Lab Meeting Presentation-1

Prerit Gupta

May 24, 2017

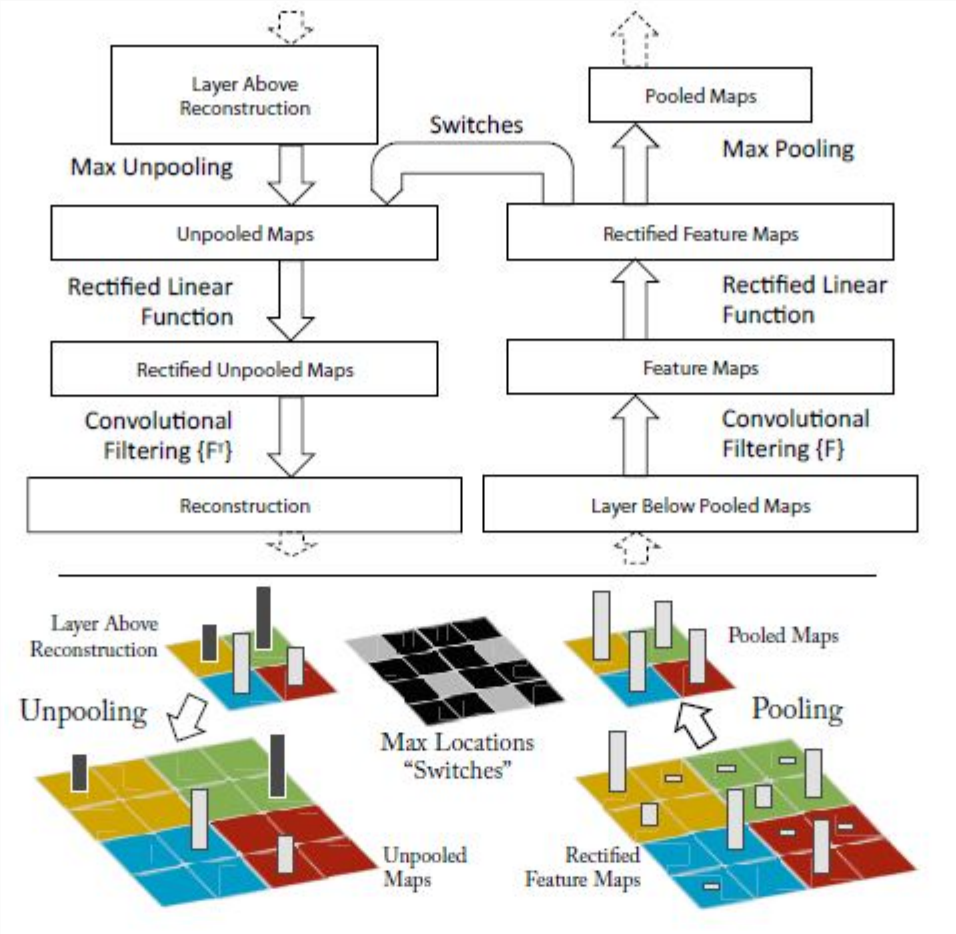
Objective

- Visualizing and Understanding Convolutional Networks
- Characterizing Visual Representations within Sketch-a-Net

