Investigations on Improving Classification Accuracy on Cubic Paintings

Chen Liu

1 Overview

Our goal is to increase face classification accuracy on cubic paintings with distorted faces, which existing models don't do well on. For benchmarking, I collected Picasso's cubic paintings from WikiArt website, picked those with faces in them, and split them into two types: Large Cubic Faces paintings and Small Cubic Faces paintings.

Large Cubic Faces dataset has 56 paintings in total, all of which have faces occupying a significant portion of the painting, so it's easier for our model to classify it as a "face" image.

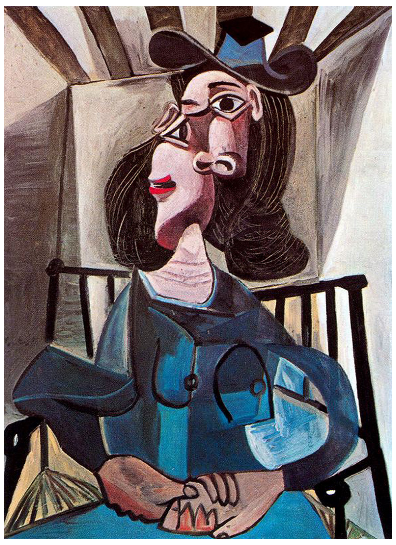

Figure 1: Example of a painting in Large Cubic Faces dataset

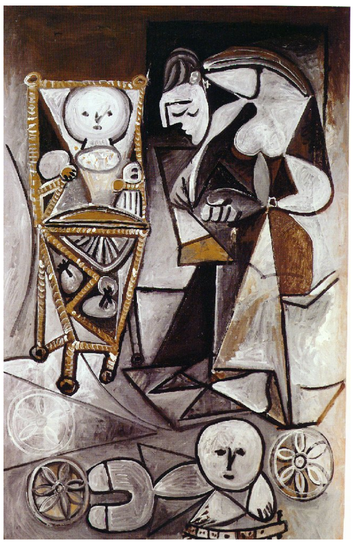

Figure 2: Example of a painting in Small Cubic Faces dataset

Small Cubic Faces dataset has 41 paintings in total. Faces in them occupy a small region in the painting, and there could also be multiple small faces, combined with many other objects. It's more difficult for the model classify these kind of paintings as "face".

In addition to cubic paintings dataset, I also collected 40,000 images from ImageNet for finetuning models (which will be referred to as "ImageNet Training Set"), and another 10,000 images from ImageNet as a test set ("ImageNet Test Set").

2 Original AlexNet Model

AlexNet model trained on ImageNet (ILSVRC 2012) dataset doesn't have a face class. The closest class is "mask".

Figure 3: A "mask" class image in ImageNet

Accuracy of the original AlexNet:

	Top 1	**Top 5**
Large Cubic Faces (classified as ”mask”)	5.72%	24.56%
Small Cubic Faces (classified as ”mask”)	0%	2.64%
ImageNet Test Set	62%	88%

3 AlexNet Finetuned with Face Images

3.1 Method

To better measure classification accuracy, we need to have an actual ”face” class. To achieve this, I finetuned AlexNet by turning the #643 class, ”mask”, into a ”face” class.

Among the 5 convolution layers of AlexNet, I fixed the first three convolution layers (conv1, conv2, conv3) by having a learning rate of 0 for these layers.

I downloaded 40,000 face photos from Flickr, combined with 11,878 images from the Labeled Faces in the Wild dataset. Another 1,355 images from the Labeled Faces in the Wild dataset are used for testing. These images are all labeled with the #643 class.

To balance face images with non-face images, I also included the 40,000 ImageNet Training Set images with the ”mask” images removed for finetuning.

