Art and Machine Learning
CMU 2019 Spring
Project 1

Deep Nightmare

Mira Murali
Loki Ravi
Izzy Stephen
Yuhan Xiao

Concept

In its most basic form, Deep Dream manipulates imagery through an algorithmic application of pareidolia: the perception of patterns, like images, symbols, or messages, from data wherein no such patterns exist. But trained on nightmarish imagery, a program could be susceptible to more detrimental visual disturbances. Deep Nightmare is motivated by one of the questions discussed at the 'Paradox: Frames and Biases' symposium. Could AI experience the mental chaos that plagues so many people? Could we teach it to understand fear and to feel it?

When biological neurons malfunction or are artificially stimulated, delicate systems of the visual cortex are thrown off balance, and what was once a clear view of reality gives way to altered forms of perception. Humans suffering from mental illness experience terrifying visual hallucinations. But perception itself is not the only domain of consciousness that can be haunted by dark imagery: as the brain accesses memory traces during sleep, memories of fear-inducing scenarios are activated and manifest as nightmares.

The instinctual terror we feel when faced with the deathly, dangerous, or uncanny is deeply ensconced in our biology. To be afraid of imagery produced by one's own consciousness seems like a failure of cognitive heuristics. In sharp contrast to the weaknesses of a complex biological neural network like a human being, AI is conventionally seen as perfectly rational, cold-blooded, created in the image of a perfect, flawlessly logical intelligence. Through this project, we question whether AI will remain unsusceptible to the detriments of the human intellectual experience as it approaches our level of consciousness.

The artwork presented here, created by a neural network trained on a wealth of nightmare and hellscape imagery, imagines what an artificial neural nightmare might look like. An AI with the consciousness level of a human would likely fear different stimuli from a human, so in actuality, its 'nightmare' might look slightly different. Nonetheless, the conceptualized 'nightmares' born from this program offer insight into the possibilities for an AI's greatest sufferings - and perhaps its potential for creativity.

Technique

We implemented the DeepDream algorithm using a ResNet architecture trained on a dataset of "scary" images.

- We made the depth of ResNet a hyperparameter, allowing us to experiment with layers.
- We constructed a binary dataset of "nightmare" and "non-nightmare" images.
- We constructed a test set with diverse visual composition to dream on.
- **Training**:
  - We split the data into training + validation
  - We tried using a model pre-trained on ImageNet features.
- **Testing**:
  - *Octaves* : Dream better using finer details from the input image.
    - This feature allows us to zoom into the input image and dream at various levels of zoom.
  - *Layers* : Choose between simple features and complex features to dream.
    - This control comes from choosing to maximize the activations of deeper layers or shallower layers. The deeper layers tend to have more complex features.
  - *Iterations* : Pass the image in a loop through the dream generator.
    - Activations get maximized multiple times through this process and brings out sharp clear dream features that dominate the input image.
- **Interface** : We constructed an interactive interface to let users interact with our algorithm and model. This interface lets users see the "dream" superimposed on the original image.

Process

Problem Formulation
Our initial problem statement called for training a binary classifier on nightmares and landscapes. This approach was intended to learn the features of nightmares and then dream on an innocuous landscape. We expected to see “scary” features on the landscape images we had hand-picked for visual composition. Unfortunately, this was an ill posed problem as the test images got classified as landscapes and maximizing the activations simply highlighted the existing features.
As a result we re-formulated our problem statement to dream on a class of images the model was not trained on.

Network Architecture
Our initial use of models pre-trained on ImageNet features turned out to be counterproductive as these models were trained on 1.2 million images for 140 epochs and had learnt the features of those classes (birds,dogs, etc) too well. As a result, our “dreams” were made of features from the pre-training dataset.
Thus we decided to abandon the use of pre-trained models in favor of training purely on our own dataset.

Dataset Design
The dataset we initially constructed was a simple binary set of 2000 images classified as scary/non-scary. Unfortunately, the features in this dataset were too far apart in terms of variance to be able to train a model from scratch. While training we found that we were rarely able to beat a random classifier with an accuracy of 52%. We called this dataset ‘Initial Dataset of Landscapes’ (IDL).