Visualizing & Understanding Convolutional Networks

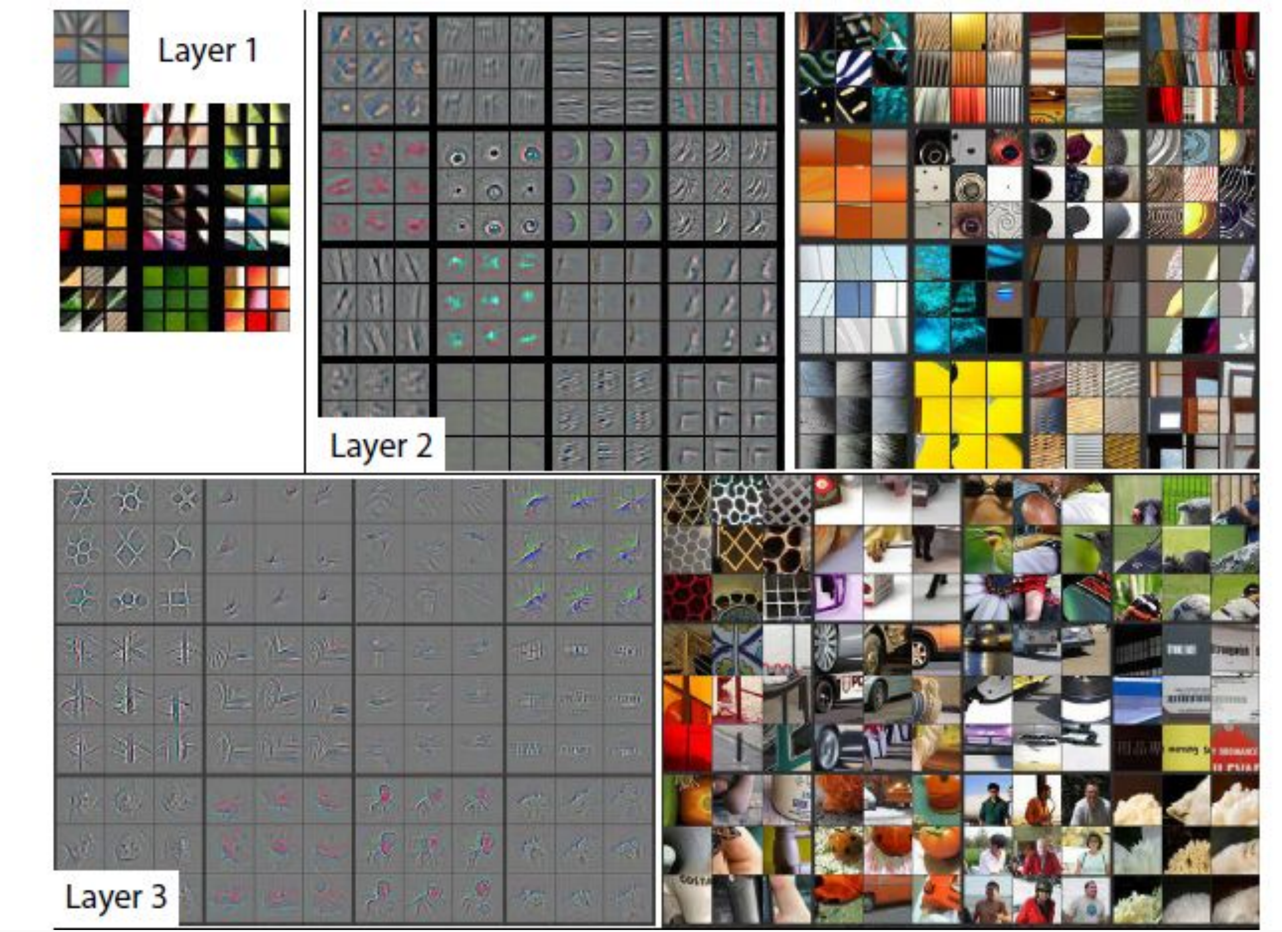
Visualization with Deconvnet

1. Unpooling
2. Rectification
3. Filtering

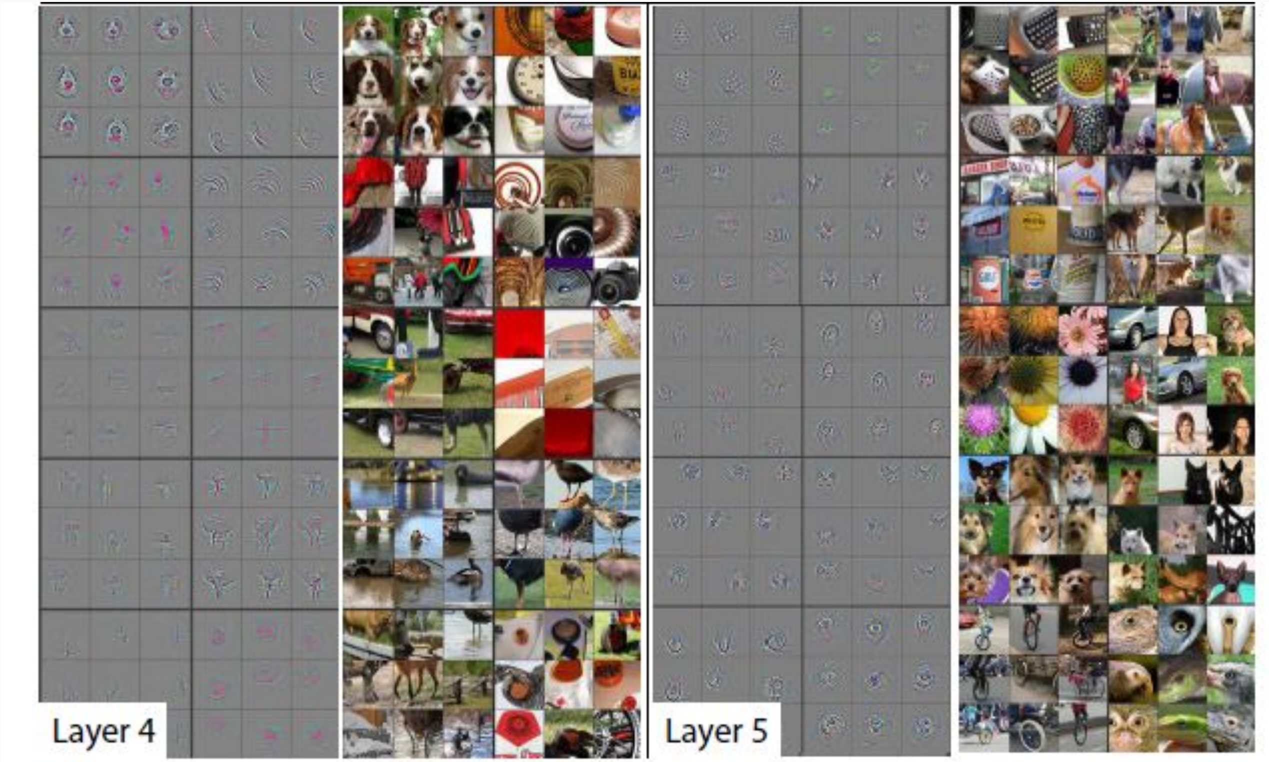


Zeiler, Matthew D. and Fergus, Rob. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. URL http://arxiv.org/abs/1311.2901.