3.2 Results

Classification accuracy with AlexNet model finetuned with **Flickr** and **ImageNet Training Set** images:

	Top 1	**Top 5**
Large Cubic Faces	28.32%	41.56%
Small Cubic Faces	0%	7.92%
ImageNet Test Set	52.52%	75.72%
Labeled Faces in the Wild Test Set	73.4%	97.64%

Classification accuracy with AlexNet model finetuned with **Flickr, Labeled Faces in the Wild, and ImageNet Training Set** images:

	Top 1	**Top 5**
Large Cubic Faces	18.92%	32.12%
Small Cubic Faces	0%	2.64%
ImageNet Test Set	53.8%	76.48%

After finetuning with Flickr face images, I was able to achieve higher classification accuracy on LFW Test Set (73.4%) while causing accuracy on ImageNet Test Set to drop by 10%, from 62% to 52%. It also becomes better at classifying Large Cubic Faces with an accuracy of 28.32%, although with this accuracy it means the model still fails to recognize most of cubic face images.

However, after adding Labeled Faces in the Wild dataset in the training set, accuracy on Large Cubic Faces drops from 28% to 19%.

4 AlexNet Finetuned with Jittered Face Images

4.1 Method

One hypothesis for the reason why finetuned AlexNet fails to recognize faces in cubic paintings is that it's not flexible enough. Due to the nature of cubic paintings, the relative locations of facial elements, such as eyes and noses, are not as fixed as natural faces. To increase flexibility of the model, we can finetune the model using face images with ”jittered” facial elements.

To be able to jitter facial elements, we first need a facial elements detector. This is done through training a classifier that tell us whether an image patch contains an eye, a nose, or a mouth by classifying the hypercolumn corresponding to that patch in the image in a certain layer (such as conv3_2) in VGG into yes/no classes. There are mainly steps:

1. Feed face images into VGG network and randomly sample a certain percentage of hypercolumns from one layer (e.g. conv3_2);
2. Use KMeans to cluster these hypercolumns into a number of clusters (such as 32);
3. Pick clusters that are eyes, noses and mouths by viewing the image patches those hypercolumns correspond to;
4. Given a new test image, go through every hypercolumn in the previously selected layer, and use KMeans to predict a cluster for the hypercolumn.

With a facial element classifier, we find all patches with eyes in them, and randomly move the patch by a given radius (e.g. 20px) to a surrounding location. The resulted image is shown below.

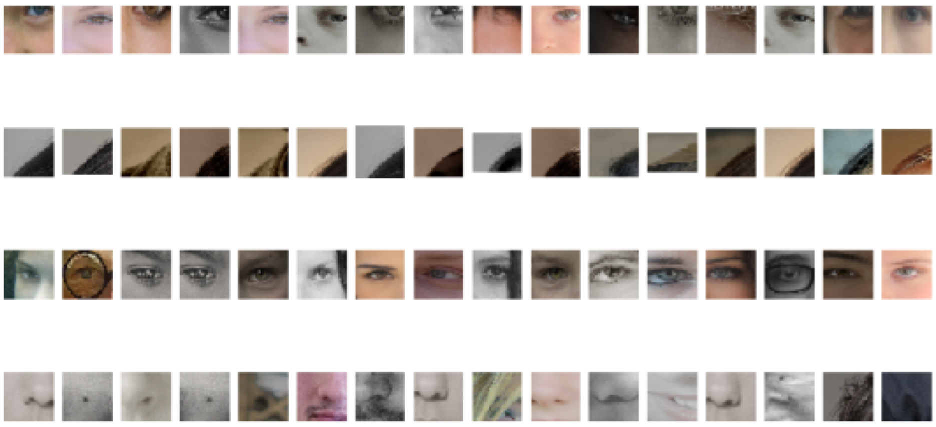

Figure 4: Some clusters from VGG16's conv3_2 layer. Each row corresponds to one cluster.

