11/12) Complete base functionality to recreate an image based on the style of a single image.

Milestone 3 (12/3) Enhanced functionality: develop support for recreating an image based on preconfigured artist style representations. For now we'll pick a couple of well known artists such as Van Gogh and Picasso. Polish and finalize CLI tool.

Progress

As described in Milestone 1, we've already set up the infrastructure required to begin developing the original algorithm described in the paper. Because our laptops aren't particularly powerful, I purchased a temporary Linux server with some student credit via DigitalOcean[3], and set up Caffe. We've done preliminary investigation on the downloaded VGG-19 network on an iPython notebook, including initializing the VGG network, loading images, creating a (rudimentary) preprocessor, and calling one iteration of forward propagation with an example image.

Deliverables

Deliverables include a GitHub repository of all of the work we'll complete this semester, including a pdf writeup of the steps taken in implementing our system, the Python and Caffe code, and instructions and examples.

[1] e.g. https://github.com/jcjohnson/neural-style

[2] http://www.wikiart.org/

[3] https://www.digitalocean.com/

Helpful topics from the AI course

- Machine learning, and specifically artificial neural networks. We will be building particularly advanced models of ANNs to handle image processing tasks. In learning more about deep learning, a survey by LeCun, Bengio, and Hinton [LBH15], and a review of representation learning by Bengio et al. [BCV12] will be helpful.
- Probability can be used to incorporate random elements into the processing of the images (e.g. controlling style versus content tradeoffs, perhaps randomly selecting paintings)

Challenges

- The principal challenge is not only understanding the paper's algorithm, but extending it to incorporate the stylistic content of a complete body of images. From a technical perspective, this involves training the CNN in charge of handling style with many images of a single artist, which will also require a fundamental understanding of how the network works.
- Conceptually, it's unclear whether the stylistic information derived from training the network on several images will result in a general sense of an artist's style, or a confusing and meaningless average of an artist's work. For example, some artists vary in style considerably over their careers. If the latter, we need to make decisions about how to obtain a sensible stylistic representation. It may be useful to limit the training images to a specific subset of similar paintings.
- One challenge will be optimizing computation time. CNNs are computationally intensive and can take significant time to train. Optimizations on both the software level, such as saving model parameters, and the hardware level, such as enabling CUDA GPU computation, must be taken into account when designing our system. **Update** we've partially circumvented the hardware problem by renting a moderately poweful Linux server.

References

[BCV12] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR*, abs/1206.5538, 2012.

[GEB15] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015.

The most important article detailing the algorithm for replicating painting styles.

[JHB$^+$08] C.R. Johnson, E. Hendriks, I.J. Berezhnoy, E. Brevdo, S.M. Hughes, I. Daubechies, Jia Li, E. Postma, and J.Z. Wang. Image processing for artist identification. *Signal Processing Magazine, IEEE*, 25(4):37–48, July 2008.

Survey of a related problem of training a classifier to identify the artist of an image.

[JSD$^+$14] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B. Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *CoRR*, abs/1408.5093, 2014.

Original paper describing the Caffe deep learning framework which we'll be using to implement our algorithm.

[KHW$^+$13] Sergey Karayev, Aaron Hertzmann, Holger Winnemoeller, Aseem Agarwala, and Trevor Darrell. Recognizing image style. *CoRR*, abs/1311.3715, 2013.

CV article detailing how neural networks can be used to identify the style rather than the content of the image.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. Insight.