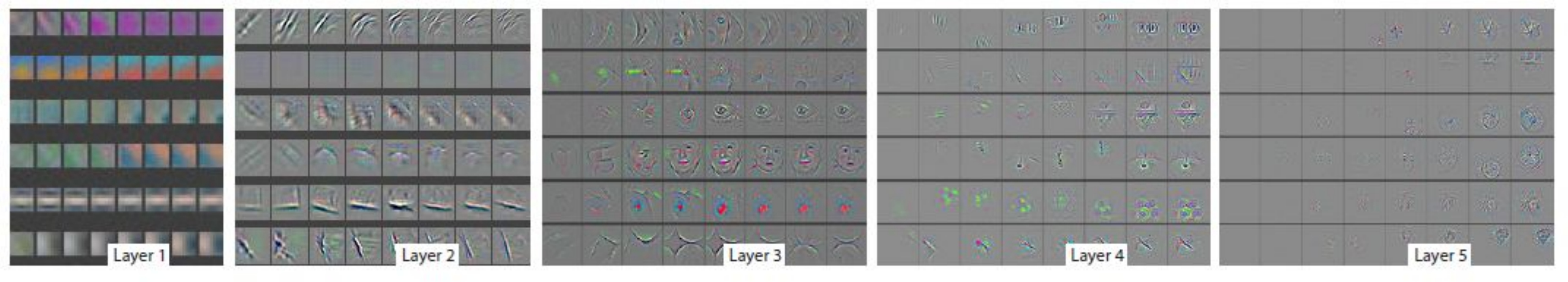
Feature Visualization : Lower Layers



Feature Visualization : Higher Layers

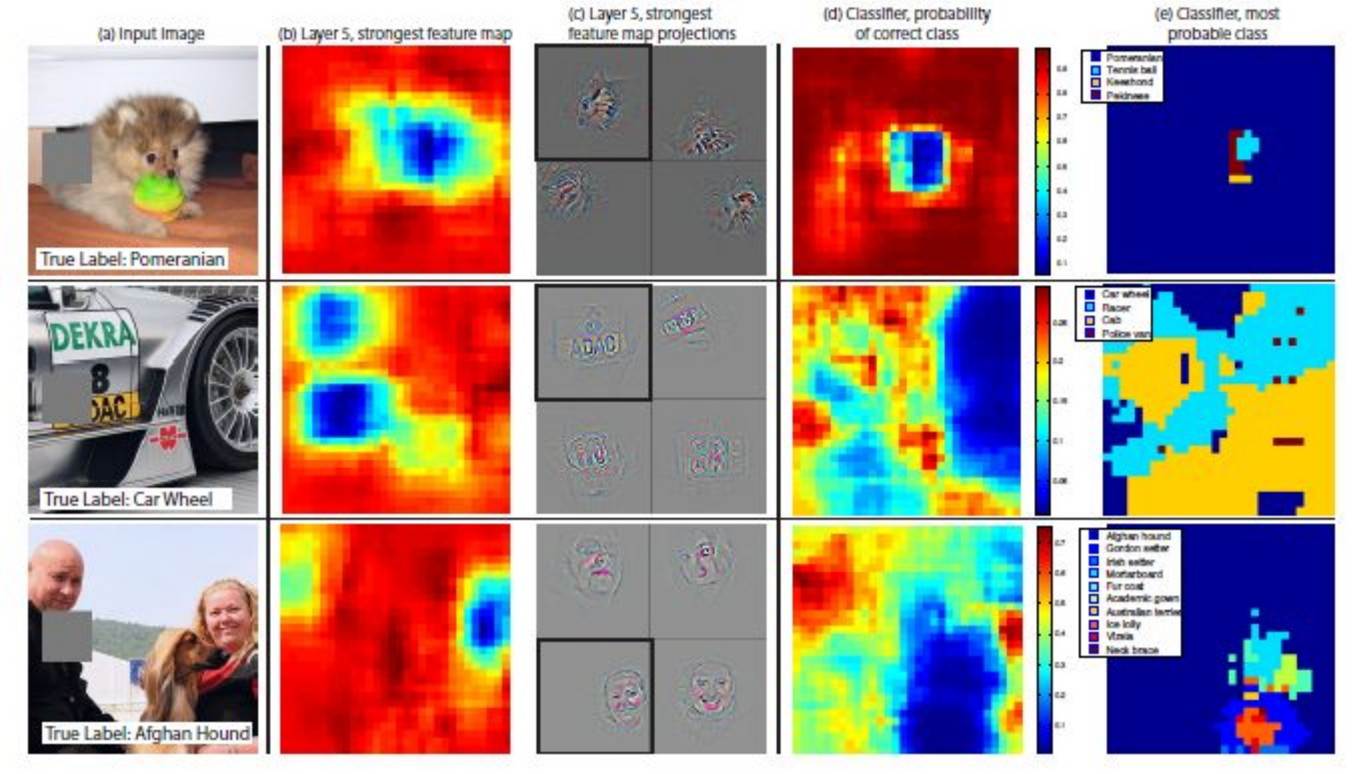


Feature Evolution during Training

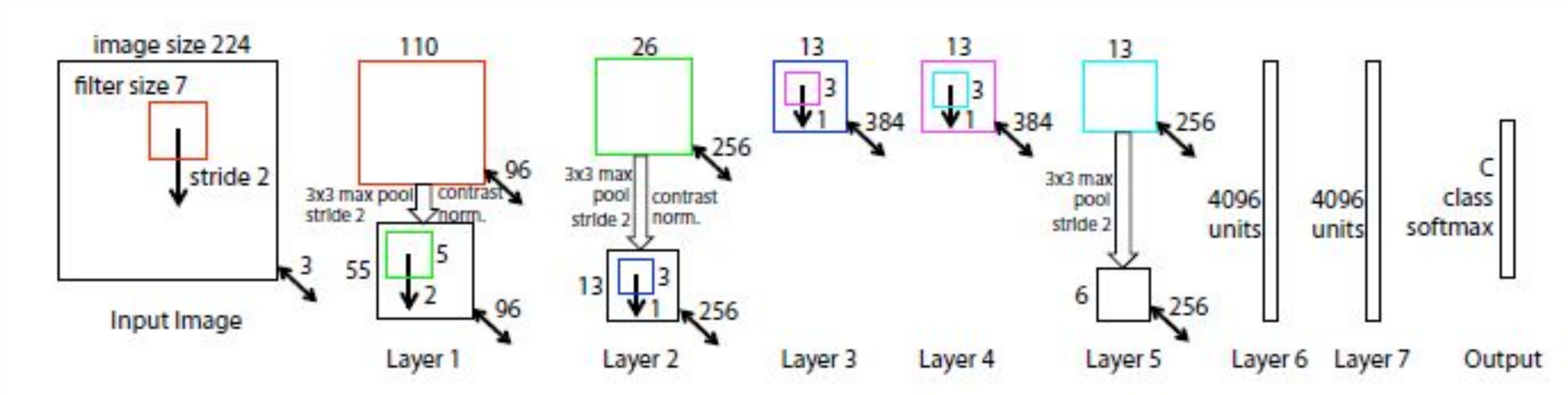


- Epochs in each block : [1,2,5,10,20,30,40,64]
- Strongest activation across all training examples for a given feature map.
- More epochs required for learning at higher layers.

Occlusion Sensitivity



CNN Architecture



8 Layer Convnet Model

Modification in AlexNet Architecture

- First Layer Convolution : 11 x 11 changed to 7 x 7
- First Layer Stride : 4 changed to 2

Comparing Error Rates on layer removal

Error %	Train Top-1	Val Top-1	Val Top-5
Our replication of Krizhevsky *et al.* [18], 1 convnet	35.1	40.5	18.1
Removed layers 3,4	41.8	45.4	22.1
Removed layer 7	27.4	40.0	18.4
Removed layers 6,7	27.4	44.8	22.4
Removed layer 3,4,6,7	71.1	71.3	50.1
Adjust layers 6,7: 2048 units	40.3	41.7	18.8
Adjust layers 6,7: 8192 units	26.8	40.0	18.1
Our Model (as per Fig. 3)	33.1	38.4	16.5
Adjust layers 6,7: 2048 units	38.2	40.2	17.6
Adjust layers 6,7: 8192 units	22.0	38.8	17.0
Adjust layers 3,4,5: 512,1024,512 maps	18.8	**37.5**	**16.0**
Adjust layers 6,7: 8192 units and Layers 3,4,5: 512,1024,512 maps	**10.0**	38.3	16.9

