Lab Presentation - 2

July 10, 2017

Prerit Gupta
Summer Intern

Ravi Kiran Sarvadevabhatla
PhD Student

Unique characteristics of Sketch-a-Net Architecture

- Larger First Layer Filters
- No Local Response Normalization
- Larger Pooling Size

Index	Layer	Type	Filter Size	Filter Num	Stride	Pad	Output Size
0		Input	-	-	-	-	225×225
1	L1	Conv	15×15	64	3	0	71×71
2		ReLU	-	-	-	-	71×71
3		Maxpool	3×3	-	2	0	35×35
4	L2	Conv	5×5	128	1	0	31×31
5		ReLU	-	-	-	-	31×31
6		Maxpool	3×3	-	2	0	15×15
7	L3	Conv	3×3	256	1	1	15×15
8		ReLU	-	-	-	-	15×15
9	L4	Conv	3×3	256	1	1	15×15
10		ReLU	-	-	-	-	15×15
11	L5	Conv	3×3	256	1	1	15×15
12		ReLU	-	-	-	-	15×15
13		Maxpool	3×3	-	2	0	7×7
14	L6	Conv(=FC)	7×7	512	1	0	1×1
15		ReLU	-	-	-	-	1×1
16		Dropout (0.50)	-	-	-	-	1×1
17	L7	Conv(=FC)	1×1	512	1	0	1×1
18		ReLU	-	-	-	-	1×1
19		Dropout (0.50)	-	-	-	-	1×1
20	L8	Conv(=FC)	1×1	250	1	0	1×1

Q. Yu, Y. Yang, Y.-Z. Song, T. Xiang, and T. Hospedales. Sketch-a-net that beats humans. *BMVC*, 2015.

Deconvnet Visualization : Lower Layers

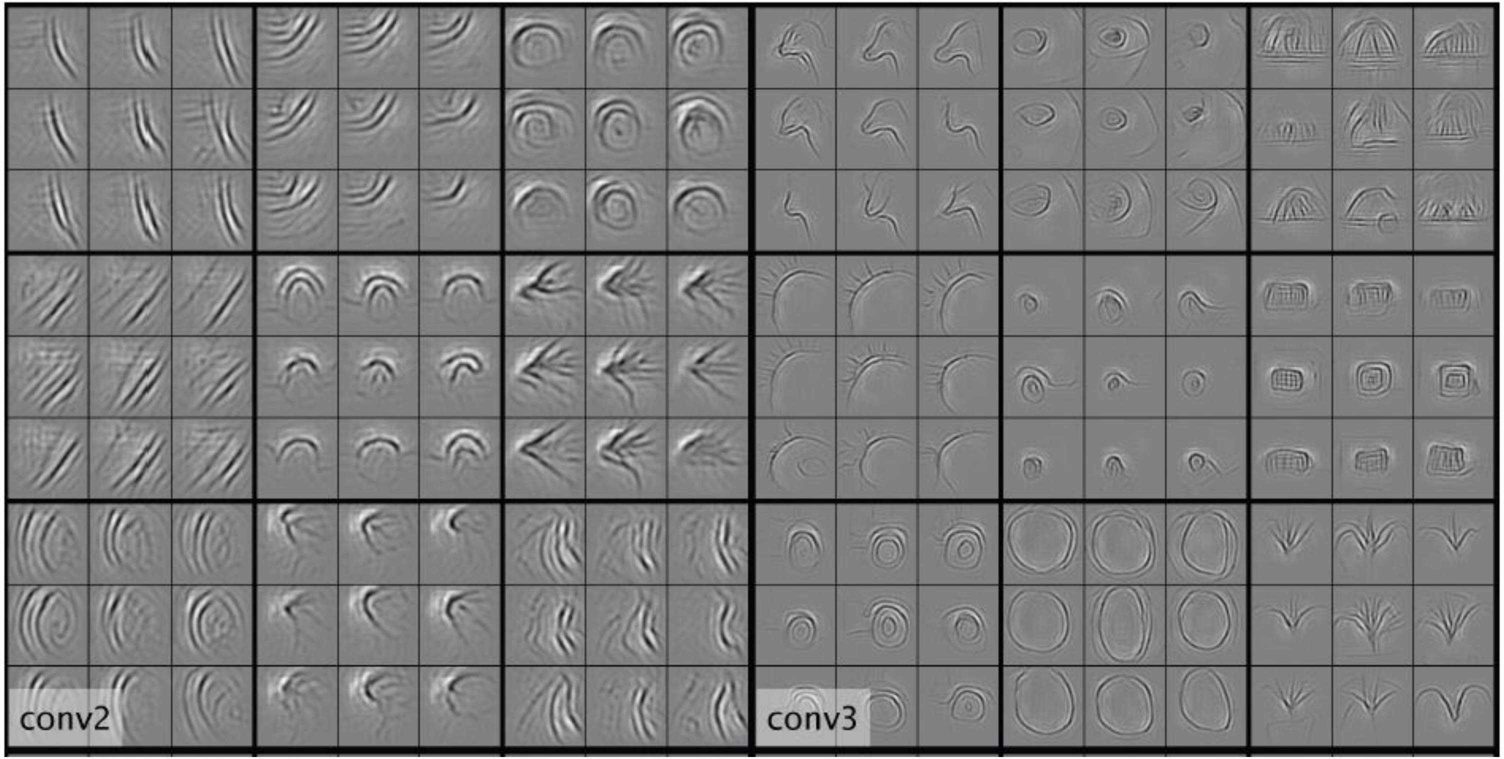


Fig. 9 Visualisation of the learned filters by deconvolution. Through visualisation of the filters by deconvolution, we can see that filter of higher-level layer are modeling more complex concepts. For example, what neurons represented in conv2 are basic building blocks to compose other concepts like lines, circles and textures; layer conv3 learns more mid-level concepts or object parts, like eye and wheel; and in conv4 and conv5, neurons are representing complex concepts like head, roof, and body.

Deconvnet Visualization : Higher Layers

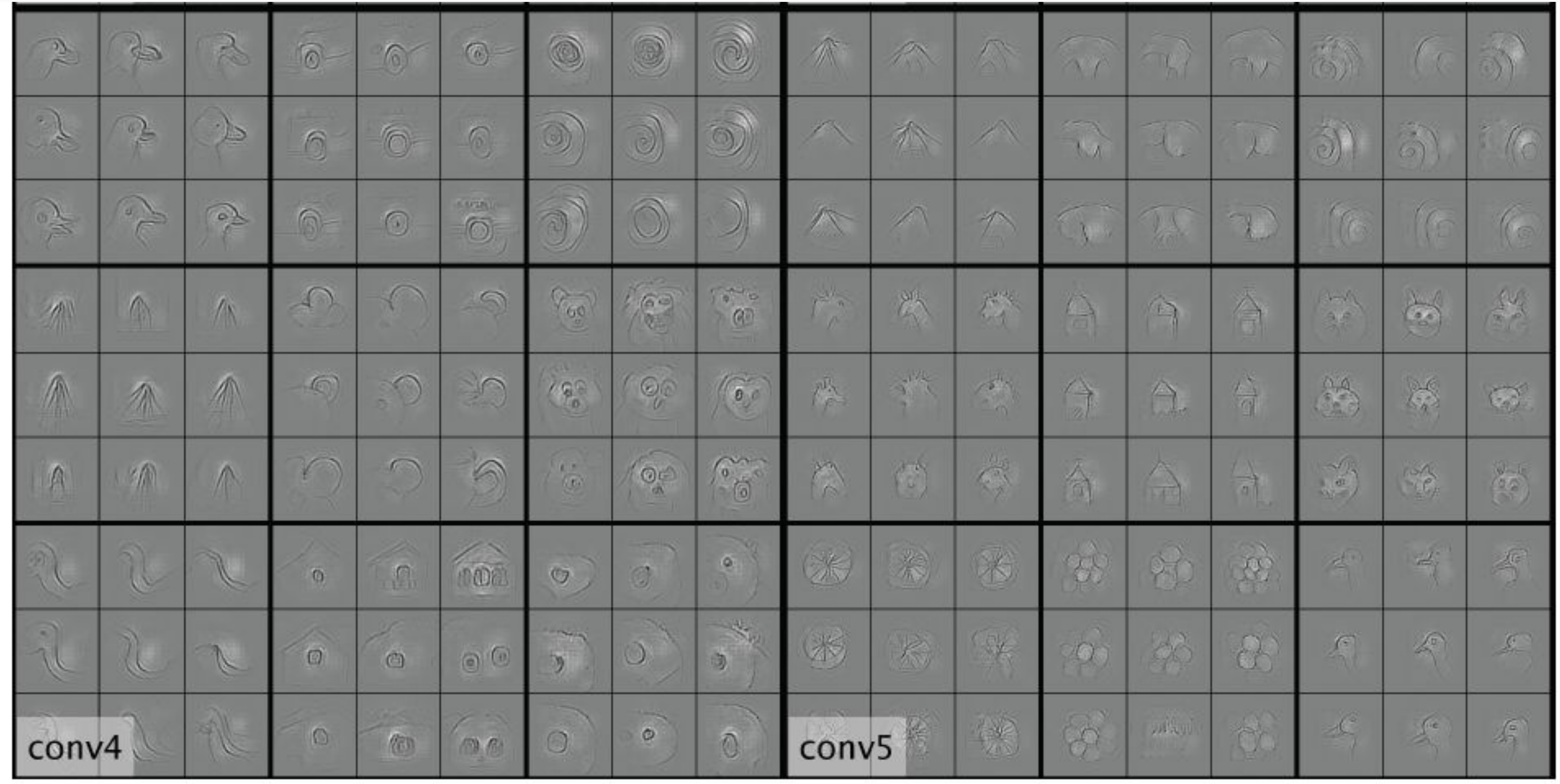


Fig. 9 Visualisation of the learned filters by deconvolution. Through visualisation of the filters by deconvolution, we can see that filter of higher-level layer are modeling more complex concepts. For example, what neurons represented in conv2 are basic building blocks to compose other concepts like lines, circles and textures; layer conv3 learns more mid-level concepts or object parts, like eye and wheel; and in conv4 and conv5, neurons are representing complex concepts like head, roof, and body.