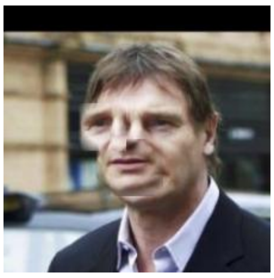

Figure 5: A face image jittered by 15 pixels

4.2 Results

Accuracy of AlexNet finetuned using Flickr dataset with facial elements jittered by 5 pixels (facial elements classifier is trained using VGG's conv3_2 layer hypercolumns):

	Top 1	**Top 5**
Large Cubic Faces	3.76%	5.64%
Small Cubic Faces	0%	0%
ImageNet Test Set	53.4%	75.9%
Labeled Faces in the Wild Test Set	100%	100%

Accuracy of AlexNet finetuned using Labeled Faces in the Wild dataset with facial elements jittered by 15 pixels (facial elements classifier is trained using VGG's conv3_3 layer hypercolumns):

	Top 1	**Top 5**
Large Cubic Faces	0%	0%
Small Cubic Faces	0%	0%
ImageNet Test Set	54%	76.5%

Finetuning face images with jittered facial elements does not appear to help with Large Cubic Faces classification accuracy. However, what is interesting is that by using jittered Flickr images for training, we easily achieved 100% accuracy in Labeled Faces in the Wild Test Set classification, improved from 73.4% from the results in the previous section, with only very small impact on classification accuracy on ImageNet Test Set. It appears that by using jittered face images to make the model more flexible, we are able to recognize face images more easily, instead of being better at recognizing cubic face paintings.

4.3 Analysis of Neural Response

I plotted responses of the "face" neuron in AlexNet's fully connected layer, and it seems that, although after finetuning AlexNet with face images, the face neuron is better at telling face images apart from other types of images, it's still not associating cubic face paintings with face photos:

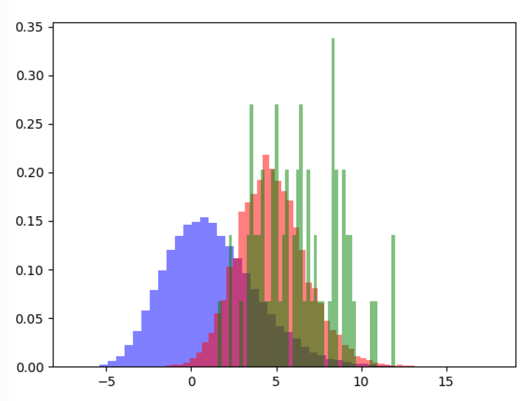

Figure 6: "Mask" neuron in original AlexNet response to face images, ImageNet images and cubic paintings. Blue: ImageNet images; red: face images; green: cubic paintings. The "mask" neuron is unable to tell the three apart.

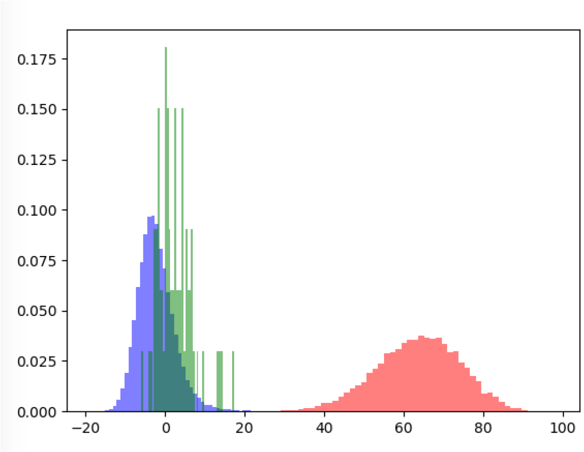

Figure 7: "Face" neuron in AlexNet finetuned with face images response to face images, ImageNet images and cubic paintings. The neuron differentiates between face images and ImageNet images, but it did not associate cubic faces with regular face images.