Understanding which layers are more important

Classification Accuracies

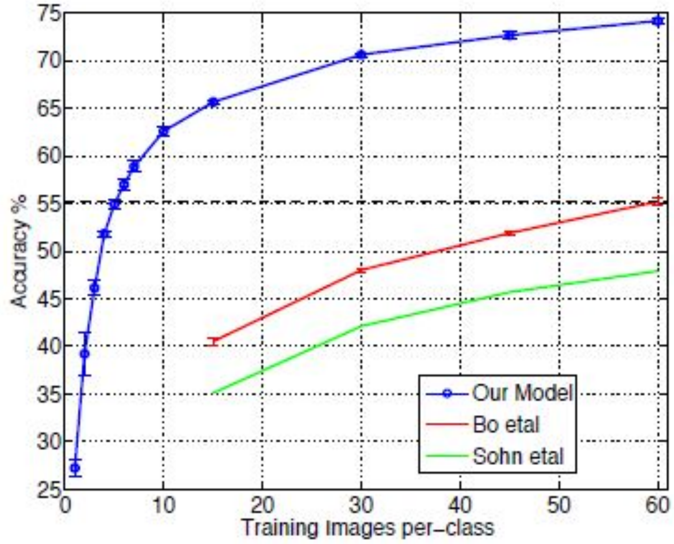


Table 3. Caltech-101 classification accuracy for our convnet models, against two leading alternate approaches

# Train	Acc % 15/class	Acc % 30/class
Bo *et al.* [3]	–	81.4 ± 0.33
Yang *et al.* [17]	73.2	84.3
Non-pretrained convnet	22.8 ± 1.5	46.5 ± 1.7
ImageNet-pretrained convnet	$\mathbf{83.8 \pm 0.5}$	$\mathbf{86.5 \pm 0.5}$

Table 4. Caltech 256 classification accuracies

# Train	Acc % 15/class	Acc % 30/class	Acc % 45/class	Acc % 60/class
Sohn *et al.* [24]	35.1	42.1	45.7	47.9
Bo *et al.* [3]	40.5 ± 0.4	48.0 ± 0.2	51.9 ± 0.2	55.2 ± 0.3
Non-pretr.	9.0 ± 1.4	22.5 ± 0.7	31.2 ± 0.5	38.8 ± 1.4
ImageNet-pretr.	$\mathbf{65.7 \pm 0.2}$	$\mathbf{70.6 \pm 0.2}$	$\mathbf{72.7 \pm 0.4}$	$\mathbf{74.2 \pm 0.3}$

DeConvnet Approach

Information usage in different approaches to determine which pixel values are important in input:

- Backprop - input image & lower layers
- Deconvnet - higher layer gradient information
- Guided Backpropagation - combines both methods

Grun, Rupprecht, Nawab & Tombari, A Taxonomy and Library for Visualizing Learned Features in Convolutional Neural Networks.
URL: http://icmlviz.github.io/assets/papers/20.pdf