Analysis through Deep Visualization Toolbox

- Software tool that provides a live, interactive visualization of every neuron in a trained convnet as it responds to a user-provided image or video.
- The tool displays forward activation values, preferred stimuli via gradient ascent, top images for each unit from the training set, deconv highlighting (Zeiler & Fergus, 2013) of top images, and backward diffs computed via backprop or deconv starting from arbitrary units.

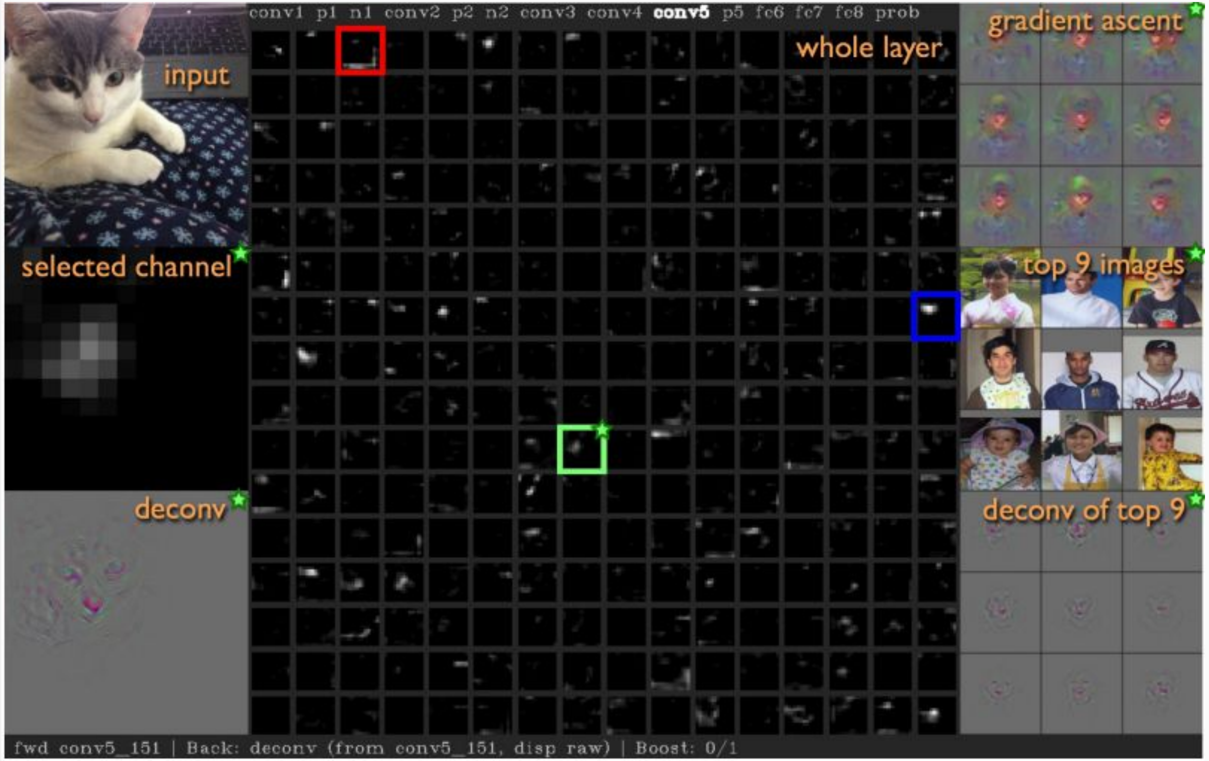
Reference: J. Yosinki, Understanding Neural Networks through Deep Visualization, ICML DL Workshop, 2015
Github Repository: https://github.com/yosinski/deep-visualization-toolbox
Link: http://yosinski.com/deepvis

Yosinki's Deep Visualization Toolbox is used to visualize different filters within selected layers.

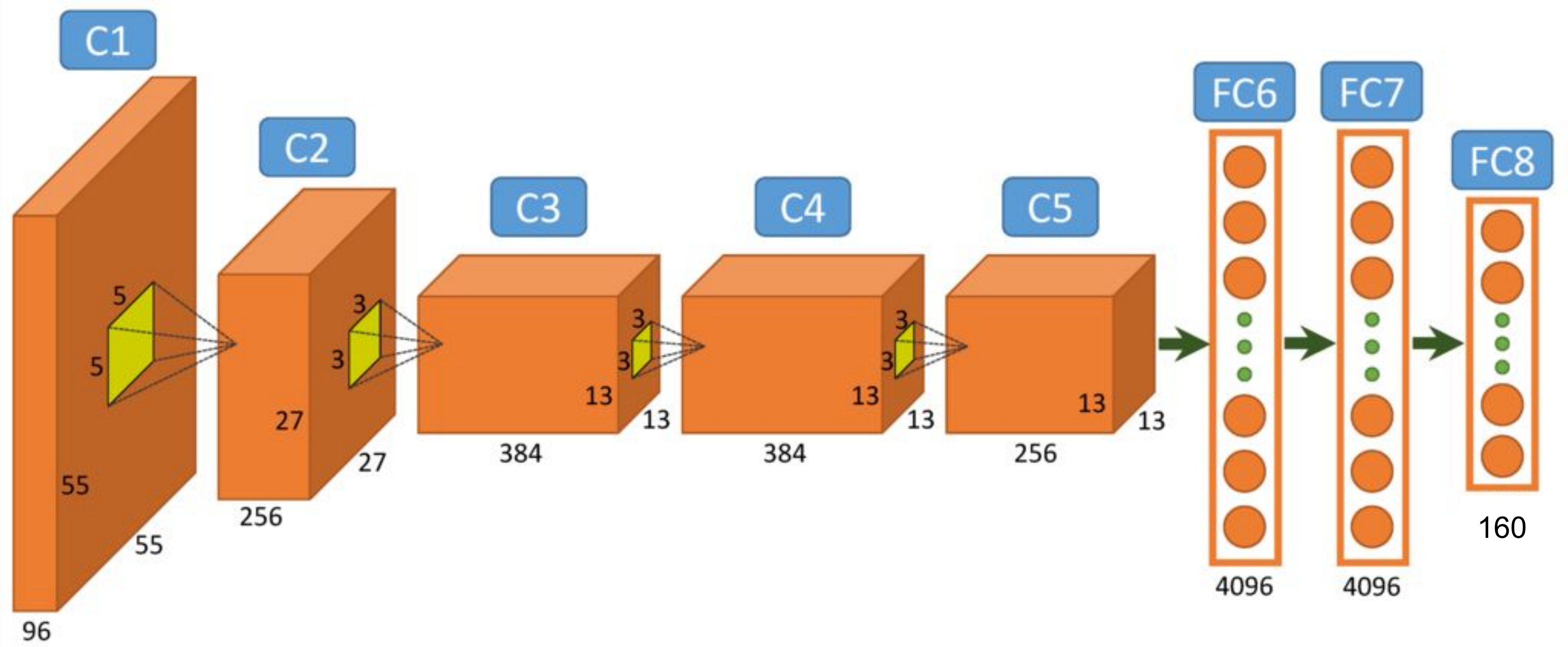
Toolbox displays:

- **Input:** Selected input image for visualization
- **Selected Channel:** Filter selected within layer
- **Deconv:** Deconvolution of the selected filter to observe for which input pixels it was activated.
- **Whole layer:** All filters within selected layer.
- **Gradient Ascent:** Synthetic image generation for maximum activation from random noise by gradient ascent
- **Top 9 images:** From validation set leading to maximum activation.
- **Deconv of top 9:** Deconvolution of the filters of top 9 images leading to maximum activation.