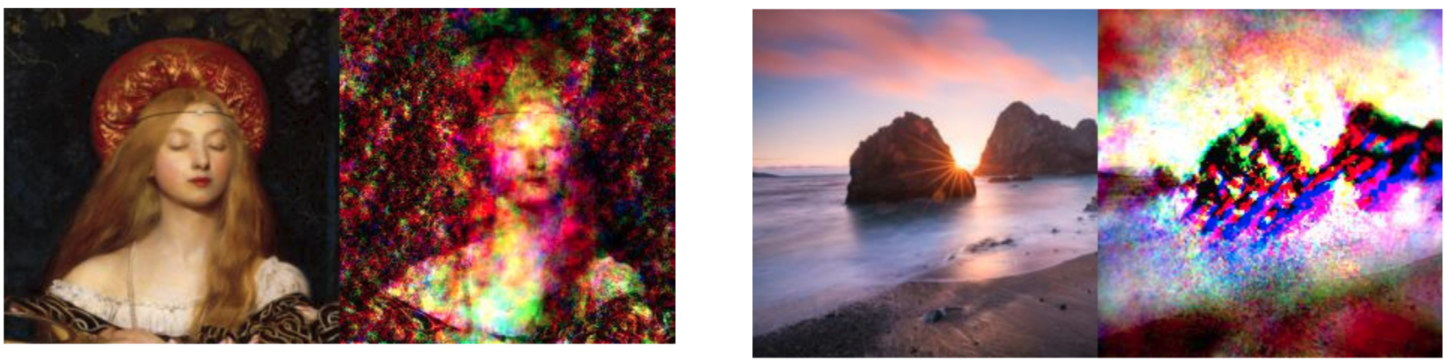

Fig 1:: ResNet50 trained on the IDL
ur inference from the previous failure was that the model was unable to learn any key features of a nightmare, instead focussing on the pleasantness of the landscapes and lack thereof in the nightmares. This was supported by the significant darkening of images while dreaming. Hence we decided to extract the features of a nightmare manually into the dataset. We constructed a dataset with fangs, devils faces, and other features from our original dataset. We named it ‘Improved Dataset of Nightmare images’ (ImPN). Unfortunately this did not improve our results.

Fig 2: Possible side-effect of the color jitter transformation on ResNet34 trained on ImPN
Hence, we set out to build a new dataset keeping in mind our past mistakes. We built a dataset with 3000 images of 10 classes namely : creepy crawlies, creepy walkies, skulls, satan, girls from horror movies, frogs, jellyfish, clowns, lizards, scorpions. This dataset was aptly named ‘Animals and Nightmare Faces’ (ANF). With this dataset we faced a new problem with the limitations of deep learning in vision tasks. Some of the images had minor occlusions, reflections, and multiple classes in the same image. Despite this, we were able to achieve an accuracy of 81%.

Process

Fig 3: ResNet34 trained from scratch on ANF
We identified these problem images using validation misclassification as a clue and sought to remove the same. Unfortunately this culling led to the dataset numbers dwindling to 2800 images (and a new version of ANF, named ANF--). While we were able to train a model to 97.7% accuracy, the CNN filters were not well formed indicative of insufficient training instances.

Fig 4: ResNet18 trained from scratch on ANF--

Experiments
The following table summarizes the list of experiments we conducted on our three datasets. There was one particular data augmentation method worth mentioning as a hyperparameter: PyTorch's ColorJitter data transformation. Using this function, it was possible to increase the brightness and saturation of an image. We used this transformation method to ensure that our model did not misinterpret a dark image to always be a nightmare image. However, after observing the results of our experiments when this transformation was used, we were a little uncertain about its usefulness.
IDL: Initial Dataset of Landscapes
ImPN: Improved Dataset of Nightmare images
ANF: Animals and Nightmare Faces
ANF-- : Animals and Nightmare Faces with commonly misclassified images removed
*It must be noted that the validation loss and accuracy presented above did not occur simultaneously. They denote the lowest loss and the highest accuracy achieved using that particular configuration of hyperparameters. It must also be mentioned that these evaluation metrics do not directly relate to the visual results obtained.