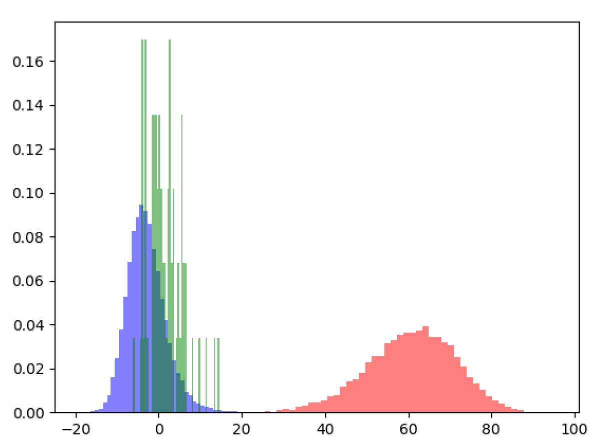

Figure 8: "Face" neuron in AlexNet finetuned with jittered face images response to face images, ImageNet images and cubic paintings

5 Sparsity of Neural Response

By clustering hypercolumns in a layer as described in previous sections, we are able to find clusters for specific facial elements, such as eyes and noses.

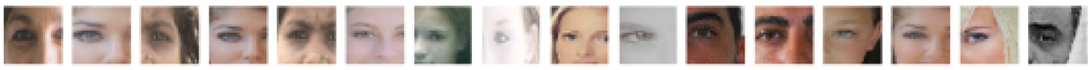

Figure 9: Using KMeans to group hypercolumns in conv4_1 layer of VGG16 into 64 clusters, the #9 cluster shown in this plot is an eye cluster

VGG16's conv4_1 layer has 512 neurons. By computing the average activation value of each neuron for all images in this cluster, we find that only a few neurons have high activation – the majority of neurons do not.

We see that Neuron #141 and #487 have the highest activations among all. If we plot stimuli response of these two neurons by going through all images in the dataset, randomly sample a fraction of patches from the image and record the neural response of these two neurons, we will find that their response is sparse as well – only a small fraction of stimuli will activate each of these two neurons:

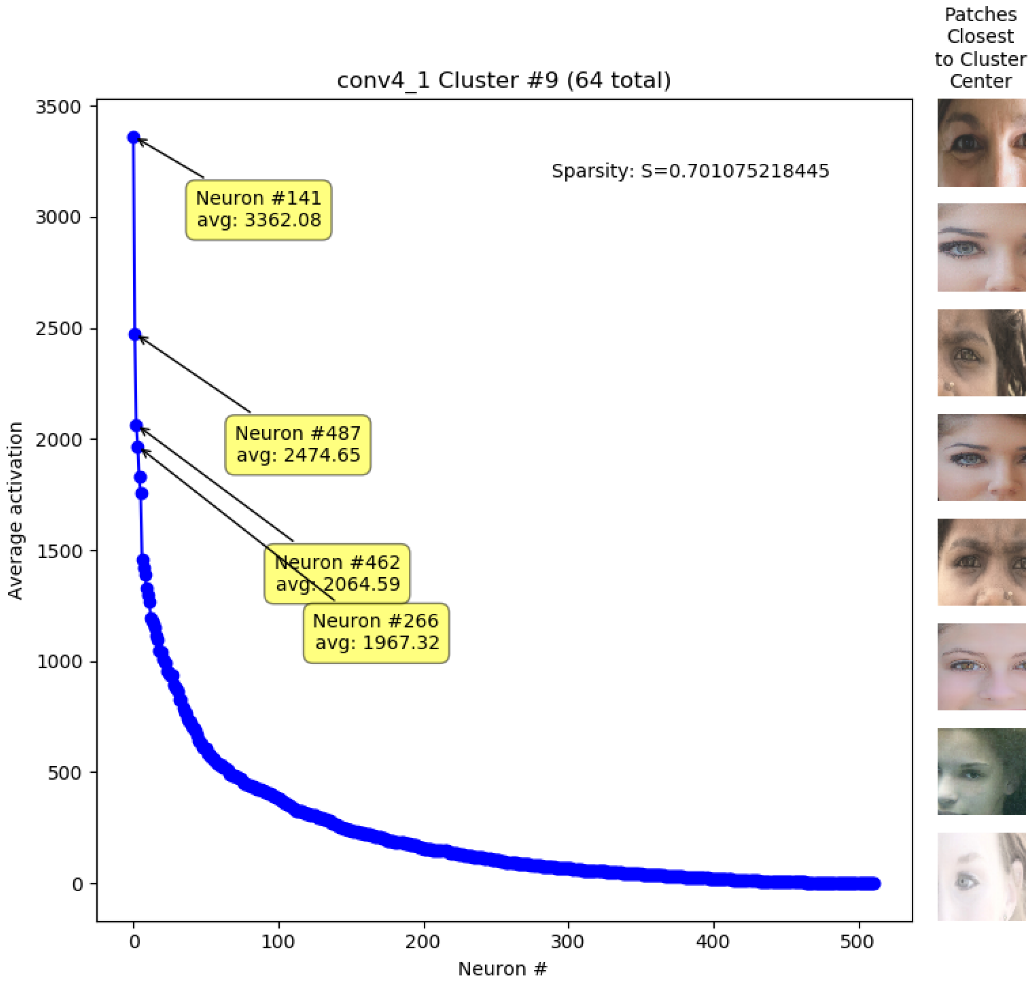

Figure 10: Average neuron activation plot of #9 cluster in VGG16's conv4_1 layer. Neurons are sorted by their average activation, from the highest to the lowest.