Deep Visualization toolbox



Fine-tuned Alexnet Architecture for 160 class output



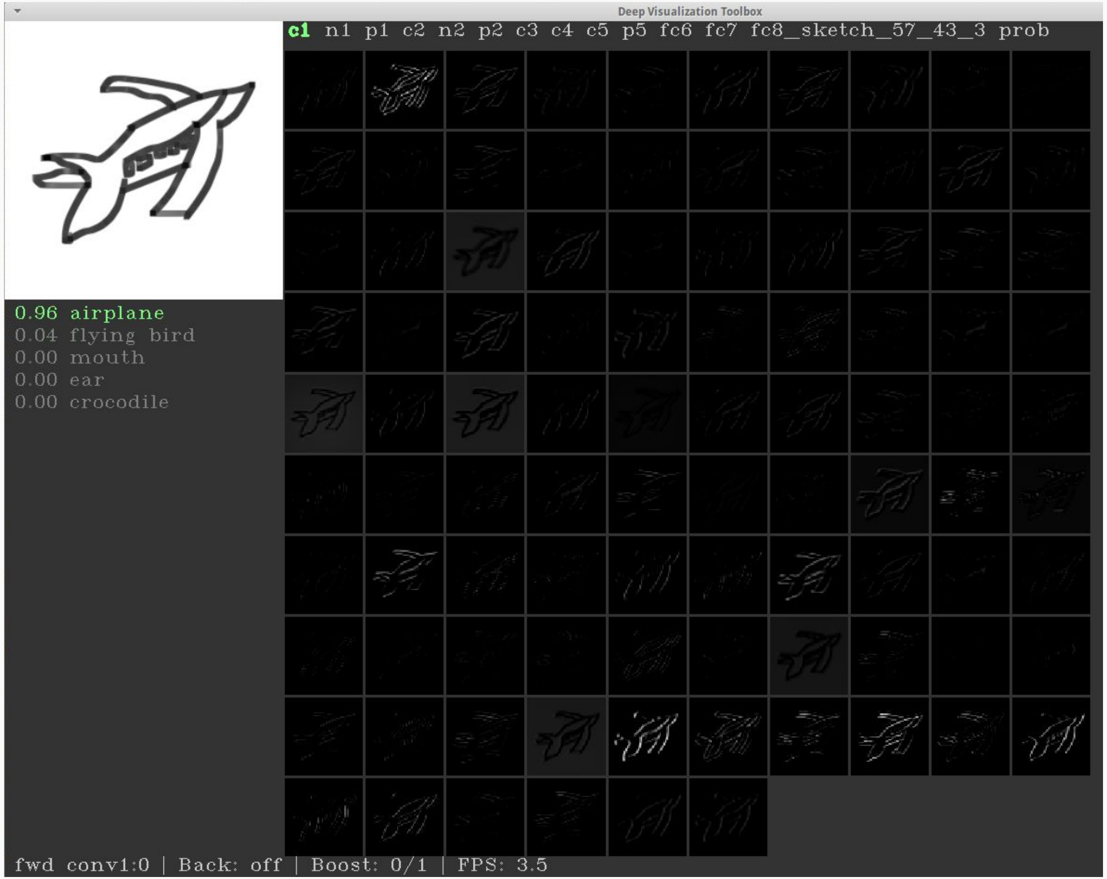
Alexnet Architecture for 160 class output is fine tuned upto 10500 iterations with final layer parameters:
lr_mult = 10 for weights
lr_mult = 20 for bias

Deep Visualization toolbox

Yosinki's Deep Visualization Toolbox is used to visualize different filters within selected layer for sketches.

Target Class: Airplane
Predicted Probabilities:

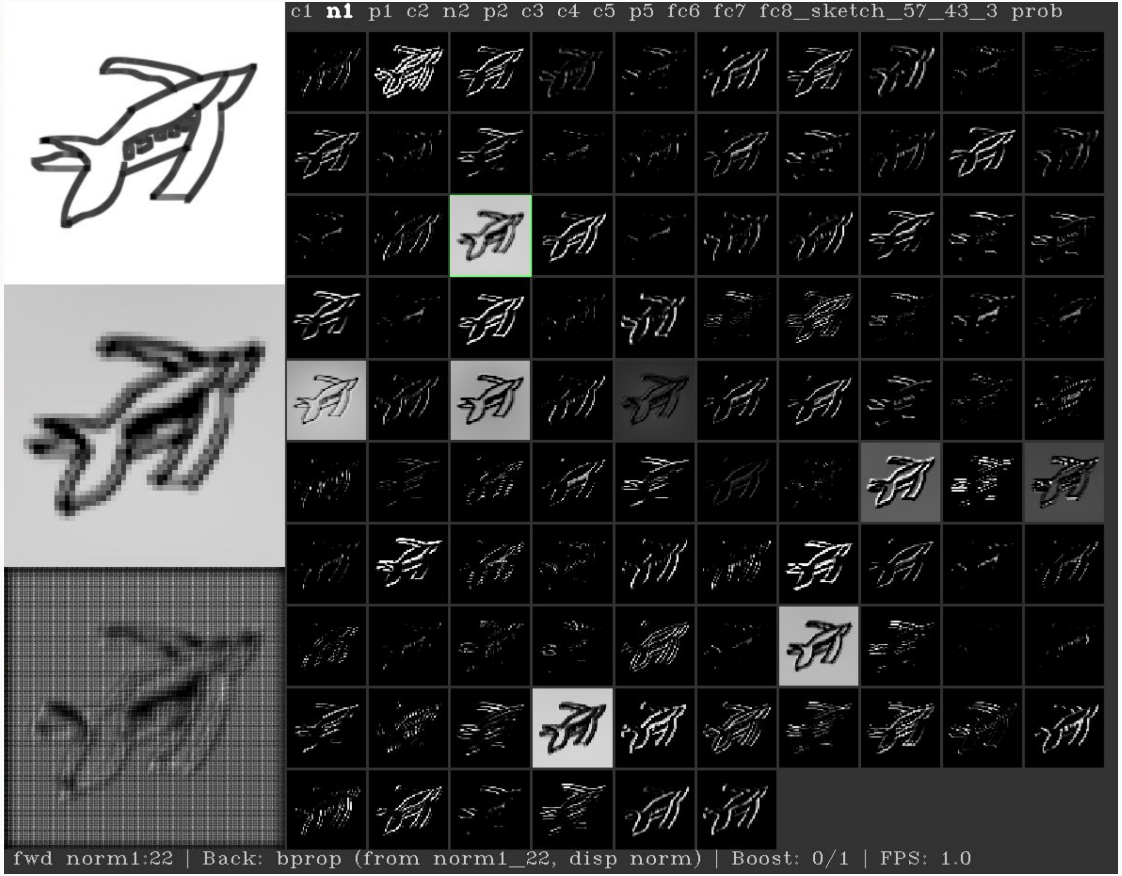
- 0.96 Airplane
- 0.04 Flying Bird



Normalized Layer 1

The Local Response Normalization (LRN) Layer after the 1st convolutional operation:

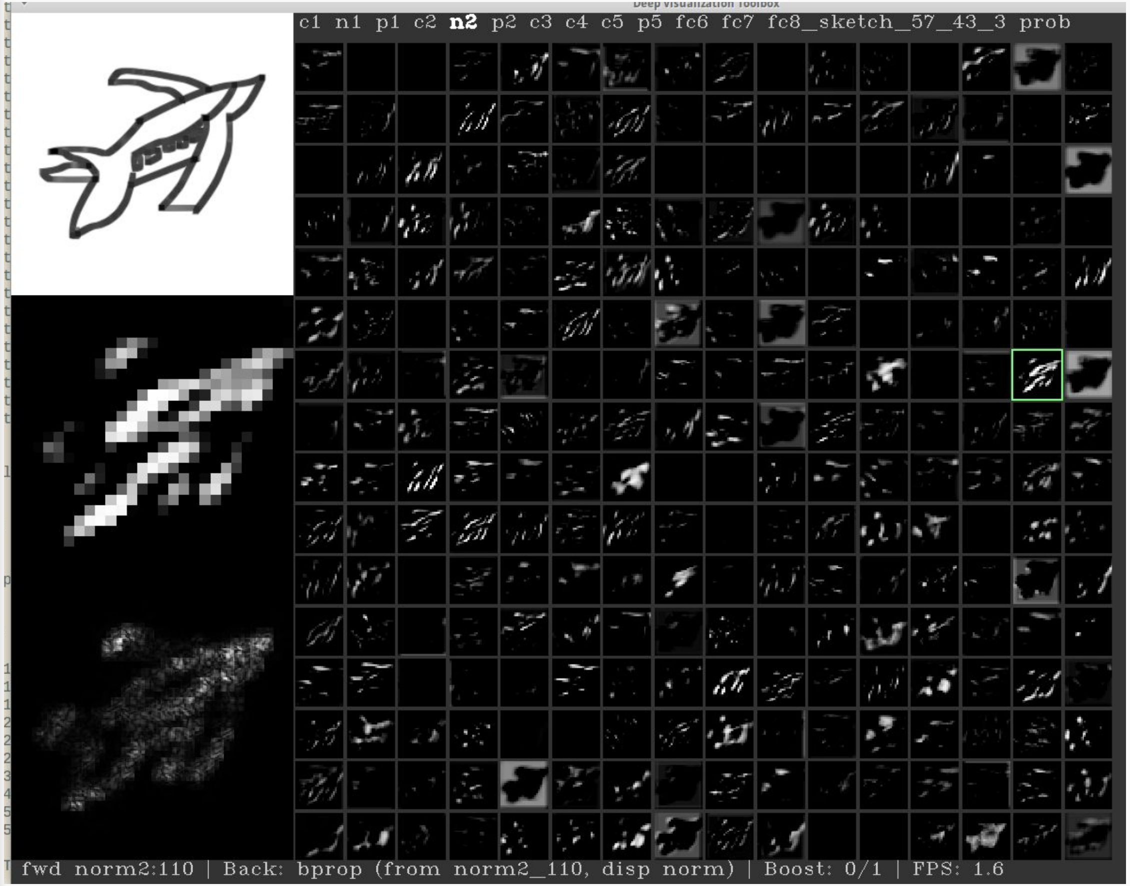
- Filter Size : 11 x 11
- Stride: 4
- Output filters: 96



