Dreamy Food: Turning World of Warcraft Art into Food

Bjarke Larsen
balarsen@ucsc.edu
UC Santa Cruz
California, USA

Anya Osborne
anyaosborne@ucsc.edu
UC Santa Cruz
California, USA

Ethan Osborne
etosborn@ucsc.edu
UC Santa Cruz
California, USA

Abstract

This paper presents the results of the final class project about artifacts that appreciate art. The project is based on the trained Generative Adversarial Network (GAN) of World of Warcraft (WoW) icons that generates new stylized art using the Google Deep Dream algorithm and Gradient Ascent. We then used a trained food classifier using the Inception V3 model and applied it to classify the resulting Deep Dreamed images of WoW art. We show that our algorithm can be employed for creating stylized images that can be viewed by the machine as food.

CCS Concepts: • **Networks → Network experimentation**.

Keywords: Generative Adversarial Network (GAN), Google Deep Dream, Gradient Ascent; Inception V3, World of Warcraft (WoW), food classifier.

ACM Reference Format:
Bjarke Larsen, Anya Osborne, and Ethan Osborne. 2021. Dreamy Food: Turning World of Warcraft Art into Food. In *Proceedings of UCSC Computational Media 202 (Conference 2021).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

The idea behind the project is to have the machine create new stylized World of Warcraft icon art and “understand” it by classifying the generated images within a specific domain. Researchers use Deep Dream and Gradient Ascent to train a network by showing it a lot of examples of what they want them to learn, hoping they extract the essence of what matters, and learn to ignore what does not matter. However, the challenge is how do we validate that the network has learned the correct features? In attempting to answer this question, we propose to use a trained food classifier model that functions as a desired classification vector in our project. It generates a machine-hallucinated WoW art in the realm of food and validates its own product by classifying it with 82.5% accuracy. The results are intriguing - we showed how a relatively simple neural network structure can be applied to interpret art within the given domain. When we were children, we enjoyed watching clouds and interpreting their random shapes, so does our network - although trained on images of food, it tends to interpret abstract strokes in WoW art as food.

2 Related Work

Our project is related to the popular topic of attribute-based image processing. This task has the longest history in image processing with deep convolutional neural networks (CNNs). The concept of it is in generating an image of a certain object, in which a machine performs a pre-image search to look for an image that maximises the activation of a certain unit in the classification layer of a CNN that’s trained on visual recognition. In the pioneering work, Simonyan et al. showed that this technique indeed produces images that somewhat resemble the features of a given image and class [11]. The pre-image search procedures have been widely used in the past to understand and visualise the feature representations of CNNs [3, 6, 10, 12]. In more recent work, Nguyen et al. [8, 9] showed that a machine can generate diverse and fully realistic natural images of certain object classes by using a generator network that is trained to invert CNN representations [3] as a system that regulates the pre-image search. In computer vision, the visualization techniques applied inside the CNNs help understand what features a machine thinks are distinctive in an object to differentiate between classes. The idea of visualizing the convolutional layers was proposed by Erhan et al., they showed techniques to gain insight into what a particular unit of a neural network represents [4]. The widely popular Deep Dream algorithm developed by Google engineer, Alexander Mordivintsev, is also developed based on this principle of activation maximisation [7]. It uses a convolutional neural network to find and enhance patterns in images with powerful AI algorithms. It allows to create a dreamlike hallucinogenic appearance in the deliberately over-processed images. Such technique is based on a gradient-based approach that has shown to be very attractive in its simplicity. However, as Yosinski et al. noted,

Conference 2021, March 15–19, 2021, Santa Cruz, CA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

the optimization process tends to produce images that do not greatly resemble neutral images [14], so they proposed a technique to solve that problem. To validate that the Deep Dream algorithm learned the correct features in computer vision, we decided to use an alternative approach - a food classifier model that could tell us about what type of image it thought it produced. As a reference, we used the project developed by Harim Kang in 2020, which is publicly available on GitHub at harimkang/food-image-classifier [5]. This project allowed us to train the classifier and produce the trained model with 82.5% accuracy in food prediction.

