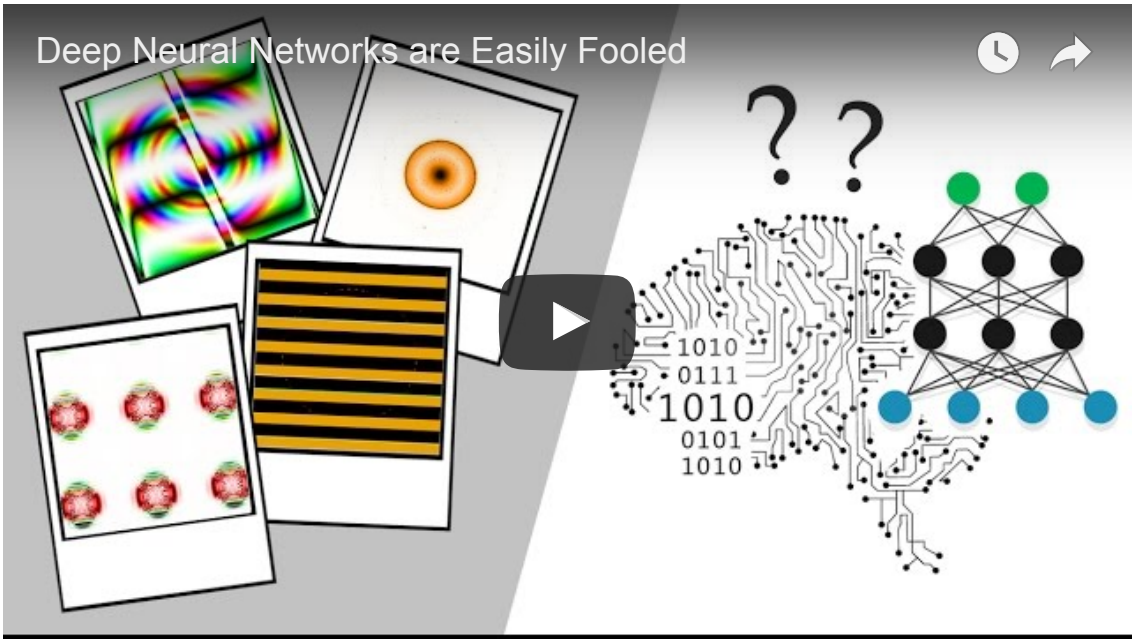
MAS.S63 Final Project

An attempt at understanding the differences between the representations used by deep convolutional neural nets and those used by humans

State-of-the-art deep neural nets have achieved some pretty incredible computer vision results recognizing faces, reading handwriting, interpreting EKGs and even describing what's happening in a photograph. Yet computers don't see things the same way as humans, and they're not without flaws. Fooling images exploit some of these flaws. A fooling image is an image that state-of-the-art neural nets classify with a high degree of confidence as one thing, whereas a human observer would find the image unrecognizable.

One well known paper on the topic is Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images (http://arxiv.org/pdf/1412.1897v2.pdf) by Nguyen et al., summarized in this video:



Computer vision and human vision are nothing alike, and it's clear that deep neural nets approach image recognition in a fundamentally different way than human observers. Yet, we don't really understand how computer vision differs from our own. One particularly fascinating finding from the paper is that the fooling image results are fairly consistent. Different neural nets presented with the same seemingly indecipherable image will assign the same label. Even iPhone image recognition apps powered by deep learning exhibit the same mistakes. I played arond with a few such iOS apps, and found WhatsThatPic the most convincing. If you have an iPhone, download the app and point it at the below image. Does it think it's the wheel of a car?