Normalized Layer 2

The Local Response Normalization (LRN) Layer after the 2nd convolutional operation:

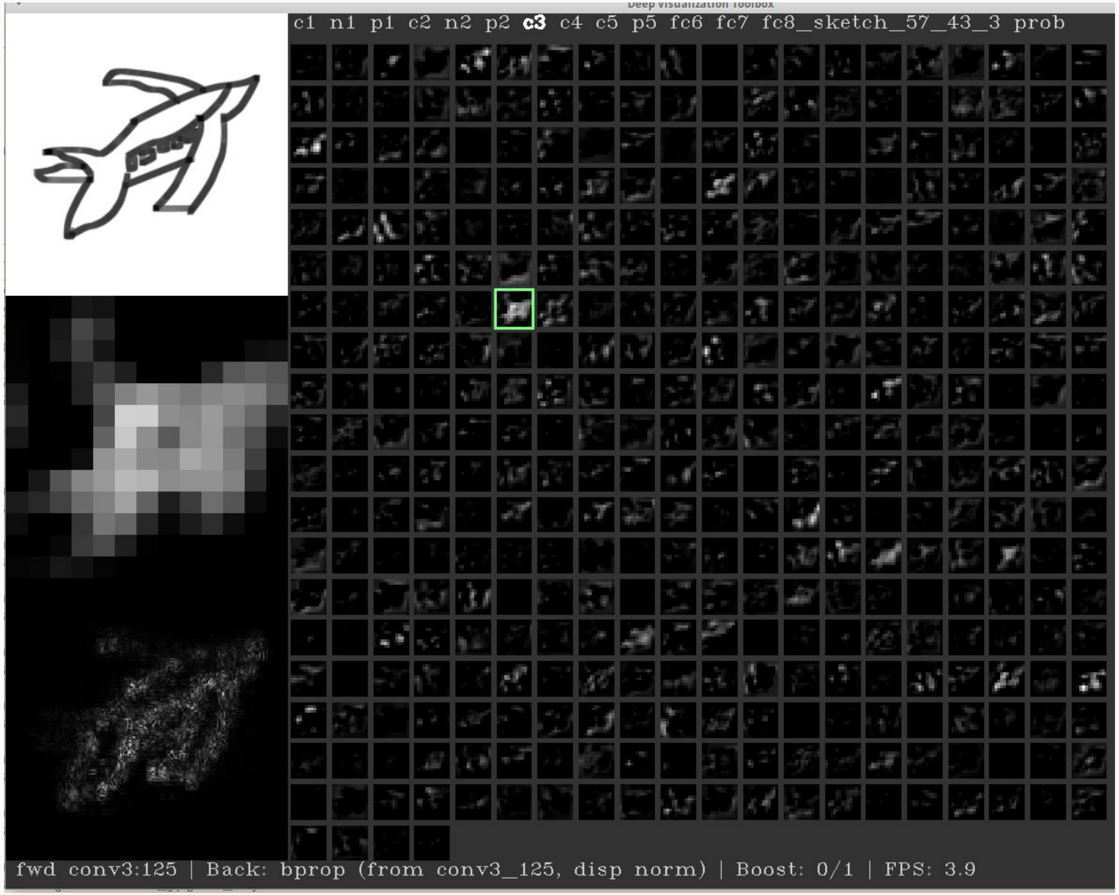
- Filter Size : 5 x 5
- Stride: 1
- Output filters: 256



Convolutional Layer 3

3rd Convolutional layer:

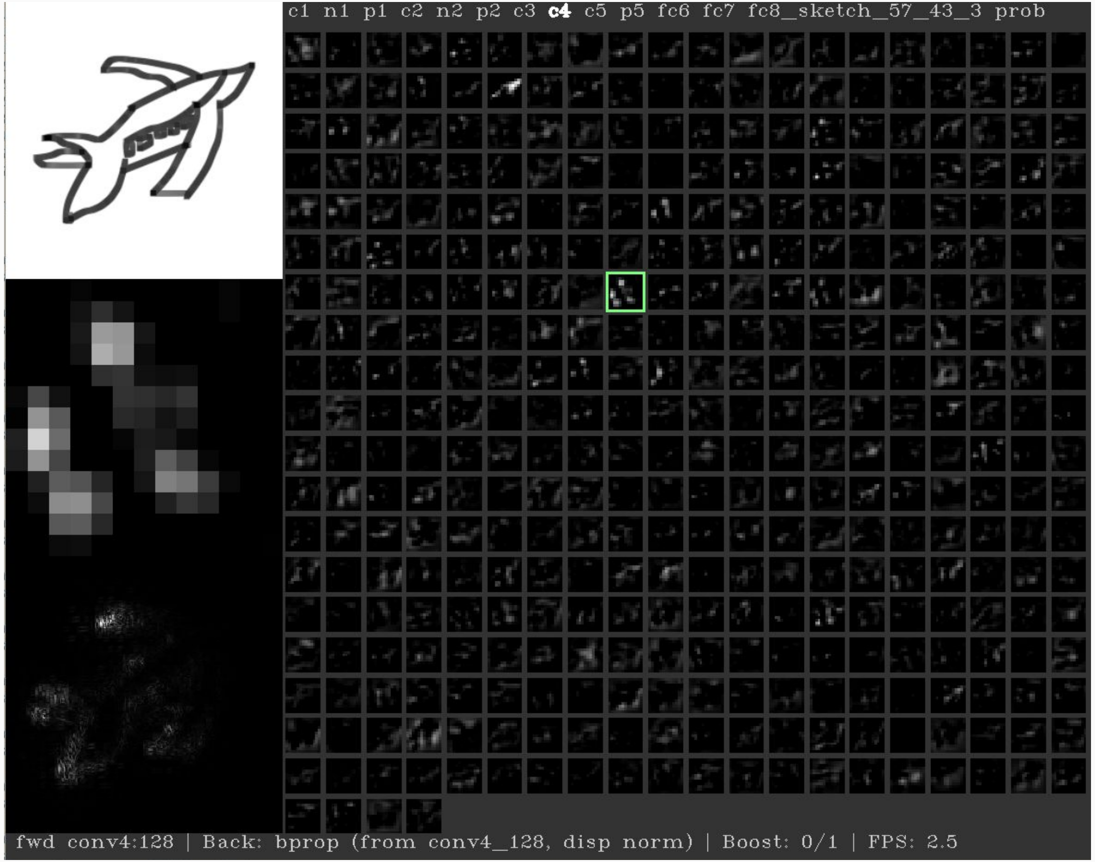
- Filter Size : 3 x 3
- Stride: 1
- Output filters: 384



Convolutional Layer 4

4th Convolutional layer:

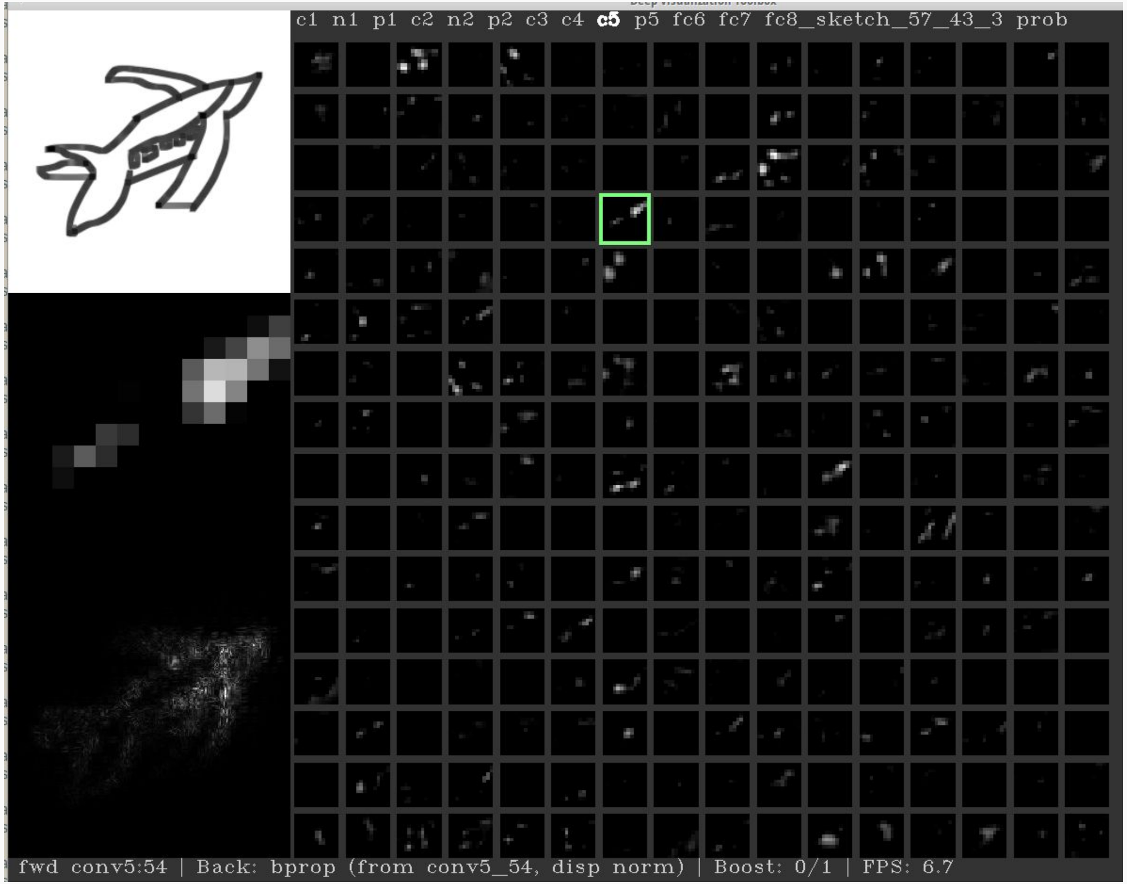
- Filter Size : 3 x 3
- Stride: 1
- Output filters: 384



Convolutional Layer 5

5th Convolutional layer:

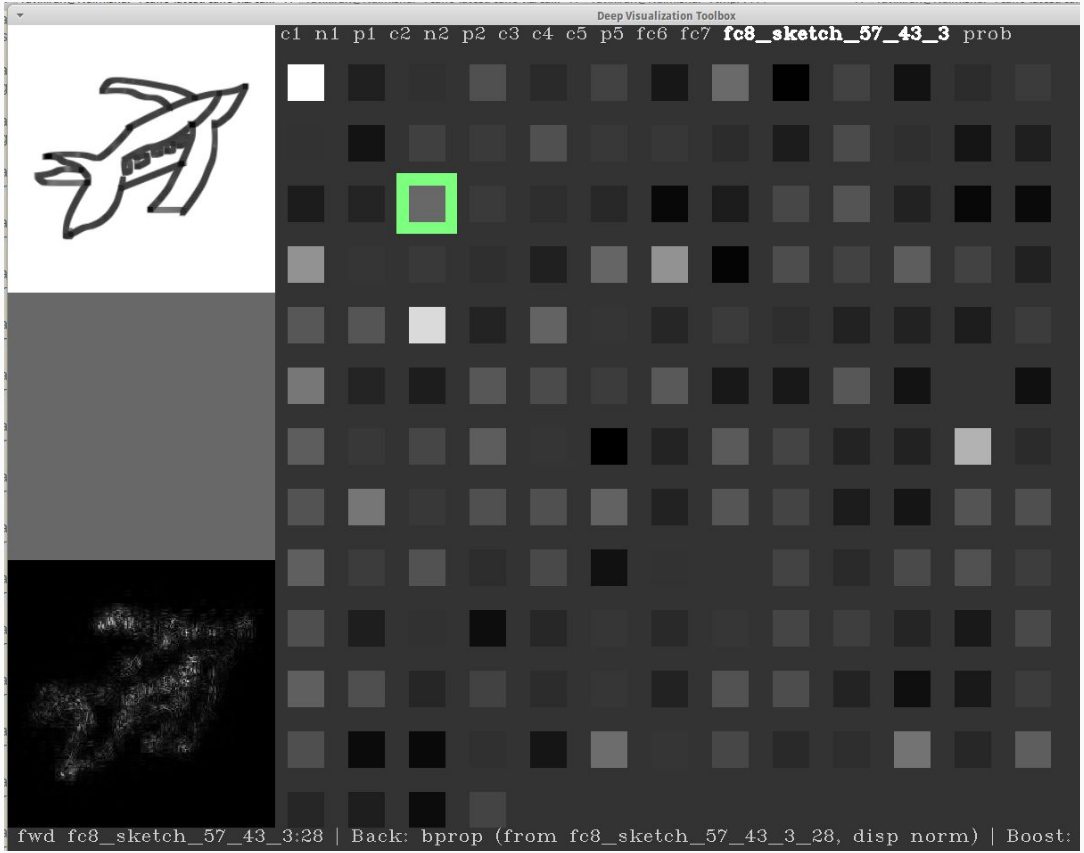
- Filter Size : 3 x 3
- Stride: 1
- Output filters: 256



Fully Connected Layer 8

Three fully connected layers with output neurons:

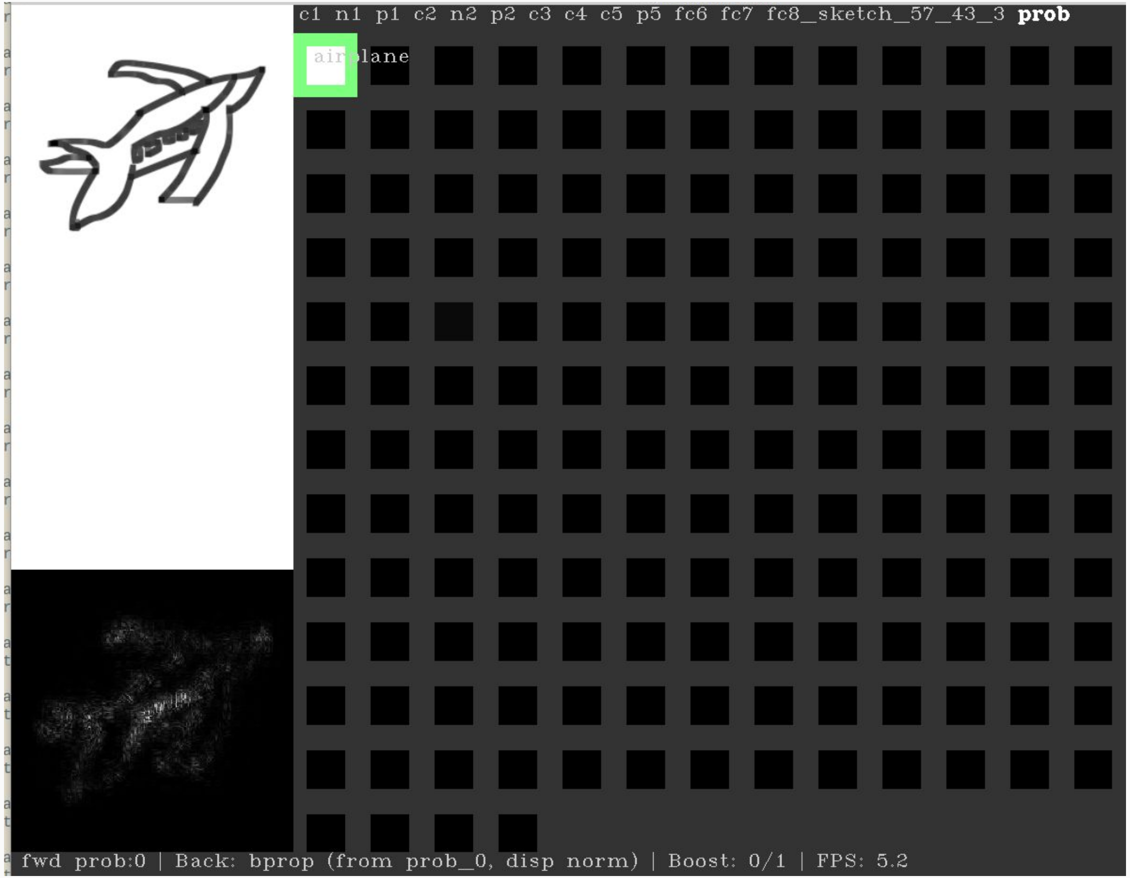
- FC6 : 4096
- FC7 : 4096
- FC8 : 160



Output Probability

Softmax operation applied to the Fully Connected Layer 8 to observe the class probabilities

- White filter corresponds to the class predicted for the sketch by the network.