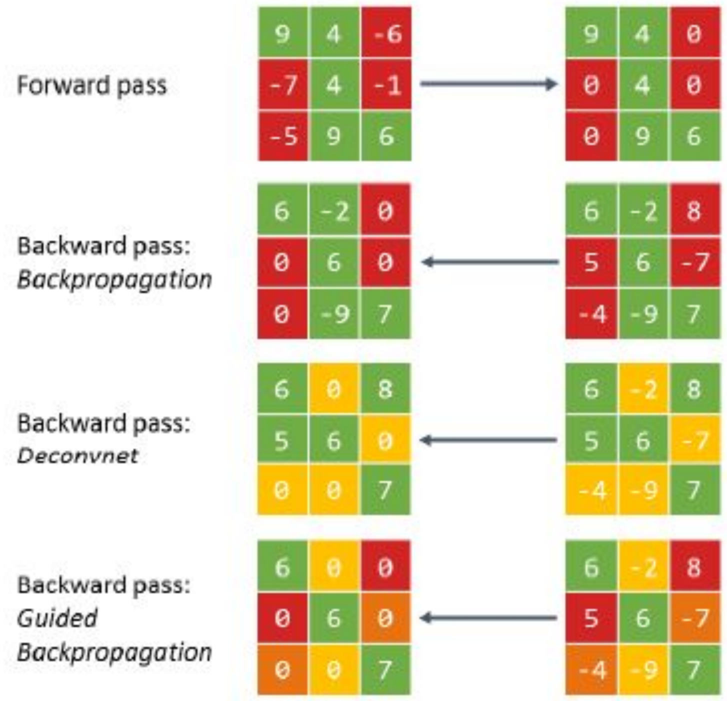
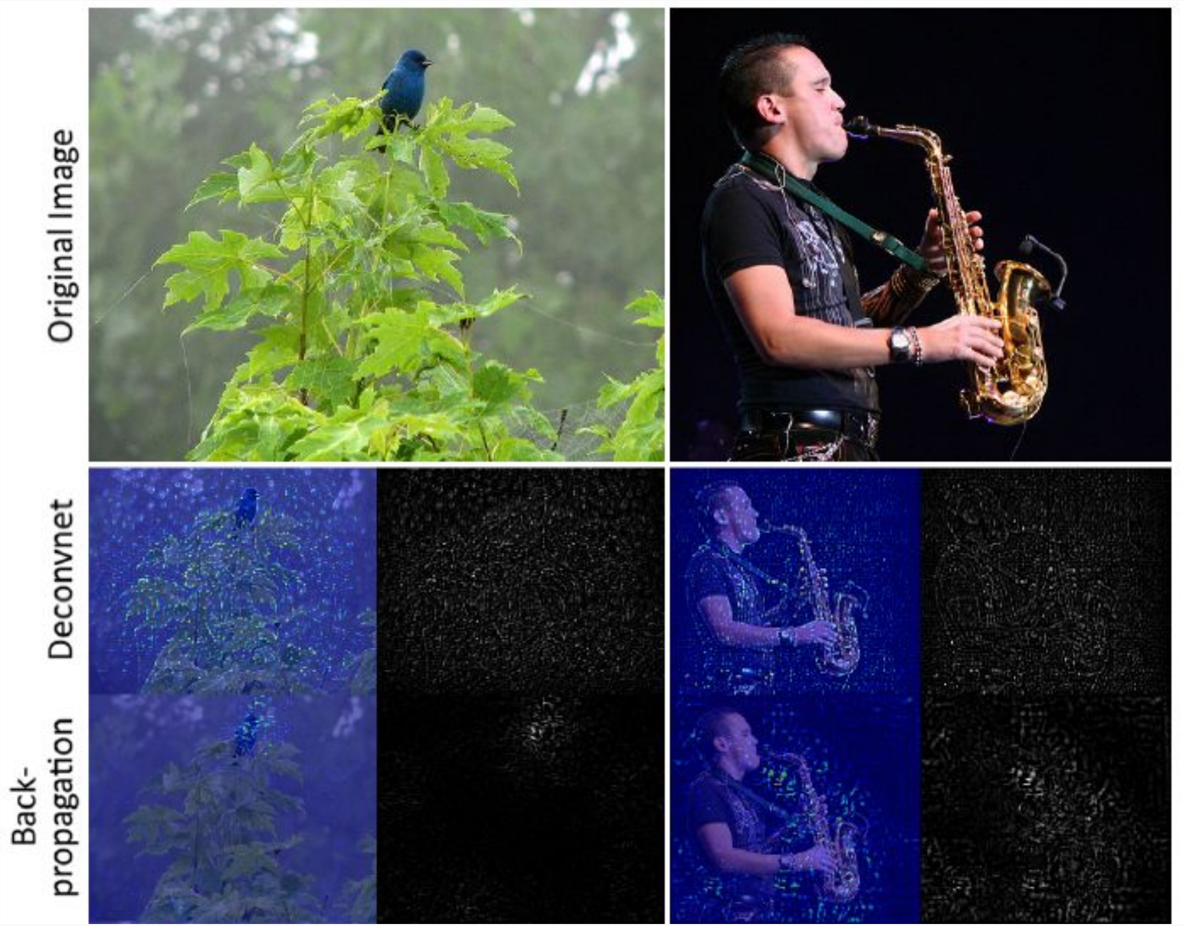