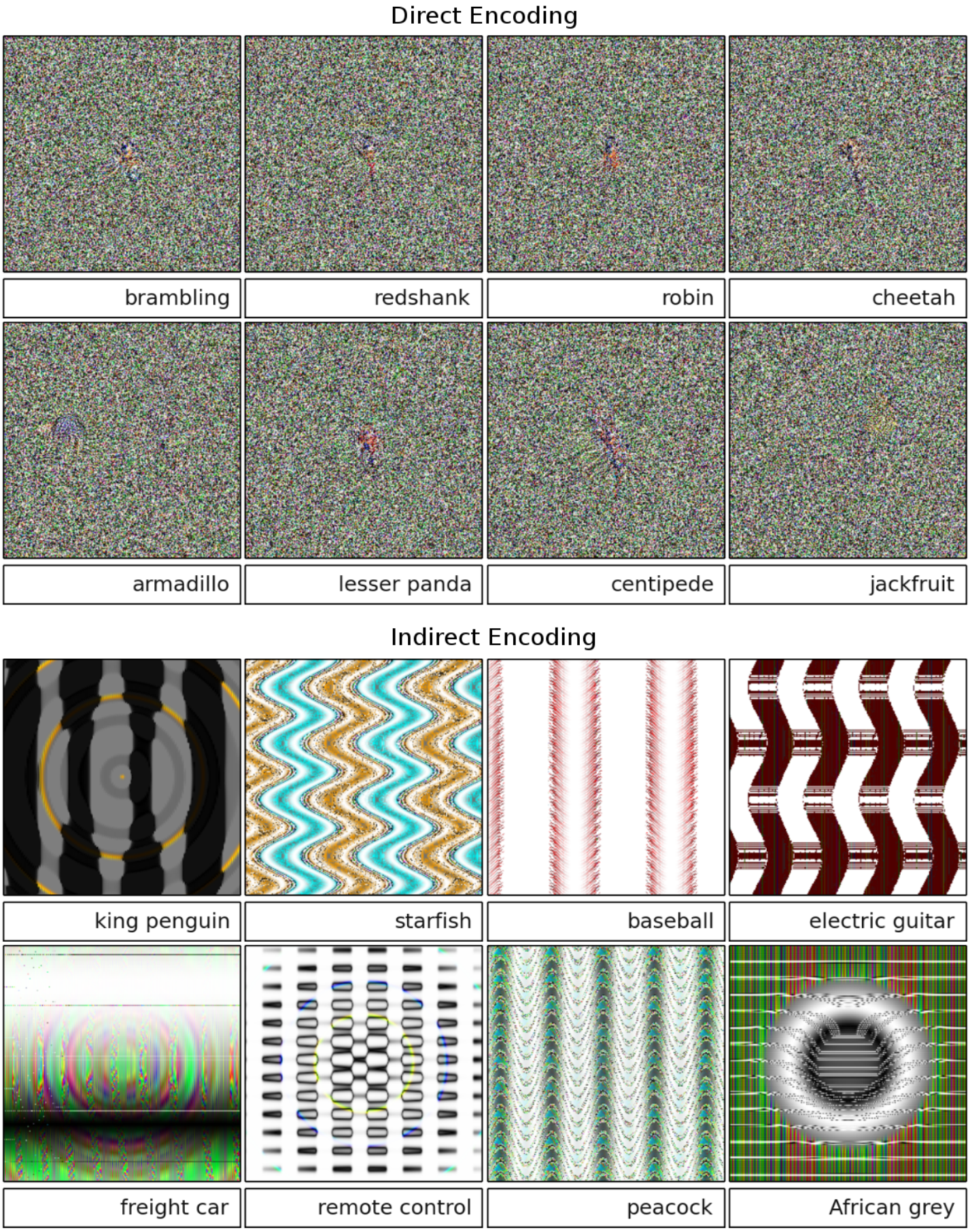
A fooling image classified as a car wheel with ≥ 99% confidence (Nguyen et. al, 2014).

The app didn't give me the 99.9% confidence of Nguyen et al., but that's still a pretty crazy result given that the neural net used by the app is completely different than the one from the paper, and the photo taken by your phone has the glare from a computer screen and different lighting, angle, and camera lens. The result implies that different DCNNs learn the same discriminative features for each class. Yet even though both neural nets are fairly certain that you're looking at a car wheel, that wouldn't be my first identification. If you spin in a circle a few times and then squint your eyes, you can sort of see a car wheel, but you would be unlikely to assign that label without priming.

Nguyen et al. produce their novel fooling images by taking deep convolutional neural nets (DCNNs) trained to perform well on the ImageNet dataset (http://www.image-net.org/) and applying evolutionary algorithms (EA) or gradient ascent to the images the DCNNs label with high confidence as belonging to each dataset class. EA are inspired by Darwinian evolution. In this case, the images represent a population of "organisms" that undergo a random perturbation, with the best being selected for in each iteration based on the confidence value of the prediction (the fitness function) assigned to it by the DCNN.