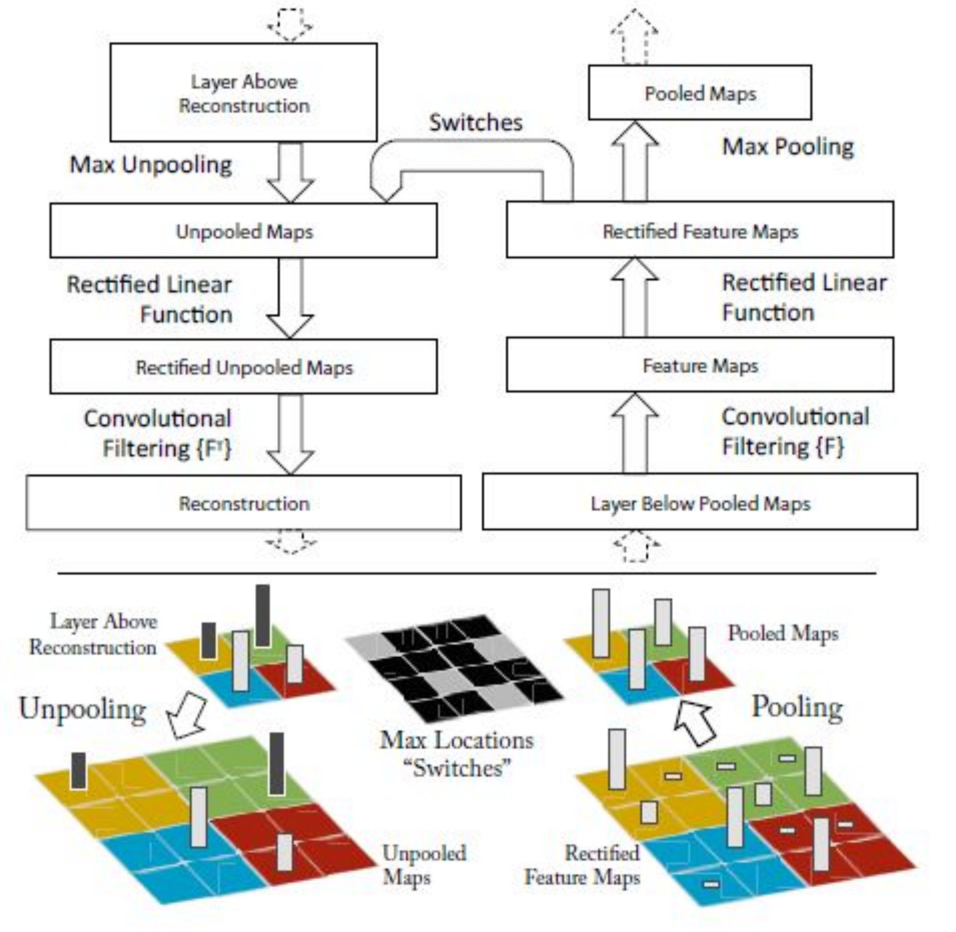
Visualizing layers by finding the Maximum Activated Input Patch & perform Deconvolution

- Finding the top 9 patches from the input images in validation set which maximally activate a filter for a certain layer.
- To understand which kind of features are learned by filters in different layers and observe their sensitivity towards certain classes & localized patches with certain geometry.
- Perform deconvolution for the filters that are activated by maximally activating input patches.

Visualization with Deconvnet

1. Unpooling
2. Rectification
3. Filtering

Zeiler, Matthew D. and Fergus, Rob. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. URL http://arxiv.org/abs/1311.2901.



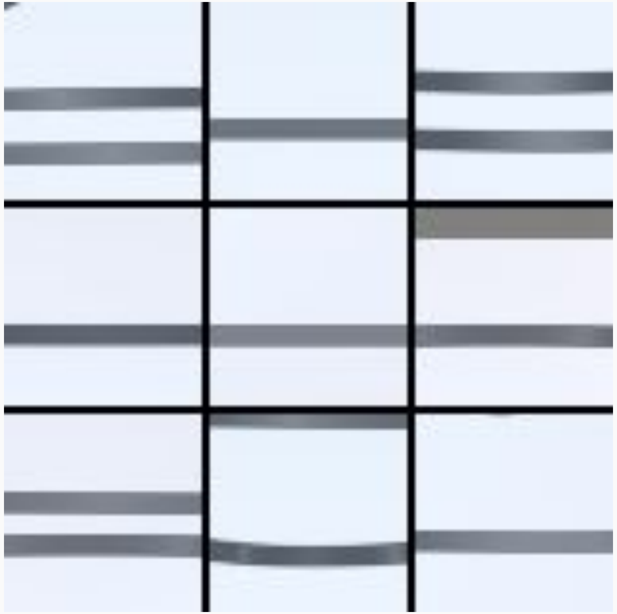
Convolution Layer 1

Input Patch for Maximum Activation

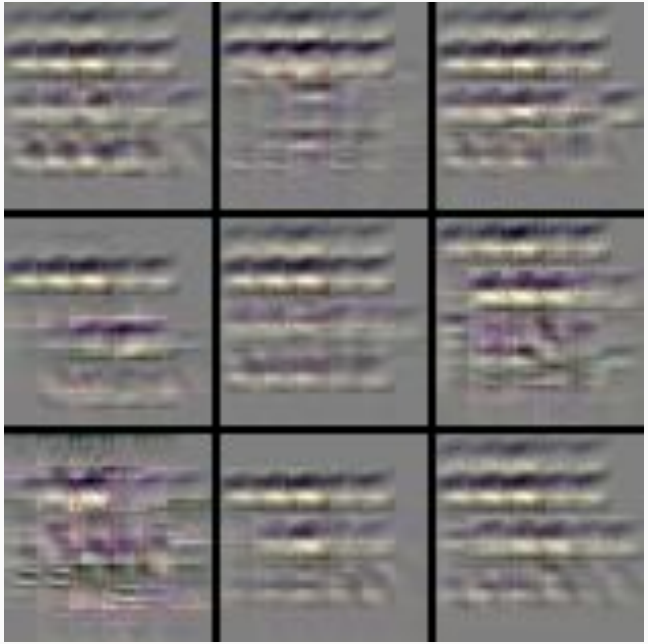
Deconvolution of the activated patch

- 11 x 11 Input Patch for 96th filter in 1st convolutional layer
- Filter sensitive to edge transition.

Convolution Layer 2



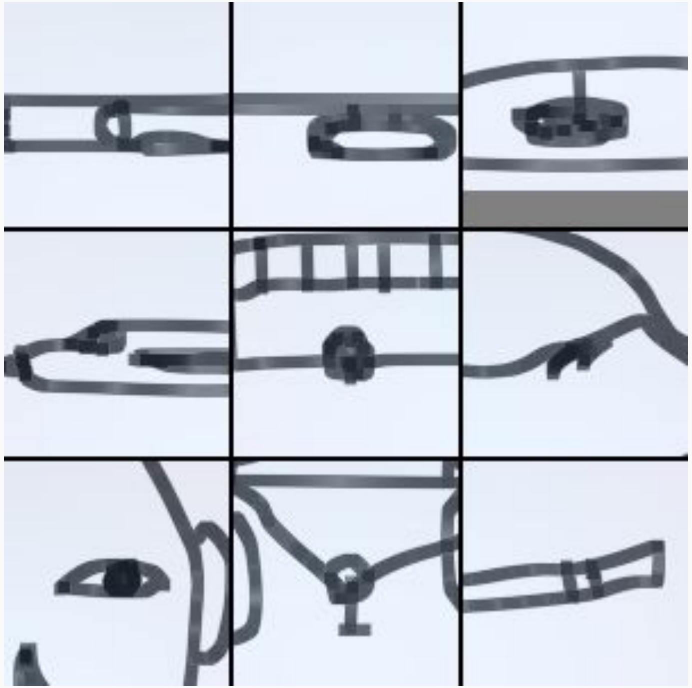
Input Patch for Maximum Activation



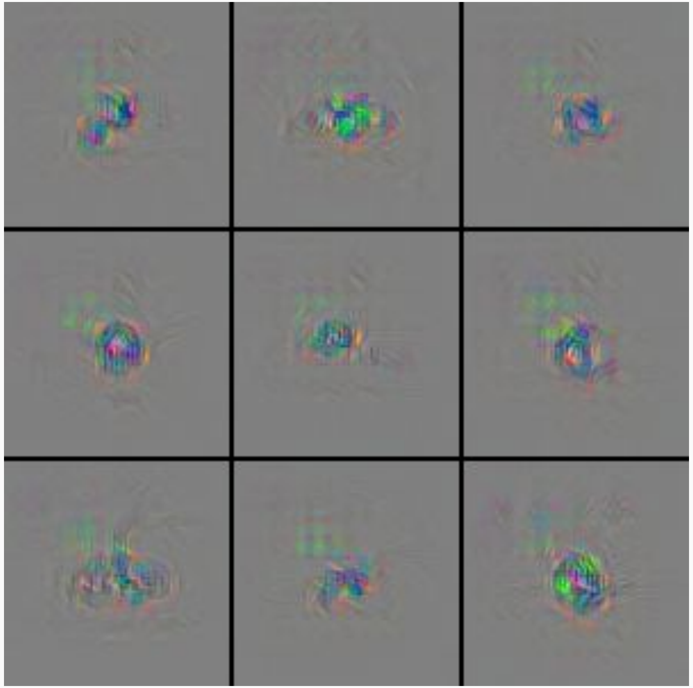
Deconvolution of the activated patch

- 51 x 51 Input Patch for 256th filter in 2nd convolutional layer
- Filter sensitive to straight lines.