3 The Dataset

The dataset we applied in this project is represented by a combination of two separate pieces. The first piece was the GAN-generated images of World of Warcraft art (Figure 1) developed during the previous class project. It was used as an input to see what a machine does with its own GAN-generated images using a Deep Dream algorithm.

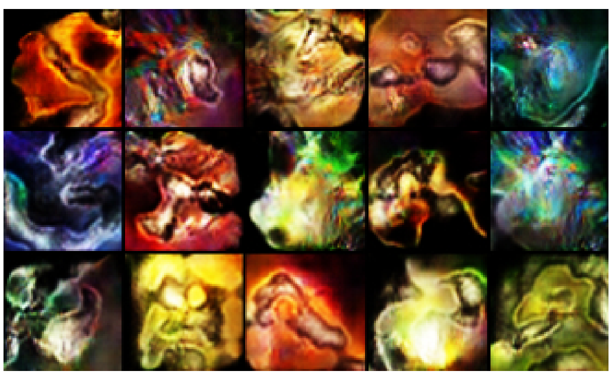

Figure 1. GAN-generated WoW art dataset used as input.

The other piece of dataset was used to train a model that classifies the Deep Dream hallucinated WoW art. This dataset is 101 Food dataset which is publicly available at www.kaggle.com [2]. It contains images of food, organized by type of food. In contrast to scene classification or object recognition, food typically does not exhibit any spatial layout (i.e. we cannot find similar patterns relating ingredients of a mixed salad). Therefore, food recognition is a specific classification problem that requires models to exploit local information. Bossard et al. manage to solve this problem by creating a model that identifies discriminative image regions which help differentiate one type of dish from the others [1]. The produced dataset is represented by real-world food recognition with 101,000 images (Figure 2).

Figure 2. 101-Food dataset used to train the classifier.

4 Approach

Given the two datasets, the primary challenge was in combining them properly in our project. The GAN and GAN-generated images were largely untouched from our previous work, whereas the food classifier needed to be trained for our specific use. We then created the gradient ascent model, using the DeepDream, and modified it to meet the following performance requirements: (1) Use the given dataset as an input; (2) Use a food classifier model to produce new images; (3) Classify these new images as food. The full project structure can be seen in Figure 3.

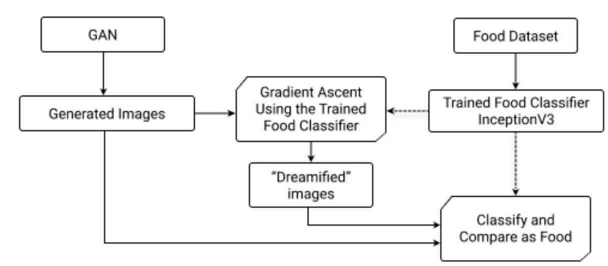

Figure 3. Project Structure.

4.1 Training the Food Classifier

The food classifier [5] was trained on the images mentioned previously ([2]), which contains 101 thousand images, 101 classes of food (thus, 1000 images each). It trained overnight for 9 Epochs, and produced an accuracy of 82.5% on its testing (Figure 4). This model and checkpoint was then saved and used for the gradient ascent and classification, where it did not change further.

4.2 Performing the Gradient Ascent

The Gradient ascent model is based on Google DeepDream, where we followed the implementation guidelines described at www.tensorflow.org/tutorials/generative/deepdream [13]. The basic gradient ascent implementation included minimum changes. The idea was to supply the DeepDream with an image to "dreamify", and a model through which to do it. Certain layers (chosen, more on that later) in the model were accentuated by maximizing the loss function to "excite" those layers in the image, creating a new image where those layers were visually present in the output. In the original DeepDream, which uses the core InceptionV3 model, this could be used to see specific uses of each layer, for example,