Encodings are how an image is represented as a genome. The car wheel from above represents an indirect encoding. In indirect encoding, compositional pattern-producing networks (CPPNs) are manipulated and serve as the encodings of images. Compositional pattern-producing networks (CPPNs), are a variation of artificial neural networks which apply different activation functions. Artificial neural nets often employ only sigmoid functions and occasionally Gaussian functions, but the CPPNs from Nguyen et al. include sine, sigmoid, Gaussian and linear activation functions. The choice of function biases towards specific types of patterns and regularities, and is more likely to produce regular images that resemble known objects. You can be your own EA fitness function at PicBreeder.org (file:///home/berwick/Downloads/PicBreeder.org) by manually selecting images you like, which then become the parents of the next generation.

Nguyen et al. also produced fooling images via direct encoding. In direct encoding, individual image pixels are directly manipulated one at a time in the HSV color space (each of the 256 x 256 pixels in each of the images is assigned three integers: H, S, V) starting from a random initialization.



Differences in EAs with direct and indirect encodings, all classified with > 99.6% confidence (Nguyen et. al, 2014).

We know our vision system is completely different than that of a DCNN, and my goal is to begin to get an intuition for some of the differences in how neural nets and humans perform visual identification. Specifically, I decided to focus on two questions:

1. **Do the identification mechanisms used by human observers and DCNNs converge at a certain level of abstraction, or for a specific image class?**

2. **Is there a pattern in the underlying structure of images in certain image classes that makes it easier to generate fooling images for that class?**

For this project, I decided to focus on indirectly encoded natural images. I found the resulting human-recognizability of the images more interesting than the static appearing to result from direct encoding or gradient descent. Furthermore, Nguyen et al. concluded that the directly encoded EA was less successful at producing high-confidence images than indirectly encoded EA (after 20,000 generations, the median confidence for the directly encoded generated images was only 21.59%).

I began my exploration with Adam Kraft's draft paper, Exploring DCNN Features via Signal-Energy Reduction. Kraft leads with the same premise as Nguyen et al. -- though DCNNs perform up to state-of-the-art on image classification, the neural nets can be adversarially fooled into confidently misclassifying images that would never fool a human -- but approaches the issue from a different perspective. He focuses on revealing the underlying structure of images by removing signal energy from correctly classified images to create minimal stimuli for DCNNs. The result is a new class of reduced-energy images, some of which are human-recognizable as belonging to the corresponding original image's image class, and others which are completely unrecognizable.

At a high level, the signal-reduction algorithm consists of 5 basic steps:

1. Divide the source image into its Laplacian pyramid

2. Generate candidate images by randomly selecting a level from the Laplacian pyramid (excluding the image-mean level), and then setting all the pixels within a randomly selected rectangular region on that level to zero

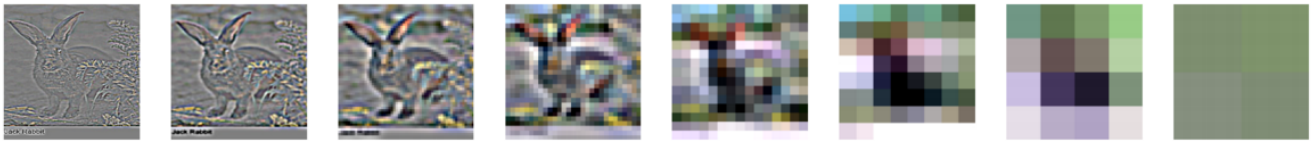
3. Evaluate candidates by reconstructing the image from the modified Laplacian pyramid and assigning a label and confidence for each candidate using the DCNN

4. Reject all candidate image modifications with a confidence level below that of the original or that get assigned a different label

5. Repeat the signal-reduction algorithm until any further removal would decrease DCNN confidence

In summary, laplace pyramids are repeatedly subtracted from the pyramid representing the orginal image, resulting in a residual image with all of the high-confidence features found from consecutive runs of the signal-reduction algorithm removed. A Laplacian pyramid is simply the difference between up-sampled Gaussian pyramid level and Gaussian pyramid level. Laplacian pyramids allow for the selective removal of signal energy, providing insight into the frequency and spatial distribution of the most salient features used by DCNNs in classifying natural images. The approach relies on the assumption that applying the signal-reduction algorithm repeatedly enables the identification of a nearly complete set of features that the DCNN relies on for detection (surprisingly, less than 10% of the signal energy contained in each of the top 5 Laplacian pyramid levels was found to contribute to the confidence detection of the image label). The below figure shows the application of the signal-reduction algorithm on an example Laplacian pyramid.