ResNet Version	Dataset	Learning Rate	#Epochs	Pre-training	Color Jitter	Validation Accuracy
152	IDL	8e-5	150	Yes	Yes	93%
152	IDL	8e-5	140	No	Yes	53%
50	ImPN	8e-5	300	No	Yes	53%
50	ANF	8e-3 (decayed twice)	300	Yes	No	87.3%
34	ANF	8e-4	400	No	Yes	80.5%
34	ANF	8e-4	500	Yes	No	73.8%
18	ANF	8e-4	500	Yes	No	65.2%
18	ANF	4e-4	700	No	Yes	81%
18	ANF--	1e-4	200	No	Yes	97.7%

Result
With our final multi-class dataset (ANF), and pre-training, the shallower versions of ResNet were somewhat biased towards the creepy crawlies class. The model often dreamt of worms when given a testing image:

Process

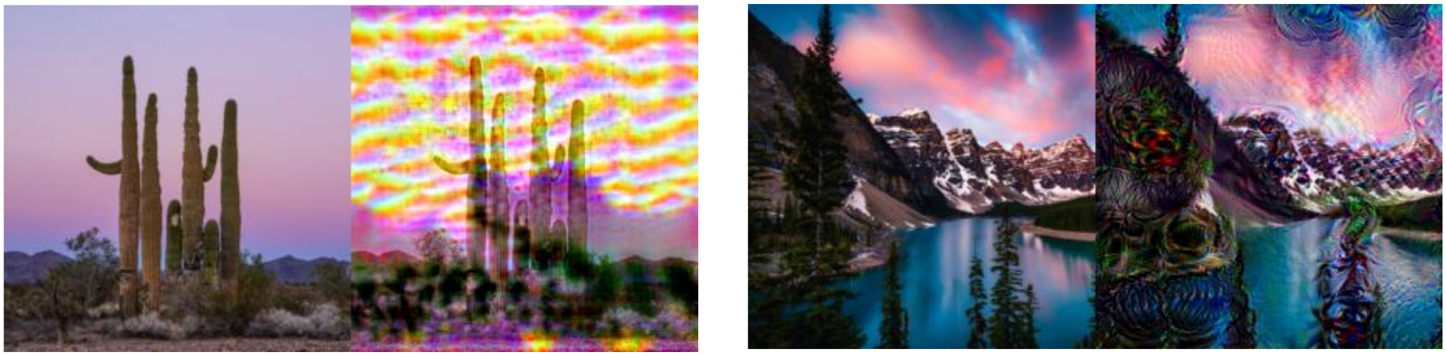

Fig 5: Result of training a pre-trained ResNet34 on ANF

Our best performing model so far has been the pre-trained ResNet50, with no color jitter transformation and decayed learning rate. Although the validation loss and accuracy do not reflect the artistic inclination of this particular model, we believe that it is apparent from the sample image given below:

Fig 6: Result of training a pre-trained ResNet50 on ANF

Reflection

While our model did not exactly create the hellscape we hoped it would, we developed some intuition about why it could not: because of the skewed data. An AI learns best from the perfect dataset: clear images with informative features that all have some underlying structure. But such a dataset does not exist and so, like Manuela Veloso mentioned in the Paradox symposium, the AI learns the biases present in the data although it is, in itself, entirely impartial.

With a more structured dataset which has far more images, and a few hundred hours of training, our model could possibly learn the concept of nightmare and gain the ability to see ‘fear’ in any landscape it dreams of.

Code

https://github.com/mira-murali/deep-nightmare

Contributions

Dataset collection and artistic meaning: Isobel Stephen and Yuhan Xiao
Preparing the data, dataloading: Mira Murali
Training, testing and network architecture code: Lokeshwaran Ravi
GUI for the project: Yuhan Xiao