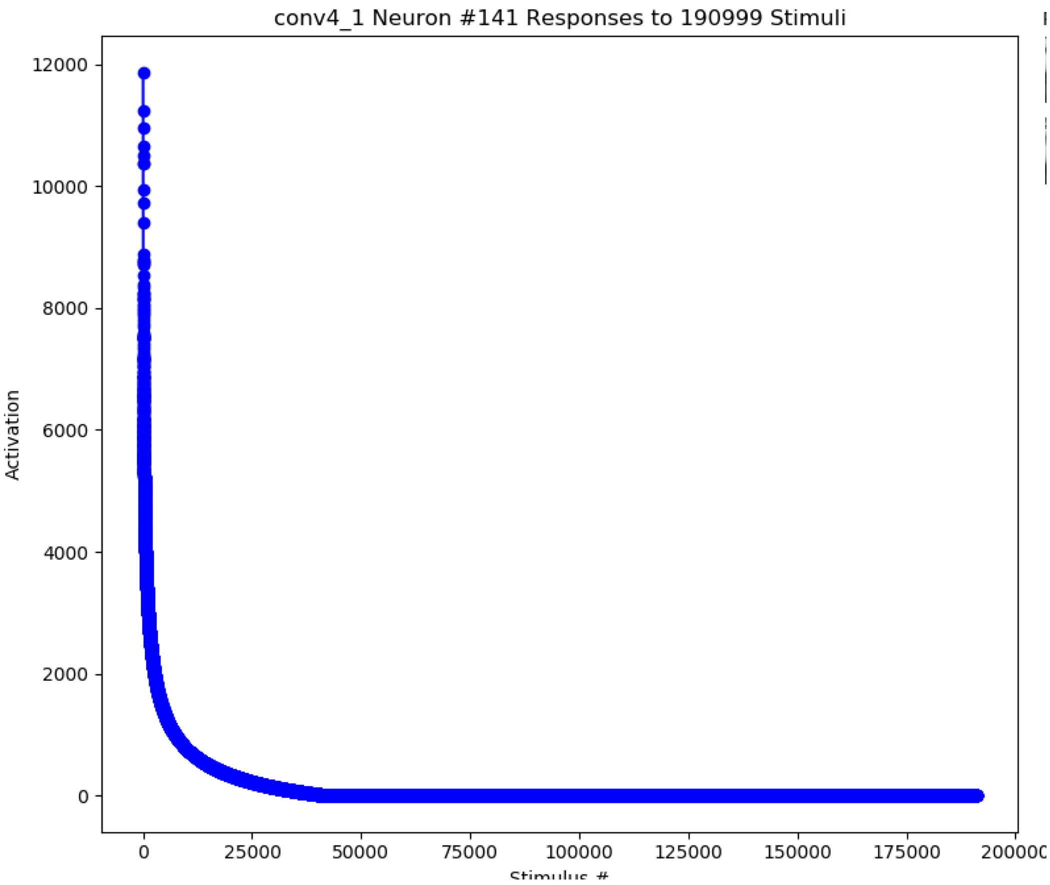

Figure 11: #141 neuron in VGG16's conv4_1 layer response to 200,000 stimuli, each of which is an image patch from the image dataset

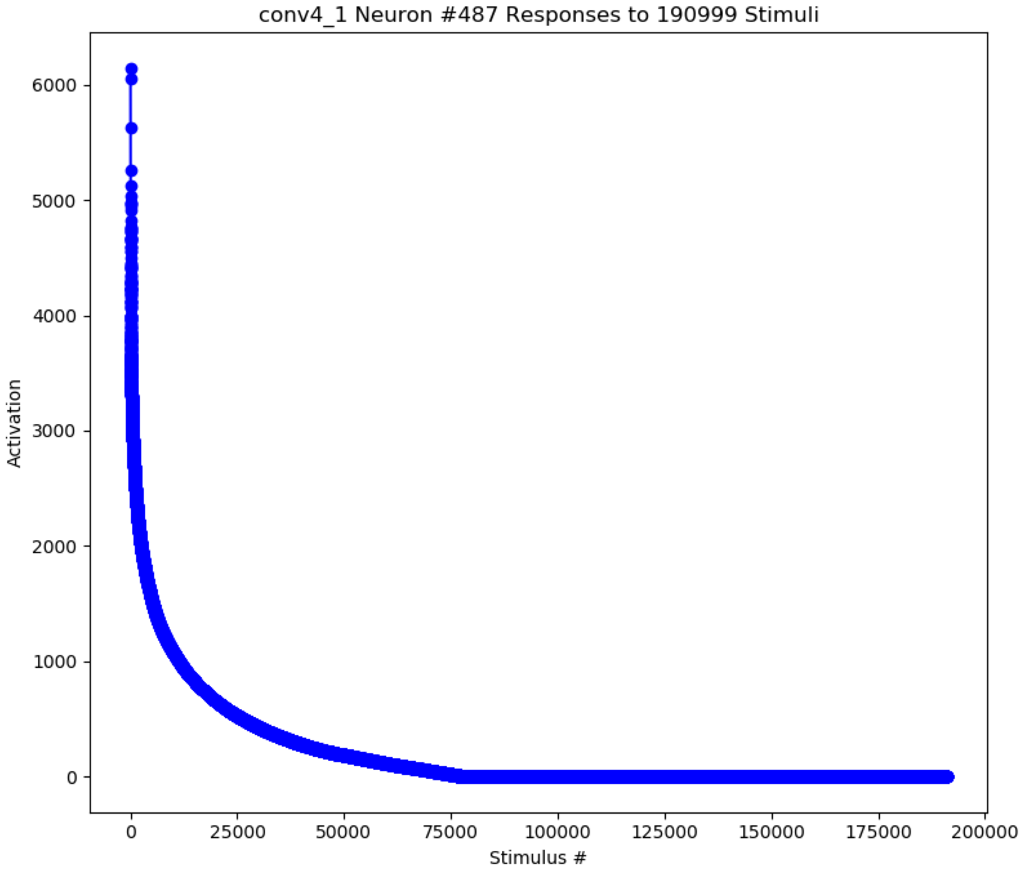

Figure 12: #487 neuron in VGG16's conv4_1 layer response to 200,000 stimuli