(a) Starting Image



(b) Starting Laplacian Pyramid

(c) Final Laplacian Pyramid

(d) Final Image

The signal-energy reduction algorithm starts with an image (a) and computes its Laplacian pyramid, a lossless representation of the input image comprised of 7 layers that act as band-pass filters for image signal energy, and a residual 4-pixel image containing the mean of each quadrant of the image. The frequency-selective layers of the Laplacian pyramid corresponding to the starting image are shown in (b), with visual enhancements to show contrast. The signal-energy reduction algorithm produces a final Laplacian pyramid, shown in (c). The composite image reconstructed from the final Laplacian pyramid is shown in (d) (Kraft, 2016).

What particularly caught my eye from Kraft's draft paper was his observation regarding the differences in recognizability between reduced-energy image classes:

> *"...the recognizability of images seems to generalize over image classes: the familiarity to a human observer of the results of signal-energy reduction algorithm seems to be highly correlated with the label of the starting image. If the recognizability to human observers of the reduced-energy images is well correlated with the label of the original image then the essential features for classifying images are, for some types of objects, likely similar to the features that human visual systems depend on. In contrast, DCNNs classify other object classes using features dissimilar to the features used by humans for classification. This strategy could be a result of the standard rules of image classification competitions in which an image is known to contain one of a set number of object types, so robust classification of some object classes is advantageous, because after ruling out some of the robustly detected object classes, the remaining can be discriminated based on more specialized features."*

This observation suggests that DCNNs employ two qualitatively distinct strategies in detecting object types. If an object is easily recognizable by humans from its reduced-energy image, this implies that DCNNs detect the object based on the presence of features according to an appearance model that roughly corresponds to a human observer's appearance model of the same object. Conversely, if an object is unrecognizable to a human from its reduced-energy image, this implies that DCNNs are employing a discriminative strategy in which they first rule out certain high-level features of the object, and then apply a specialized mode of detection that is only sensitive to features useful for detection after ruling out other possible categories. Nguyen et al. reached a similar conclusion, elaborating on the flaws in discriminative versus generative models:

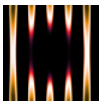
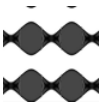
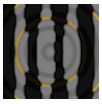
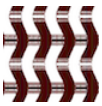
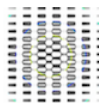
> *"Discriminative models create decision boundaries that partition data into classification regions. The further from the decision boundary an image is, even if far away from the natural images in the class, the higher the confidence. At a minimum, in a high-dimensional space, the area allocated by a discriminative model to a class could be very large, such that a DNN can be highly confident that a synthetic image is in that area without that image being close to the natural images in that area."*

With these results, I could begin to design my own experiment to further tease out differences between the way DCNNs and humans recognize objects. I chose to use the pre-trained Inception-v3 model trained for the ImageNet Large Visual Recognition Challenge using the data from 2012. Inception-v3 achieves a 3.46% top-5 error rate. To constrain my exploration

(as well as to enable training in a reasonable amount of time), I decided to select 8 classes of indirectly encoded fooling images given by Nguyen et al., half with recognizable fooling images, and half with fooling images indecipherable to humans. My goal was to determine if differences between the representations of the eight object types used by DCNNs and those used by humans could be deciphered by analyzing the reduced-energy minimal images for both the original image examples, and the corresponding fooling images. Specifically, I took chose ten example images from the ImageNet dataset (http://imagenet.stanford.edu/about-overview) (detected with ≥ 98% confidence by the DCNN) and the one fooling image from Nguyen et al. for each of the eight object types. I planned to use these 88 images as the source material from which to generate reduced-energy minimal images.

Code	Label	Fooling Image Human Recognizable
n03000134	chainlink fence	Yes
n03729826	matchstick	Yes
n03942813	ping-pong ball	Yes
n04356056	sunglasses, dark glasses, shades	Yes
n02056570	king penguin, Aptenodytes patagonica	No
n02317335	starfish, sea star	No
n03272010	electric guitar	No
n04074963	remote control, remote	No

I decided to use TensorFlow's implementation (https://www.tensorflow.org/versions/r0.8/tutorials/image_recognition/index.html) of the Inception-v3 model to test image recognition confidence, hoping to confirm that the DCNN would classify both the training images from ImageNet and the fooling image from Nguyen et al. with a high confidence. I ran into my first problem because suprisingly to me the Inception-v3 model was not as easily fooled, or at least not by Nguyen's fooling images. After my first try failed, I next attempted to use an AlexNet implementation (https://github.com/guerzh/tf_weights) in TensorFlow (I used TensorFlow throughout the project because my computer was having a ton of difficulty with Caffe installation). The AlexNet TensorFlow model performed similarly to the results in Nguyen et al. for some of the input images, and drastically different for others.

