REMember: A Deep Dream Batch Program for Multi-Frame Formats*

Madison Blake[1], Joshua Hickman[2], Axel Hranov[3], and Matthew MacNeil[4]

***Abstract*— This is an electronic document serving as a final report for our CS 445 group's Final Project: REMember. This document provides a brief motivation of the Deep Dream project, a detailed discussion of the contexts and data that will be obtained or used in the project, description of the quantitative method used to analyze data, a time-line of milestones, expected outcomes, description of results, related work, limitations, issues encountered, future work, and conclusions.**

I. INTRODUCTION

DeepDream is a computer vision program created by Google. It finds and enhances patterns in images via algorithmic pareidolia using a convolutional neural network. This program popularized the term "deep dreaming" to refer to the generation of images that produce desired activation in a trained deep network. Now this term is used to refer to a collection of related approaches.

II. PROCEDURE FOR PAPER SUBMISSION

A. Maintaining the Integrity of the Specifications

The final report must be approximately four pages in IEEE format. The report is to contain the following information:

- a brief motivation of the project
- a detailed discussion of the context and data that will be obtained or used in the project
- a description of the quantitative method used to analyze data
- a time-line of milestones
- expected outcomes
- a description of the results
- related work
- limitations
- issues encountered
- future work
- conclusions

*This work was not supported by any organization

[1]Madison Blake is a senior interdisciplinary student in the University of Tennessee's Computer Science and Cinema Studies departments, currently taking CS 445 (Fundamentals of Digital Archeology) with Audris Mockus `web.eecs.utk.edu/~miblake6`

[2]Joshua Hickman is a senior in the University of Tennessee Knoxville's CS department, currently taking CS 445 (Fundamentals of Digital Archeology) with Audris Mockus `web.eecs.utk.edu/~prq856`

[3]Axel Hranov is a senior in the University of Tennessee Knoxville's CS department, currently taking CS 445 (Fundamentals of Digital Archeology) with Audris Mockus `web.eecs.utk.edu/~ahranov`

[4]Matthew MacNeil is a senior in the University of Tennessee Knoxville's CS department, currently taking CS 445 (Fundamentals of Digital Archeology) with Audris Mockus `web.eecs.utk.edu/~mmacneil`

B. Submission and Presentation

Final project reports are due on the last day of classes: Tuesday, December 5th, 2017.

Our team's scheduled final presentation date is Monday, December 4th, 2017.

III. MOTIVATION

Deep Dream has been around for several years, spawning several websites that allow users to upload their own images to have it be processed by a neural network. These websites include various parameters to tailor the image to the user's desires. The images generated themselves are motivating as they correlate to the original image in a way that is extrapolated and has a life of its own.

Some developers have gone as far as using Deep Dream to process individual frames in a video, producing an animated version of the aforementioned deep dream products. From this arose the motivation to create this project. While there exist on-line tools to process individual images, there is not one to easily batch together frames from videos, GIFs, and WEBMs.

Our goal is to create a script using a higher level programming language that will process a given video (in a variety of formats) by grabbing each individual frame, and running it through the neural network, applying the parameters specified by the user. The program should be able to take various parameters to specify a mode (single image, create animation, save individual frames, etc.), override default dreaming settings to produce a desired effect, specify a layer to limit the dreaming to, include a guide image, and explicitly state the output filename and format.

IV. DATA

A. Collection

Collection of data on this project is mostly limited to the image data that the neural network matches the given input on. However, this project itself further contributes to the field of Digital Archeology as it will allow futher data collection and study of the effects of image processing via neural networks.

The network operates on a trained set of data that is used as the basis for the image alterations. Fed an image, the network searches for parts that match the images that it was trained on, merging the sections that are withing a specified tolerance. There are quite a few recurring patterns that the most basic settings will implant into an image. These include dogs, birds, spires, eyes, and vehicles. The network itself is not limited to this single selection, and other trainings can be used to produce various effects.

The program allows for the specification of another image that can act as a guide during the dreaming. Rather than pulling from the neural networks trained image set, it will extract patterns from the guide image and match those to the source image.

A possible feature that may be integrated into future version is the ability to accept URL's, allowing a user to input, for example, a Youtube or Vimeo link for which to process. This can have legal ramifications, and will be require further research before it can be implemented.

In our research of the inner workings of the nueral network used by DeepDream, we found that there exist 83 layers through which the data travels through. These layers are listed in the following subsection.