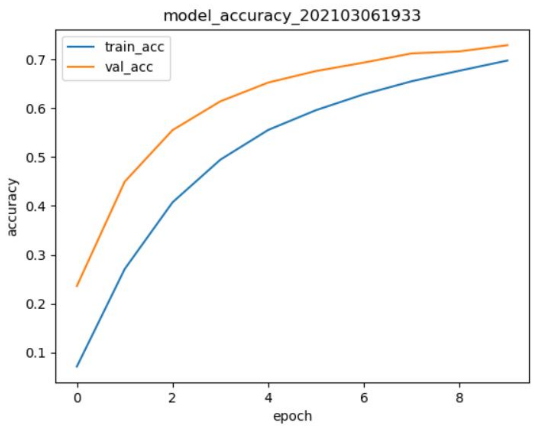

Figure 4. Model Accuracy from training food classifier.

some later layers have clear eyes or face shapes, whereas earlier layers had more primitive shapes. For this implementation, the original model is replaced with the food classifier trained for this project. DeepDream is designed to work with classifiers and the food classifier is also an InceptionV3 implementation, so replacing the models was rather straightforward. Since the model has been trained on food, rather than what the original InceptionV3 model was trained on, the idea is that the layers will contain information about "food" in general, which the model uses to classify food. By exiting these layers, we anticipated to get more "food-like" results in the output. It is worth noting that it was not related to a specific type of food (like apples), but rather all food in amalgamation that the classifier has been trained on. There were a few more tricks to the DeepDream implementation that we also took advantage of. First, was the "octaves" component, a gradual upscaling of the image to apply the gradient ascent at various scales on the output image, causing the produced patterns to process at several scales at once, rather than uniformly. This had the added benefit that it upscaled the image naturally, where the original 64x64 images were formatted into the respectable sizes in the output. We settled with an output size of 1024x1024, as this was a good middle-ground between detail and processing time, although experiments were completed with both 2K and 4K formats. A second trick from the DeepDream was to tile the image and perform octaves on each tile, which helped with large octave counts and high resolution images where the entire calculation might not fit in memory. This tweak was applied as a natural extension of the octavization method. We experimented with the parameters for each of these for a while, and settled on an octave range of -2 to 3, causing 5 octave-upscales from 64->1024, at an octave scale of 2. Tile size remained unchanged at 512, and step count and size ended at the default 100 and 0.01, respectively. Finally, the last trick was made in regard to selecting the desired activation layers in the model, which was also a result of our experimentation. Figure 5 represents the output of different layers to demonstrate the extent of the impact this choice had on the output:

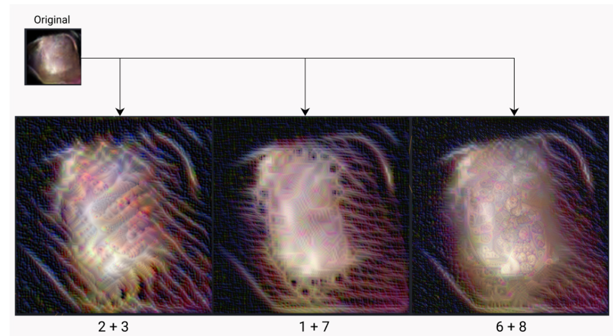

Figure 5. Example of Output of Different Activation Layers.

Earlier layers like 2+3 showed more rudimentary patterns and lines, and led to a more "dissolved" output, whereas higher layers like 6+8 triggered more intricate, detailed patterns and uniform shapes to appear in result. However, higher layers required longer processing time, so the majority of tests were done on lower layers or combos, such as 1+7, especially when dealing with large amounts of images.

4.3 Classifying Images Again

The final aspect of the project was to use the food classifier in the more regular way and classify our output images and input images as if they were food. This was completed by iterating over the generated images and asking the model for a prediction, which used the food classifier's built-in prediction method. It saved the images with the predicted label for future use as well as printed the output in the console along with adding it to a .txt file for future comparison. It was done on both the "original" (as in GAN-generated) 64x64 image and the newly generated-through-gradient-ascent 1024x1024 image, which was then used for comparison in its classification.

5 Results and Evaluation

The Output includes two steps: (1) Generating dreamified images using gradient ascent; (2) Classification of the original image (forbatch) and the resulted dreamified images using pre-trained 101-food classification model.