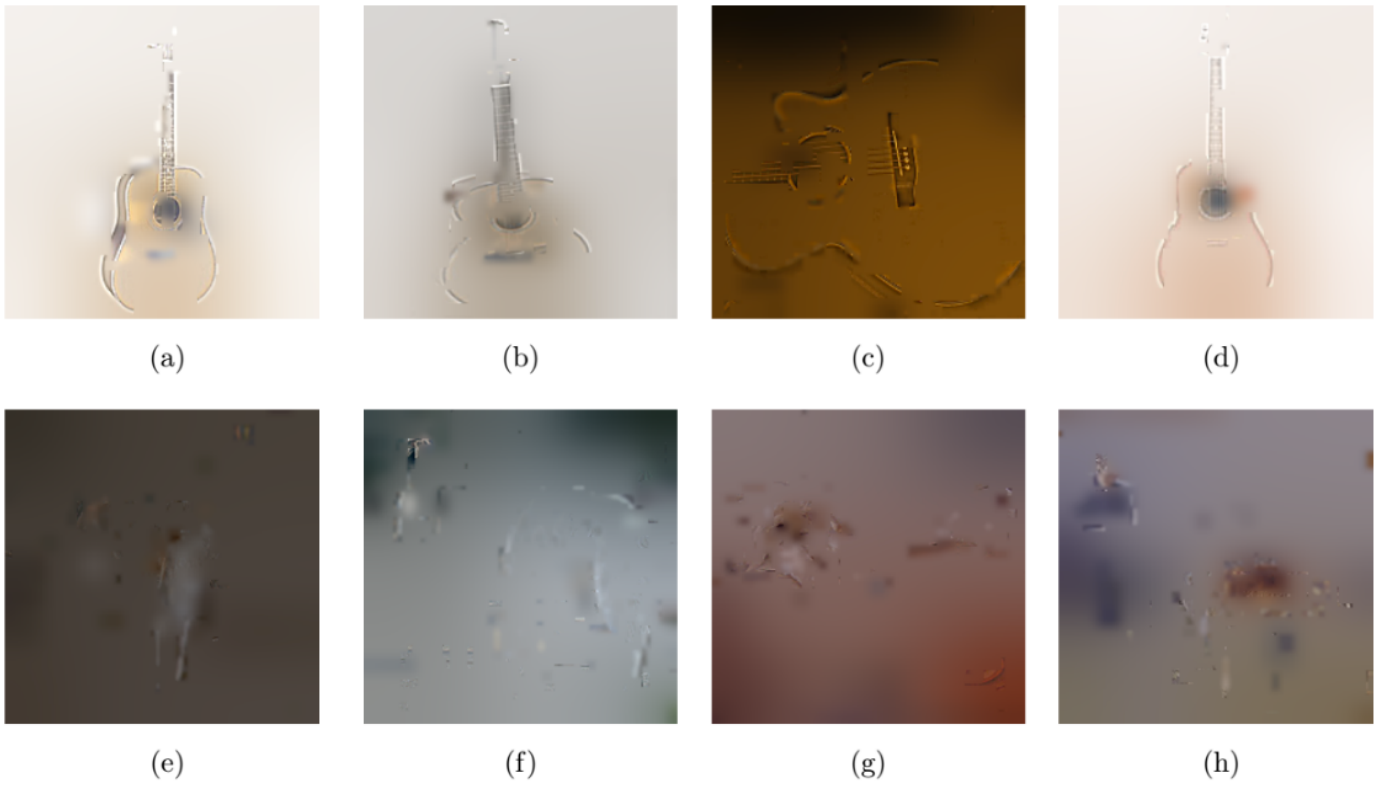
Image	Nguyen et al. Classification (AlexNet)	Tensor Flow (Inception-v3)	Tensor Flow (AlexNet)
	chainlink fence (score ≥ 98%)	fire screen, fireguard (score = 96.0%)	chainlink fence (score = 90.2%)
	matchstick (score ≥ 98%)	space heater (score = 44.9%)	chime, bell, gong (score = 8.3%)
	ping-pong ball (score ≥ 98%)	cassette (score = 60.0%)	ping-pong ball (score = 93.2%)
	sunglasses, dark glasses, shades (score ≥ 98%)	sunglass (score = 41.8%)	sunglasses (score = 73.0%)
	king penguin, Aptenodytes patagonica (score ≥ 98%)	knee pad (score = 79.3%)	Windsor tie (score = 8.6%)
	starfish, sea star (score ≥ 98%)	stole (score = 6.4%)	handkerchief, hankie, hanky, hankey (score = 49.6%)
	electric guitar (score ≥ 98%)	plate rack (score = 40.6%)	knee pad (score = 33.6%)
	remote control, remote (score ≥ 98%)	modem (score = 36.8%)	remote control, remote (score = 100.0%)