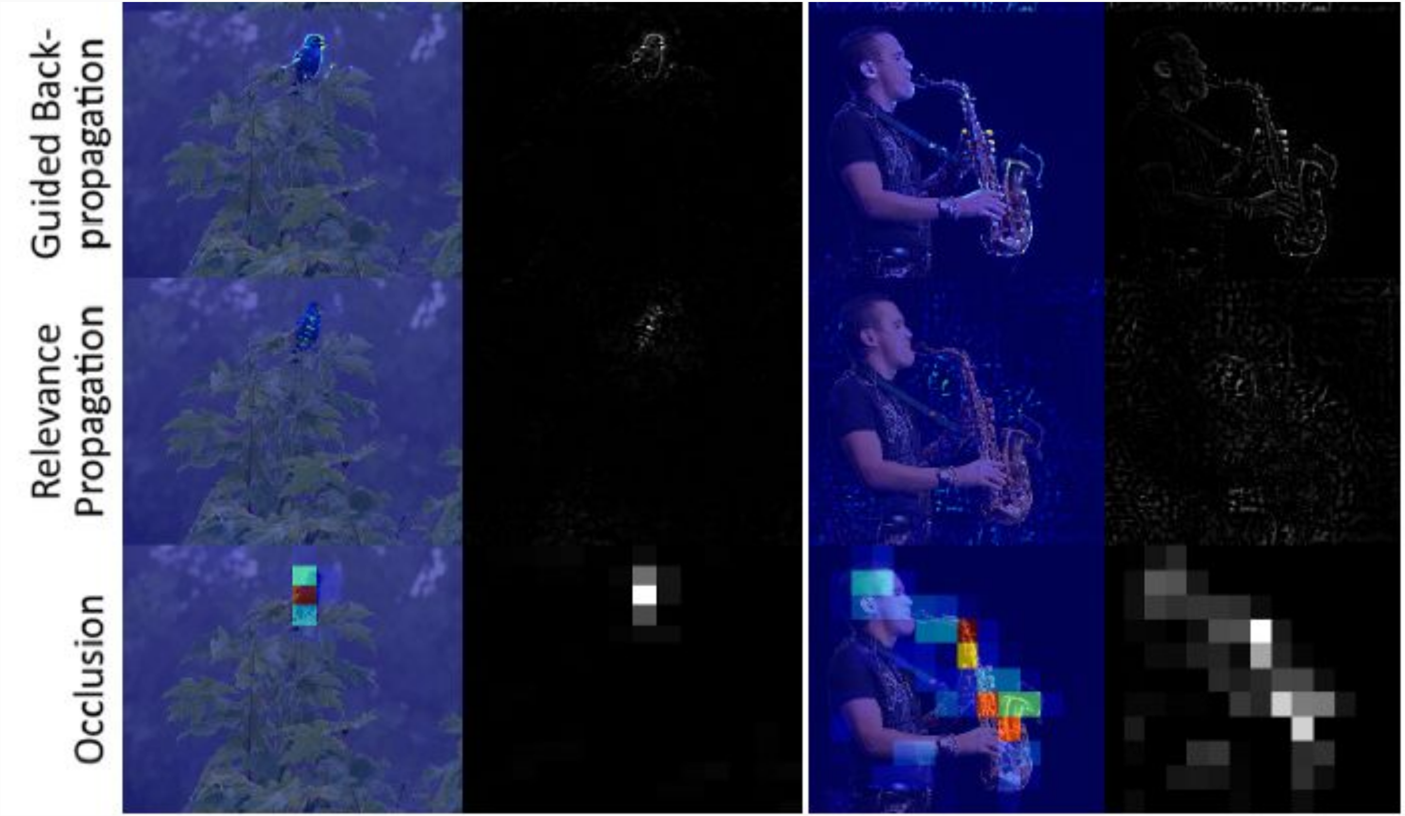