5.1 Generating Dreamified Images using Gradient Ascent

As pictured in Figure 6, the previously generated WoW icons were fed into the Deep Dream model to become dreamified. The resulting images kept their original amorphous appearance, but had the added effect of new patterning based on

Figure 6. Dreamified GAN Generated WoW Icons.

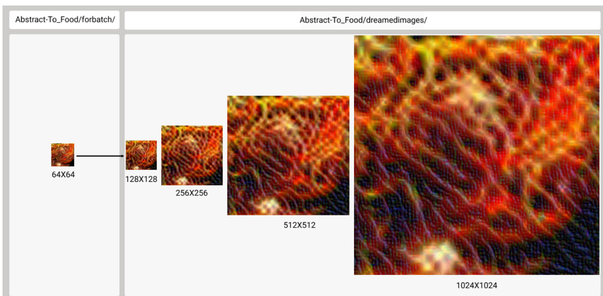

Figure 7. Octave Upscaling of Dreamified Images.

the layers used. Images in Figure 7 are the result of activating layers 1+7.

During the dreamification process, it was decided to also use the aforementioned Octave upscaling on the images. The results allowed for the natural progression of dreamifying to be applied to the generated WoW icons as they moved from 64x64 to 1024x1024 without any real feature loss. The upscaling also allowed for a greater understanding of how the dreamification was being applied to the generated WoW icons. Figure 7 details the gradual upscaling.

5.2 Classification of the Original Image and the Output using Food Domain

The next step was to test the original generated WoW icons against the dreamified WoW icons using the Food Classifier.

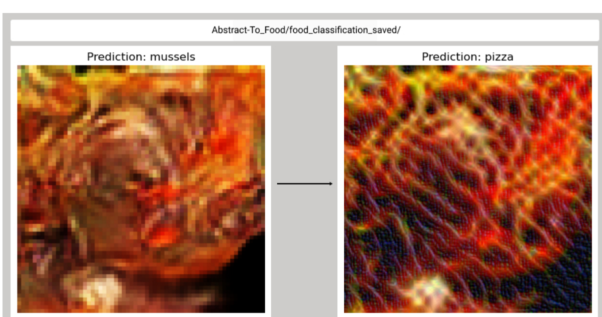

Figure 8. Non-dreamified vs Dreamified Classification.

As seen in Figure 8, the food classification model classifies the non-dreamified image as mussels and then predicts the dreamified image as pizza. There is a large difference between mussels and pizza to a human, but the food classification model picked out approximations from training that were similar to what were present in these two images.

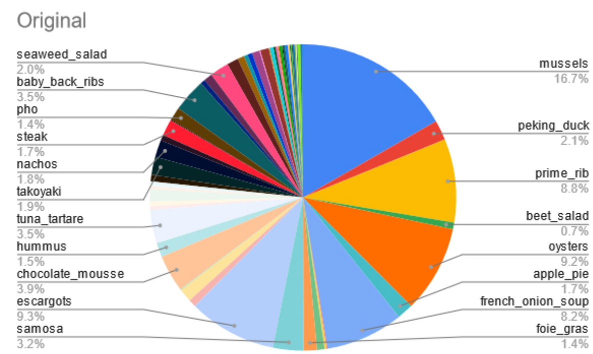

Figure 9. Non-dreamified WoW Icon Classification.

1,000 images were run through the food classifier, pre- and post-dreamification. Figure 9 details the breakdown of the pre-dreamified images. Mussels were the most predicted at 16.7% followed by oysters at 9.2%, prime rib at 8.8%, french onion soup at 8.2% and the least predicted label was beet salad at 0.7%.