The results made me reconsider my approach, and I decided to focus on just the highest confidence human-recognizable fooling image (ping-pong ball), and the highest confidence non huuman-recognizable image (remote control, remote). I again ran into issues when it came to implementing a signal-reduction algorithm to use in generating minimal energy ping-pong ball and remote control images. Generating the Laplacian pyramids for the two images was easy using OpenCV (http://opencv-python-

tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_pyramids/py_pyramids.html), but I got tripped up trying to figure out how to choose the correct size for the erasure window.

As Kraft points out in his paper, without constraints on the size of the erasure window used by the randomized signal-energy reduction algorithm, it is possible for the algorithm to create artifacts in the image by selectively deleting small regions within the pyramid layers. Furthermore, it is easy for the signal-reduction algorithm to create new features in the image by essentially painting on the image with the erasure window. This could result in the algorithm creating minimal images by generating new, incidental features, rather than eliminating everything but the essential features. Constraining the erasure window size also mitigates this effect. Kraft selected a window size by requiring the algorithm to produce a number of unsuccessful candidate images via deletions of a certain range of window sizes before moving on the to the next smallest window-size range. He combined this approach with a fitness function that did not reward increases in detection confidence beyond that of the starting image. After attempting to replicate this technique, I was still unable to successfully extract the salient features from the remote and ping-pong ball images.

I made yet another pivot, and without a signal-reduction algorithm, decided to utilize images from the Kraft paper. What caught my eye in particular were the minimial energy images for the acoustic guitar and borzoi, Russian Wolfhound classes. The photos are representative samples of minimum-energy images for the two classes.



Each image in both sets is recognized by Kraft's DCNN with greater than 98% confidence, yet to a human only the acoustic guitar images are recognizable. Interestingly, Nguyen et al. identified a trend when attempting to generate fooling images that could help elucidate the difference. Nguyen et al. found that it was harder to produce high-confidence fooling images for classes overrepresented in the ImageNet datatset (such as the dog synset), offering two potential explanations:

1. The network is tuned to identify many specific types of dogs so it has more units dedicated to this image type than others (i.e. it is less overfit so therefore harder to fool)