B. Neural Network Layers

- data
- pool1/3x3/s2
- pool1/norm1
- pool2/3x3/s2
- pool3/3x3/s2
- pool4/3x3/s2
- pool5/7x7/s1
- conv1/7x7/s2
- conv2/3x3/reduce
- conv2/3x3
- conv2/norm2
- inception/3a/1x1
- inception/3a/3x3/reduce
- inception/3a/3x3
- inception/3a/5x5/reduce
- inception/3a/5x5
- inception/3a/pool
- inception/3a/pool/proj
- inception/3a/output
- inception/3b/1x1
- inception/3b/3x3/reduce
- inception/3b/3x3
- inception/3b/5x5/reduce
- inception/3b/5x5
- inception/3b/pool
- inception/3b/pool/proj
- inception/3b/output
- inception/4a/1x1
- inception/4a/3x3/reduce
- inception/4a/3x3
- inception/4a/5x5/reduce
- inception/4a/5x5
- inception/4a/pool
- inception/4a/pool/proj
- inception/4a/output
- inception/4b/1x1
- inception/4b/3x3/reduce
- inception/4b/3x3
- inception/4b/5x5/reduce
- inception/4b/5x5
- inception/4b/pool
- inception/4b/pool/proj
- inception/4b/output
- inception/4c/1x1
- inception/4c/3x3/reduce
- inception/4c/3x3
- inception/4c/5x5/reduce
- inception/4c/5x5
- inception/4c/pool
- inception/4c/pool/proj
- inception/4c/output
- inception/4d/1x1
- inception/4d/3x3/reduce
- inception/4d/3x3
- inception/4d/5x5/reduce
- inception/4d/5x5
- inception/4d/pool
- inception/4d/pool/proj
- inception/4d/output
- inception/4e/1x1
- inception/4e/3x3/reduce
- inception/4e/3x3
- inception/4e/5x5/reduce
- inception/4e/5x5
- inception/4e/pool
- inception/4e/pool/proj
- inception/4e/output
- inception/5a/1x1
- inception/5a/3x3/reduce
- inception/5a/3x3
- inception/5a/5x5/reduce
- inception/5a/5x5
- inception/5a/pool
- inception/5a/pool/proj
- inception/5a/output
- inception/5b/1x1
- inception/5b/3x3/reduce
- inception/5b/3x3
- inception/5b/5x5/reduce
- inception/5b/5x5
- inception/5b/pool
- inception/5b/pool/proj
- inception/5b/output

C. Uses

The use of the collected data will revolve around replacing minute sections of the original image with the matched images from the neural network. On a pixel-by-pixel scale, these alterations will be hardly noticeable (due to the network finding the closest matches); however, when viewed as a whole, there will be a visible alteration. This new image can then be pumped back into the algorithm for another iteration (either with the same settings or different parameters). Doing this over and over results in a more clearly altered image with patters that more closely match the trained images.

Once the script finishes, the final product can either replace the original source file, or be saved as a separate file with the same file extension. Currently, the program is able to output

individual frames as PNG files, or combine them together into a GIF. Future iterations will allow for processed files to be exported to other file formats including WEBM and MP4.

We believe that this technology has uses other than making interesting digital art, particularly in the field of research. The goal of a neural network is to simulate a humans brain, not only to utilize the power and effinciey of thus, but to gain insight on how the human mind reacts to stimuli and processes information. The pattern matching that DeepDream does can be applied to data other than just images. Large data sets could be processed in a similar way, in an attempt to alter it, or find similarities and differences in various iterations of the data. It could also utilize the guiding feature as a way to influence the datas path through the network.

V. MILESTONES

The program went through several iterations, each that built off the previous and added functionality:

A. v0.1

The first iteration consiststed of the base source code for DeepDream that is freely available on Google's DeepDream Github repo. This base code came with a ipynb notebook that contained code that sets up the neural network and runs example images through several modes:

- single image dreaming
- recursive dreaming of an image while zooming in
- dreaming at a specified layer
- using example guide image to influence dreaming

B. v0.2

The second iteration consisted of us having copied the code over to a python script rather than a notebook, updating any functions or code to ensure it would still work as needed.

C. v0.3

This version began to implement command line arguments that could be given to control the flow of the program including specifying input and output files. At this point only a single image could be processed at a time.

D. v0.4

This version allowed for a directory to be specified as an input argument rather than a single filename, allowing batch processing of individual images.

E. v0.5

This version allowed for the zooming mode to output each processed frame to a directory. A seperate script was written to convert a directory of frames into a GIF.

F. v0.6

This version includes the GIF-making capability and a seperate script was begun to extract frames from an input GIF.

G. v0.7

This version integrated the GIF extraction script, allowing a GIF to be specified as an input argument. Work on allowing MP4's was begun at this stage.

H. v1.0

This version represents the final iteration that we were able to achieve during the semester, as well as what we used to produce the images used in our presentation and this report.

VI. EXPECTED OUTCOME

By the conclusion of this project, we would have produced a fully developed batch program to input videos, webms, GIFs and similar files and modify them via the Deep Dream Neural network according to the specifications of the user via a general user interface. This can expand the scope of the Deep Dream neural network in terms of image processing, classification, and identification by applying in the context of multiple frames to develop a more articulate context as to the nature of the video and its particular visual elements. The videos used in the program could also provide a new training set for the furthering development of the Deep Dream neural network and computer vision.