Convolution Layer 3



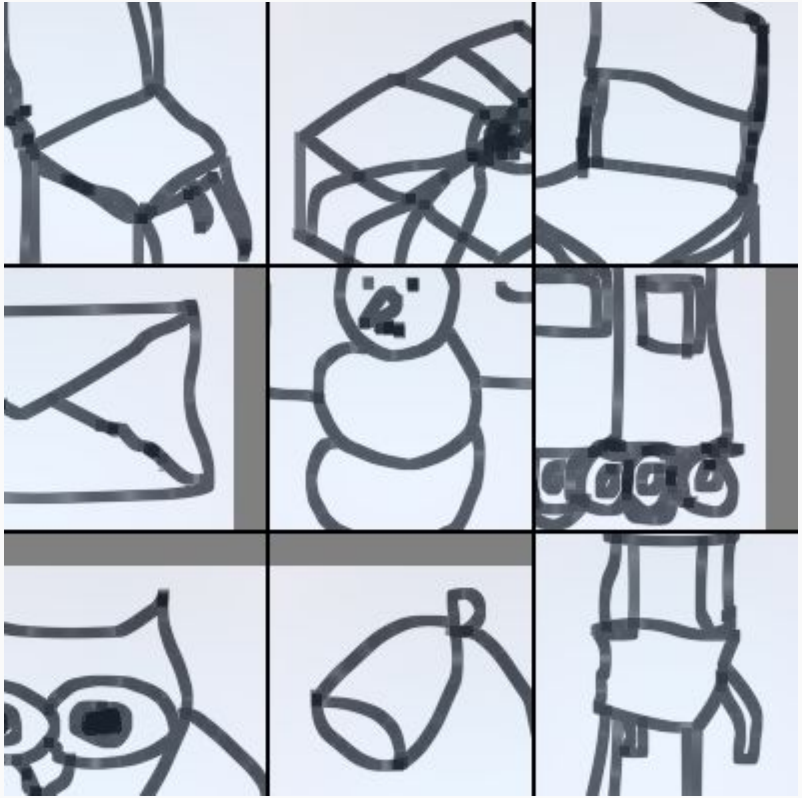
Input Patch for Maximum Activation



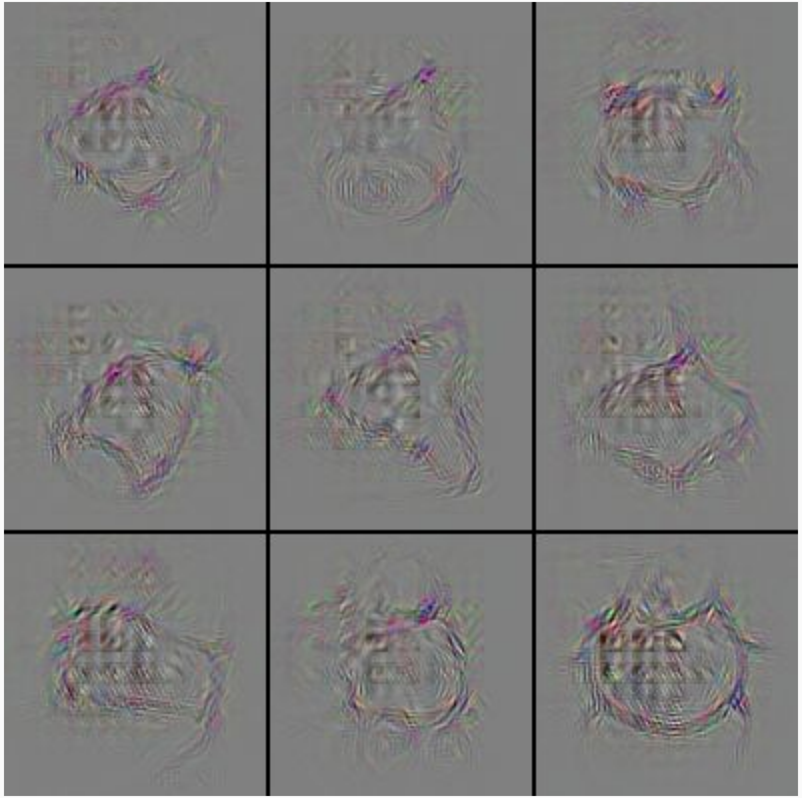
Deconvolution of the activated patch

- 99 x 99 Input Patch for 384th filter in 3rd convolutional layer
- Filter sensitive to specific parts of sketches like circles in middle i.e. eyes, wheel etc.

Convolution Layer 4



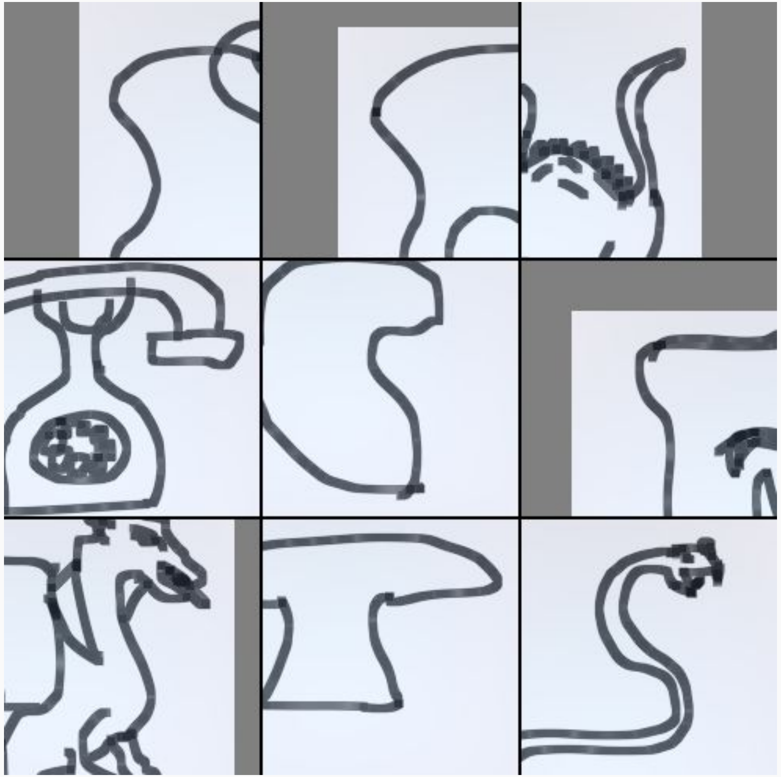
Input Patch for Maximum Activation



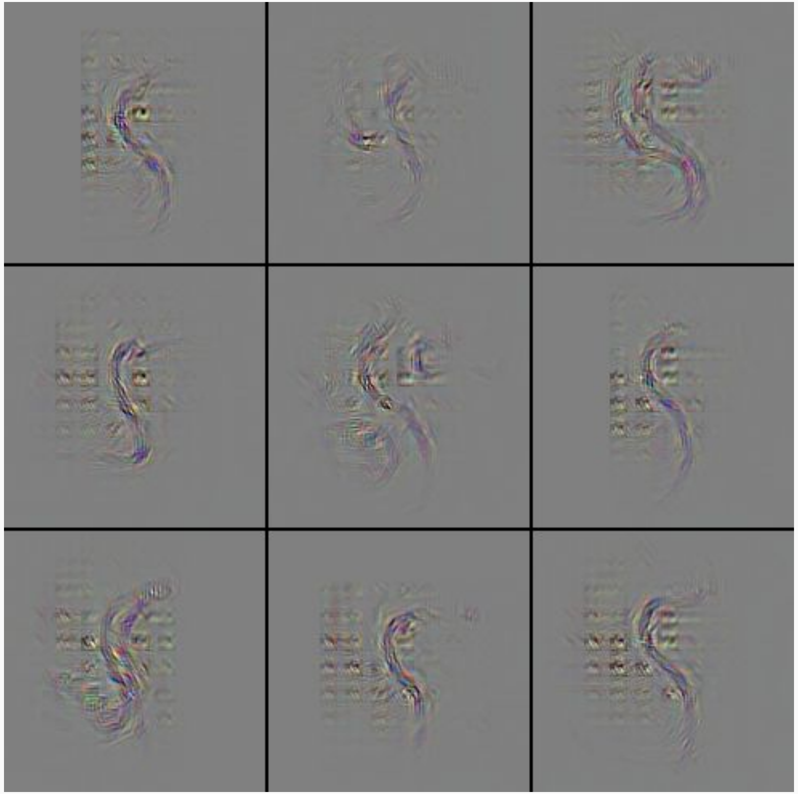
Deconvolution of the activated patch

- 131 x 131 Input Patch for 384th filter in 4th convolutional layer
- Filter sensitive to specific classes like chairs, present,envelope etc.