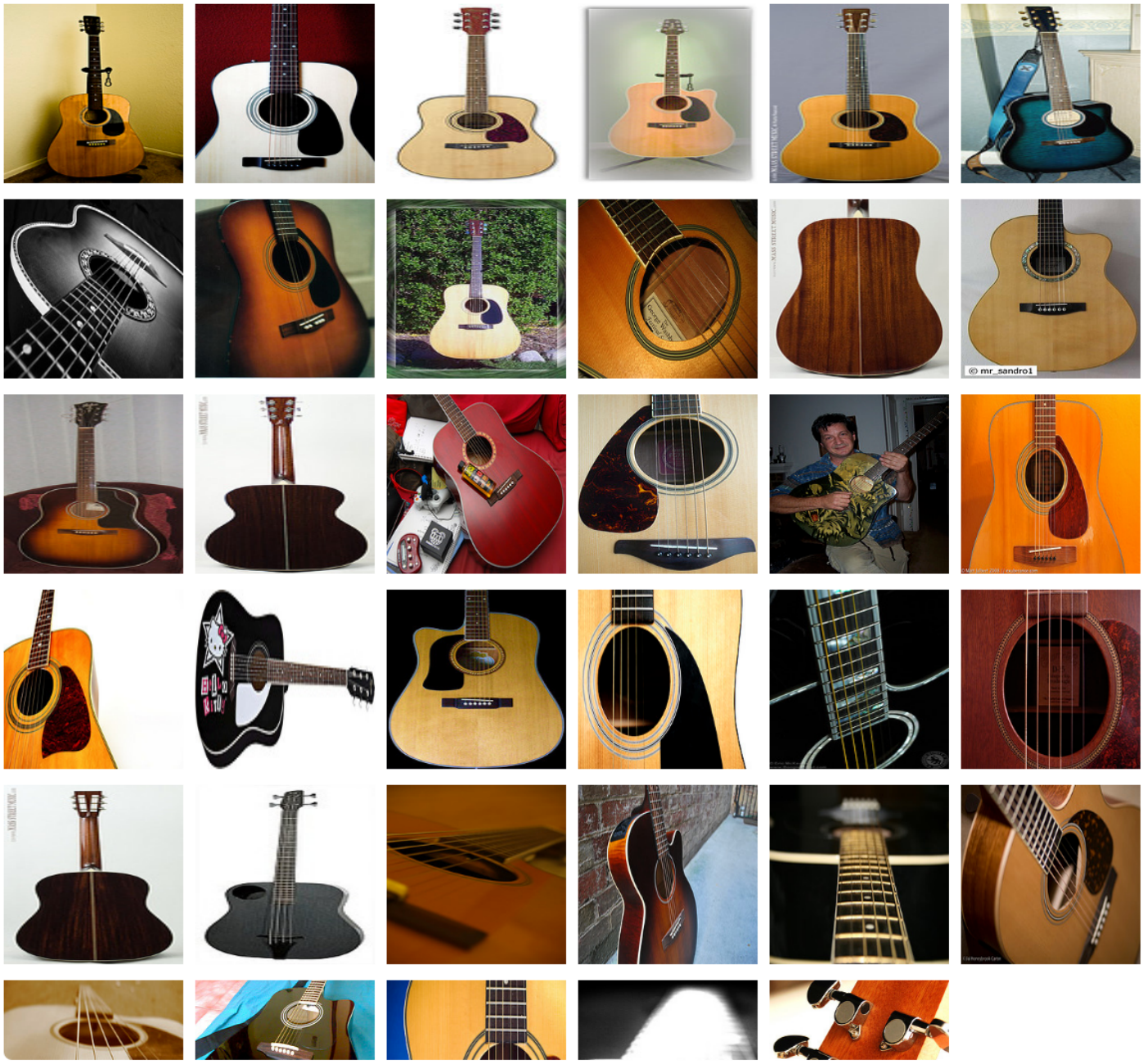
2. There are so many dog classes that the EA has trouble finding an image that scores high in a specific category but low in related categories

This line of reasoning could explain why the minimal-energy images for borzoi are not human recognizable as the DCNN is likely relying on highly specialized features to differentiate different breeds of dogs. The dog, domestic dog, Canis familiaris sysnet contains 189 classes, the subset hunting dog contains 101, the smaller subset hound dog still has 29 classes, and even within the even more specific hounddog grouping, there is both the borzoi, Russian wolfhound and Irish wolfhound.

A subset of images for borzoi, Russian wolfhound in ImageNet.

A subset of images for Irish wolfhound in ImageNet.

As a human observer, I find it difficult to distinguish between these two classes. On the other hand, there are 163 musical instrument synsets, but only 48 stringed instrument synsets, and just 6 guitar types: acoustic guitar, bass guitar, cittern, electric guitar, Hawaiian guitar, and ukulele. These classes contain much more distinct types

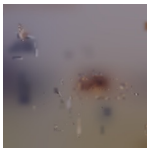


A subset of images for acoustic guitar in ImageNet.

Out of curiousity, I ran the minimal-energy images for the borzoi, Russian wolfhound class through the TensorFlow AlexNet model. The AlexNet implementaion produced results that overwhelmingly fell under the dog synset, whereas the Inception-v3 model has completely different identifications.

Image	Tensor Flow (AlexNet)	Tensor Flow (Inception-v3)
	1. Arctic fox, white fox, Alopex	1. barn spider, Araneus

	lagopus 0.128337 2. white wolf, Arctic wolf, Canis lupus tundrarum 0.11411 3. **borzoi, Russian wolfhound 0.0553836** 4. ice bear, polar bear, Ursus Maritimus, Thalarctos maritimus 0.0475548 5. Samoyed, Samoyede 0.0361909	cavaticus 0.25208 2. nematode, nematode worm, roundworm 0.20965 3. isopod 0.02492 4. tick 0.02034 5. ant, emmet, pismire 0.01542
	1. tub, vat 0.0637832 2. shower curtain 0.062885 3. bathtub, bathing tub, bath, tub 0.0543703 4. nematode, nematode worm, roundworm 0.0378981 5. beaker 0.0377563	1. nematode, nematode worm, roundworm 0.32972 2. barn spider, Araneus cavaticus 0.19129 3. isopod 0.06626 4. Petri dish 0.03621 5. common newt, Triturus vulgaris 0.03092
	1. ice bear, polar bear, Ursus Maritimus, Thalarctos maritimus 0.208913 2. Arctic fox, white fox, Alopex lagopus 0.166923 3. great white shark, white shark, man-eater, man-eating shark, Carcharodon carcharias 0.0965338 4. hammerhead, hammerhead shark 0.0665496 5. stingray 0.0581107	1. candle, taper, wax light (score = 0.50401) 2. consomme 0.06374 3. eggnog 0.04933 4. West Highland white terrier 0.04152 5. seashore, coast, seacoast, sea-coast 0.01940