VII. RESULTS

The goal of the project was to create a batch program to easily produce large amounts of these images/videos in GIF/WEBM formats while also being able to easily manipulate the additional parameters to the dream process. These additional parameters are displayed in figures 1-3 as they show 3 different end results from the same image. The project was also able to demonstrate that these large batches of end images can be utilized for further research in the fields of neural networks and potentially to other fields as well. The end images hold information about the original image and more, so any field dealing in large image processing could utilize the data that the dream process gives. These dreams hold the vision of the neural network, so using that information we can better tune a neural network to see what we want it to see and classify the images we want it to classify. This makes this program incredibly useful in large image scraping, as we can pick and choose large numbers of images we want to search the web for and to save based on what the neural network classifies it as. Given these potential applications, the project could be improved upon and utilized for all these different purposes.

VIII. LIMITATIONS

The limitations of the batch program are in the functionality and user-friendliness of its current design.

As of yet, a General User Interface (GUI) has yet to be developed for the program. Such implementations are deliberated in the future work section below. With the current iteration of the project, users must interface via command line in order to make runs images or GIFs or videos through the neural network.

Fig. 1. Layer: conv1/7x7/s2/20

Fig. 2. Layer: inception/3b/pool/20

Fig. 3. Layer: inception/5b/output/20

The current iteration of the design is limited to images, GIFS, and mp4 file formats. As expressed in the Milestones section, an eventual goal in terms of design is to accept a vast array of media formats, including, but not limited to: WEBMs, Audio Video Interleave (AVI), Flash Video Format (FLV), Windows Media Video (WMV), and Apple Quicktime Movie (MOV). The plans for further development of the batch program are outlined in the FUTURE WORK section below.

IX. ISSUES ENCOUNTERED

The major issues encountered throughout the completion of the Deep Dream project mostly derived from setting up the environment and loading the appropriate modules needed for the convolutional neural network to run and process data.

One such issue arose from importing the Caffe module. The Caffe module provides the deep learning framework on which deep dream is based. Caffe required many dependencies in order to function and make use of the neural network. Some of these dependencies include: CUDA, BLAS via ATLAS, MKL, or OpenBLAS, Boost (version 1.55 or greater), protobuf, gflags, hdf5, and glog. Several optional dependencies were also included into the environment: OpenCV (version 3.0), the I/O libraries lmdb, and leveldb (which in turn required the snappy library), and cuDNN for GPU acceleration.

The other major issue that arose from implementing the neural network came from the IPython notebook that was first used to demonstrate the capabilities of DeepDream, which was provided as a frame work for developing the batch program. The notebook could not import some modules since it could not find the proper path, and thus it could not propagate through the notebook and find the relevant libraries stored in the environment.

The solution to the issue of loading dependencies and libraries for Caffe was to set up a docker that had all the necessary dependencies needed to run Caffe and thus have a functional Deep Dream neural networks to process the images and videos and output the computer vision to a new image or video. This was accomplished with the assistance of Doctor Audris Mockus, who aided the discovery of a useful environment to develop the program.

The solution for handling the environment issues with the IPython notebooks was addressed by simply breaking up the notebook in to modular Python scripts, which then allowed the project a greater deal of flexibility with what formats and parameters users could specify as to not only what the neural network processed, but how it processed the input as well.

X. FUTURE WORK

If this project were to continue, there are three main goal objectives that our team would pursue development.

The first goal objective that our team would pursue would be to include handling of more file types. Images files accepted by our software are dependent upon the library python module and include exensions such as .png, .jpeg, and .ppm. Our software also supports .gif and .mp4 extensions. To further develop this functionality, our team would also implement the logic for the .webm extension which is a high quality open video format for the web.

The second goal objective that our team would pursue would be to create a Graphical User Interface, or GUI. Since our team did not accomplish Milestone Release v3.0 (Future version goals will include creating a GUI to more easily allow the user to select files and set parameters rather than having to deal with command-line arguments), this is a relatively high priority for the future of REMember. Having a graphical user interface would make the software our team has developed more accessible to a wider audience. General purpose users would be contributing

The third goal objective that our team would pursue would be to train new neural networks. The Google Deep Dream's orignal set of trained images is not the only source of translations for the specified image. There are other sets of trained images that are able to be ported into the software. These are the guiding images for the visual phenomenon of the translated image. This is a phenomenon that is deserving of further study because it corresponds to how neural networks could operate on data that was not of a visual nature.

XI. CONCLUSIONS

This project contributes to the field of Digital Archaeology as it will allow for further analysis and recording of data related to the effects of neural networks.

ACKNOWLEDGMENTS

Our team wishes to acknowledge the distinguished University of Tennessee in the glorious city of Knoxville. Our success is due to the honorable Doctor Audris Mockus. His guidance and influence gives inspiration to all who are involved in the field of Digital Archaeology as well as the many other disciplines of Computer Science. We would also like to thank the EECS Department at the University of Tennessee for providing us with the lab machines and host to contain the docker environment to run Caffe, and by extension the Deep Dream neural network.