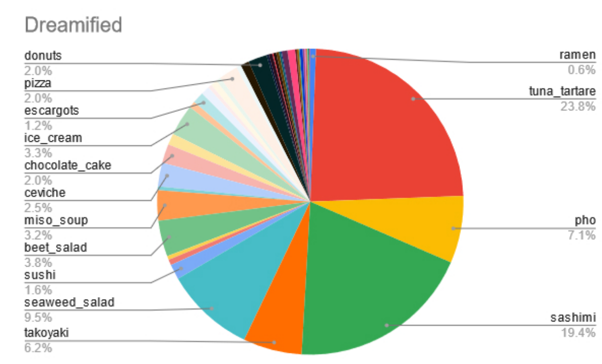

Figure 10. Dreamified WoW Icon Classification.

Figure 10 details the post-dreamification breakdown of classification. Dissimilarly, none of the top labels from the pre-dreamified images were present in the post-dreamified labels. Also, beet salad made up 3.8% of the labels this time. Other labels include tuna tartare at 23.8%, sashimi at 19.4%, seaweed salad at 9.5%, and the least predicted label at 0.6% was ramen.

6 Conclusion

In the future, we would like to test a different model that would create more food-like images. For example, if there exists a food model that has less categories, or just a single category then we could see if the dreamified images came out more food-like and less ambiguous. As mentioned before, one of the issues of the current dataset was too many different food labels which led to the gradient ascent model not focusing on individual aspects to accentuate, but rather an amalgamation of all aspects. To summarize, the procedures described in this paper helped us understand and visualize how neural networks are capable of carrying out difficult classification tasks, and test what the network has learned about its own product after training. Being able to see what the network thinks about the generated objects helps understand potential ways for improving the network performance.

References

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101 – Mining Discriminative Components with Random Forests. In *Computer Vision – ECCV 2014 (Lecture Notes in Computer Science)*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 446–461. https://doi.org/10.1007/978-3-319-10599-4_29

[2] DanB. 2018. Food 101. Published on Kaggle at https://www.kaggle.com/dansbecker/food-101/. Accessed March 16th, 2021.

[3] Alexey Dosovitskiy and Thomas Brox. 2016. Generating Images with Perceptual Similarity Metrics based on Deep Networks. *arXiv:1602.02644 [cs]* (Feb. 2016). http://arxiv.org/abs/1602.02644 arXiv: 1602.02644.

[4] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pascal Vincent, and P O Box. [n.d.]. Visualizing Higher-Layer Features of a Deep Network. ([n. d.]), 13.

[5] Harim Kang. 2020. FOOD!!!-Classifier. Published on Github at https://github.com/harimkang/food-image-classifier. Accessed March 16th, 2021.

[6] Aravindh Mahendran and Andrea Vedaldi. 2014. Understanding Deep Image Representations by Inverting Them. *arXiv:1412.0035 [cs]* (Nov. 2014). http://arxiv.org/abs/1412.0035 arXiv: 1412.0035.

[7] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. 2015. Inceptionism: Going Deeper into Neural Networks. https://web.archive.org/web/20150703064823/http://googleresearch.blogspot.co.uk/2015/06/inceptionism-going-deeper-into-neural.html

[8] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. 2017. Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space. *arXiv:1612.00005 [cs]* (April 2017). http://arxiv.org/abs/1612.00005 arXiv: 1612.00005.

[9] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. 2016. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *arXiv:1605.09304 [cs]* (Nov. 2016). http://arxiv.org/abs/1605.09304 arXiv: 1605.09304.

[10] Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. *arXiv:1412.1897 [cs]* (April 2015). http://arxiv.org/abs/1412.1897 arXiv: 1412.1897.

[11] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv:1312.6034 [cs]* (April 2014). http://arxiv.org/abs/1312.6034 arXiv: 1312.6034.

[12] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. *arXiv:1312.6199 [cs]* (Feb. 2014). http://arxiv.org/abs/1312.6199 arXiv: 1312.6199.

[13] TensorFlow. 2021 (last updated). DeepDream Tutorial. https://www.tensorflow.org/tutorials/generative/deepdream. Accessed March 16th, 2021.

[14] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. 2015. Understanding Neural Networks Through Deep Visualization. *arXiv:1506.06579 [cs]* (June 2015). http://arxiv.org/abs/1506.06579 arXiv: 1506.06579.