Convolution Layer 5



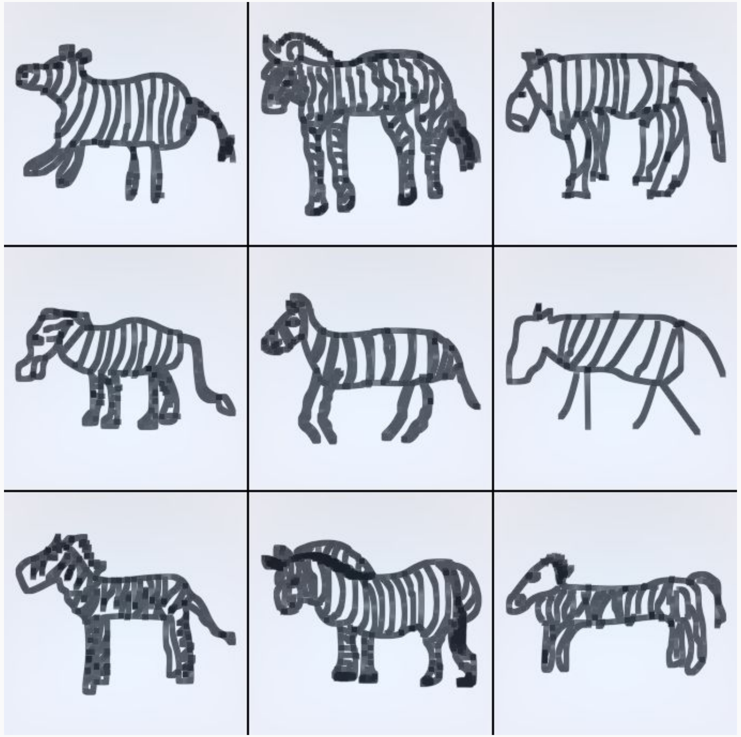
Input Match for Maximum Activation



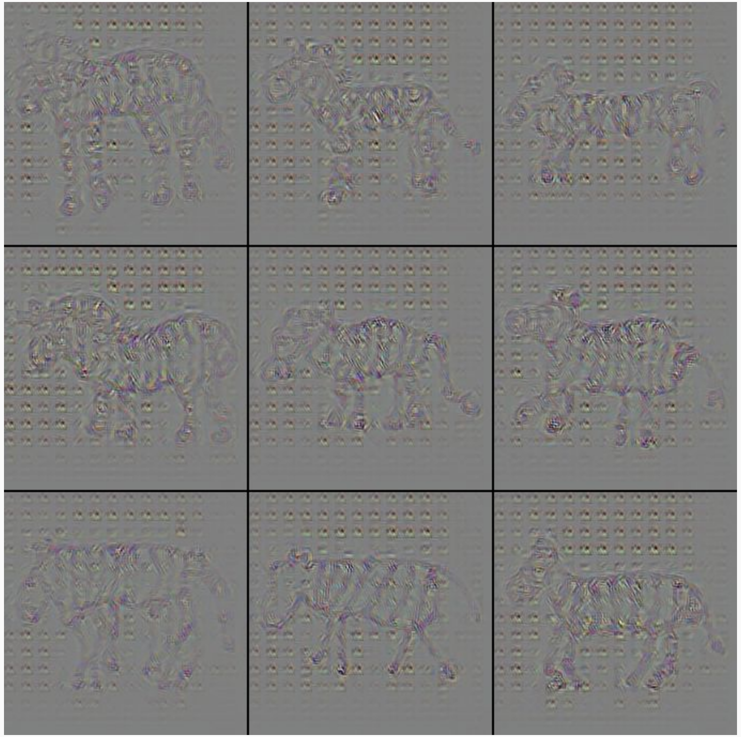
Deconvolution of the activated patch

- 163 x 163 Input Patch for 256th filter in 5th convolutional layer
- Filter sensitive to specific classes that have curved geometry like snake, dragon, telephone etc.

Fully Connected Layer 8



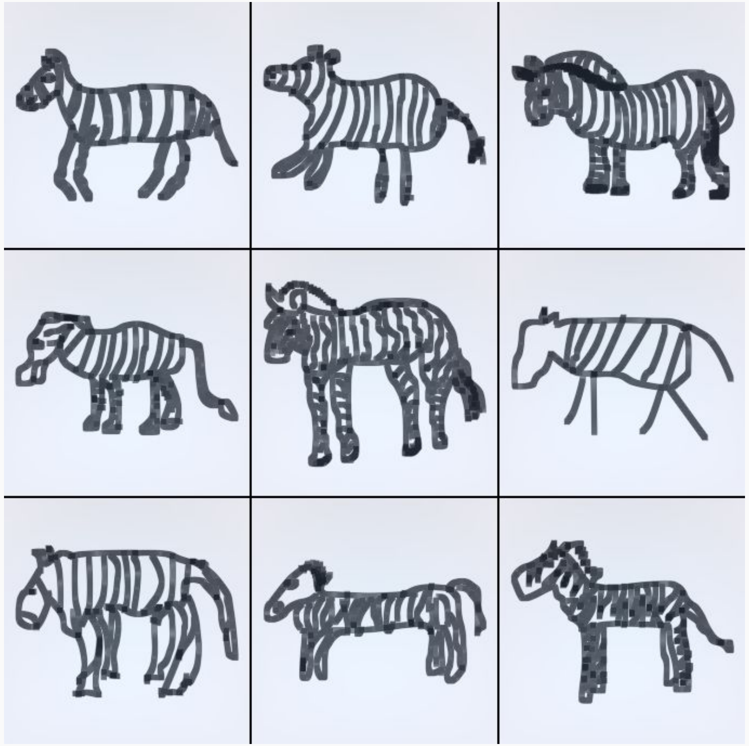
Input Patch for Maximum Activation



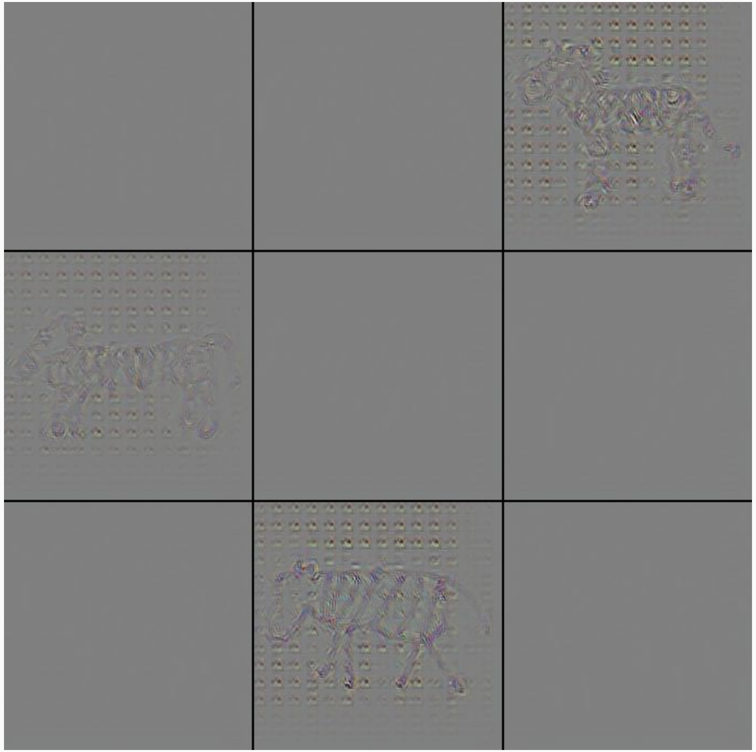
Deconvolution of the activated patch

- 227 x 227 Input Patch (complete image) for 160th filter in 8th layer which is fully connected.
- Filter sensitive to the zebra class. (which is also 160th class in target attribute)

Softmax Probability Layer



Input Patch for Maximum Activation



Deconvolution of the activated patch

- 227 x 227 Input Patch for 256th filter in the final softmax layer.
- The final layer is sensitive to only certain hand drawn images of zebras .

Lab Meeting Presentation-2

May 31, 2017

Prerit Gupta
Summer Intern

Ravi Kiran Sarvadevabhatla
PhD Student

Objective

- Characterizing Visual Representations within Sketch-a-Net
- Using Yosinki's Deep Visualization Toolbox to analyse the different layers of Alexnet for top 160 classes of TU Berlin Sketch Dataset.

Characterizing Visual Representation within SketchNets

Results from : Q. Yu, Y. Yang, Y.-Z. Song, T. Xiang , and T. Hospedales. Sketch-a-net that beats humans, BMVC, 2015

Conclusion

- Filters in lower level layers are sensitive to edges, patterns, corners, lines and certain localized geometries present in sketches.
- Filters in higher level layers are sensitive to certain classes & some specific objects in sketches.
- Deconvolution of the filters shows which input pixels resulted to activation within these filters as we map input space from feature space.

For next week ...

- Extending the usuage of Yosinski's Deep Visualization Toolbox to analyze sketch CNNs fine-tuned for sketches (VGG, GoogLeNet, ResNet) and Sketch-CNN (Matconvnet → Caffe)
- To plot histogram for every filter for the top 5 classes it is the most sensitive.
- Do a weighted sum of all the histograms to get a single histogram for every layer & compare the top classes for which it is sensitive with the class performance over network.

Thank You