	1. axolotl, mud puppy, Ambystoma mexicanum 0.0520174 2. tick 0.0315102 3. barn spider, Araneus cavaticus 0.0313211 4. ant, emmet, pismire 0.02819 5. isopod 0.0258127	1. consomme (score = 0.09062) 2. Petri dish (score = 0.08425) 3. isopod (score = 0.06651) 4. nematode, nematode worm, roundworm (score = 0.06197) 5. goldfish, Carassius auratus (score = 0.04198)

The results for the Inception-v3 model were so far removed from Kraft's classifications, that for the acoustic guitar class I focused on just the classficiations from the AlexNet model, which placed acoustic guitar in the top 5 for all four pictures.

Image	Tensor Flow (AlexNet)
	1. **acoustic guitar 0.544734** 2. hook, claw 0.0659736 3. plunger, plumber's helper 0.0309352 4. banjo 0.0278781 5. electric guitar 0.0257582
	1. plunger, plumber's helper 0.467679 2. **acoustic guitar 0.109538** 3. switch, electric switch, electrical switch 0.0274923

	4. ladle 0.025571 5. electric guitar 0.0218776
	1. **acoustic guitar 0.449543** 2. electric guitar 0.155197 3. joystick 0.146449 4. plunger, plumber's helper 0.0944864 5. soap dispenser 0.0224255
	1. electric guitar 0.0864191 2. beaker 0.0752944 3. nematode, nematode worm, roundworm 0.0469728 4. mouse, computer mouse 0.045615 5. **acoustic guitar 0.0439272**

These results seem to suggest that contrary to Nguyen et al.'s assertion that different DCNNs learn the same discriminative features for each class, it would appear that the AlexNet model and the Inception-v3 model in fact do learn different features, and images that fool an AlexNet model often will not also fool an Inception-v3 model. In summary, I don't think my analysis was thorough or quantitative enough to provide any substantial additional insights into my two focus question, but I was able to gain a better understanding of how some state-of-the-art current models perform classification.

1. **Do the identification mechanisms used by human observers and DCNNs converge at a certain level of abstraction, or for a specific image class?**

2. **Is there a pattern in the underlying structure of images in certain image classes that makes it easier to generate fooling images for that class?**

For some image classes, it seems that humans and DCNNs use very similar representations for image identification. These classes are typically identified by two primary traits: (1) representative minimial-energy images for the class are human recognizable, and (2) fooling

images are relatively easy to generate for the class. On the other hand, likely as a result of the discriminative nature of current DCNNs, neural nets recognize images in overrepresented synsets using highly specialized features that are only useful after other image classes are ruled out. Often in these cases, no matter the level of abstraction human observers and DCNNs do not converge in their identification mechanisms (this still leaves open the question of how exactly humans are able to discriminate between highly similar classes, such as Russian wolfhound and Irish wolfhound).

Appendix

- Inceptionism: Going Deeper into Neural Networks (http://googleresearch.blogspot.com/2015/06/inceptionism-going-deeper-into-neural.html), *Mordvintsev, Olah, and Tyka (2015)*

- Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images (http://arxiv.org/pdf/1412.1897v4.pdf), *Nguyen et al. (2015)*

- Understanding Deep Image Representations by Inverting Them (http://arxiv.org/pdf/1412.0035v1.pdf), *Mahendran and Vedaldi (2014)*

- Inverting Visual Representations with Convolutional Networks (http://arxiv.org/pdf/1506.02753.pdf), *Dosovitskiy and Brox (2016)*

- Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps (http://arxiv.org/pdf/1312.6034v2.pdf), *Simonyan, Vedaldi, and Zisserman (2014)*

- Object Recognition with Informative Features and Linear Classification (https://www.eecs.berkeley.edu/%7Eefros/courses/AP06/Papers/vidal-iccv-03.pdf), *Vidal-Naquet and Ullman (2003)*

- Exploring DCNN Features via Signal-Energy Reduction, *Kraft (2016)*