Figure 2. Different ways in which the pass through a ReLU layer affects contribution values for the Deconvnet method, Backpropagation, and Guided Backpropagation. The forward pass through the ReLU layer is shown for comparison.

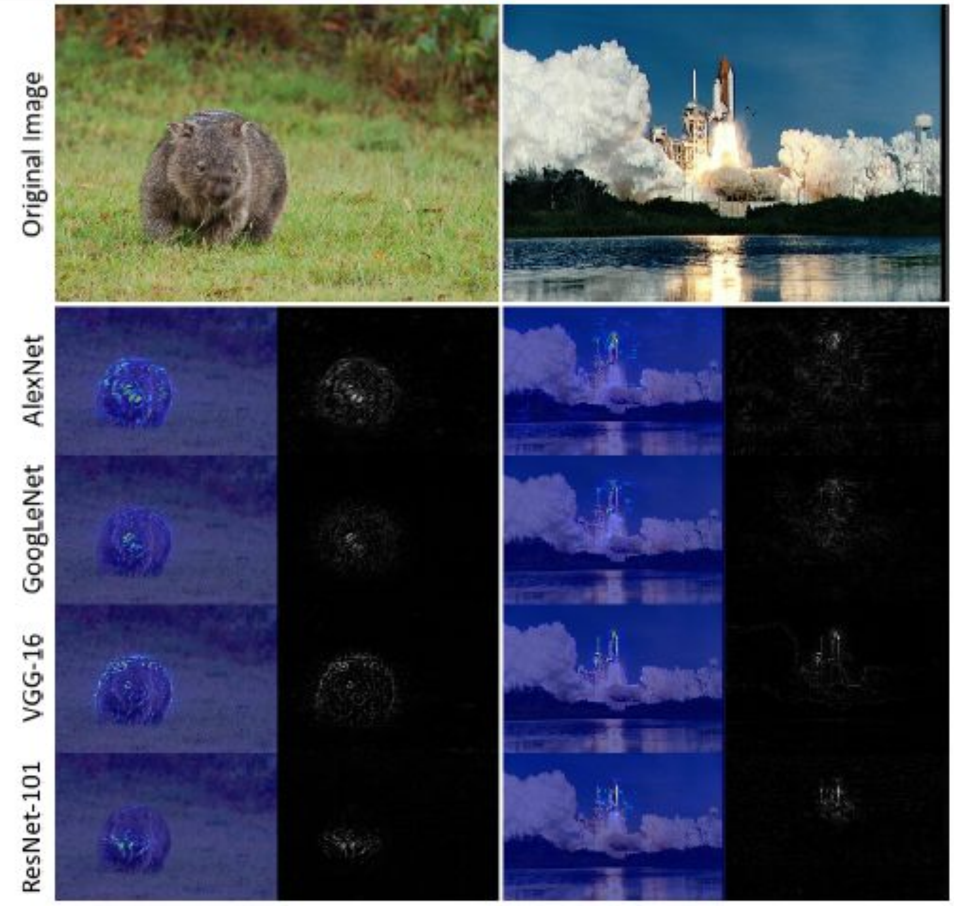
Comparing Different Visualization Methods

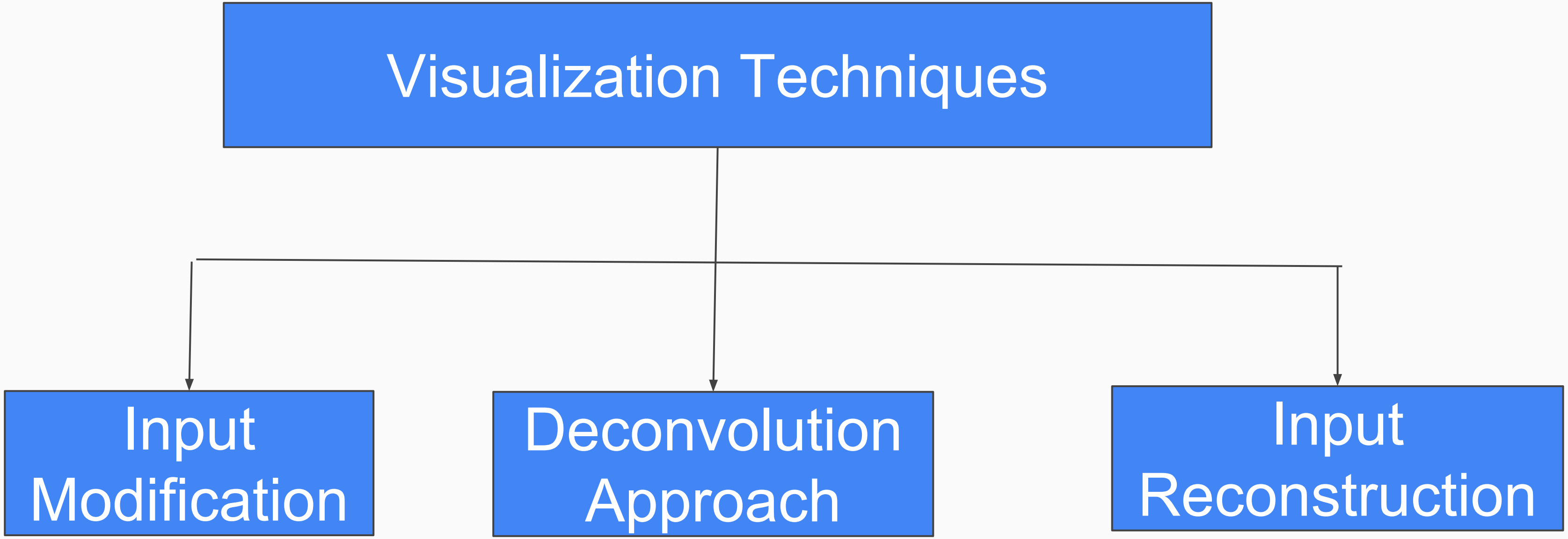


Comparing Different Visualization Methods



Comparative Visualization of Different Networks : Guided Backpropagation




Visualization Techniques
Input Modification
Deconvolution Approach
Input Reconstruction

Characterizing Visual Representation within SketchNets

Unique Characteristics of Sketches

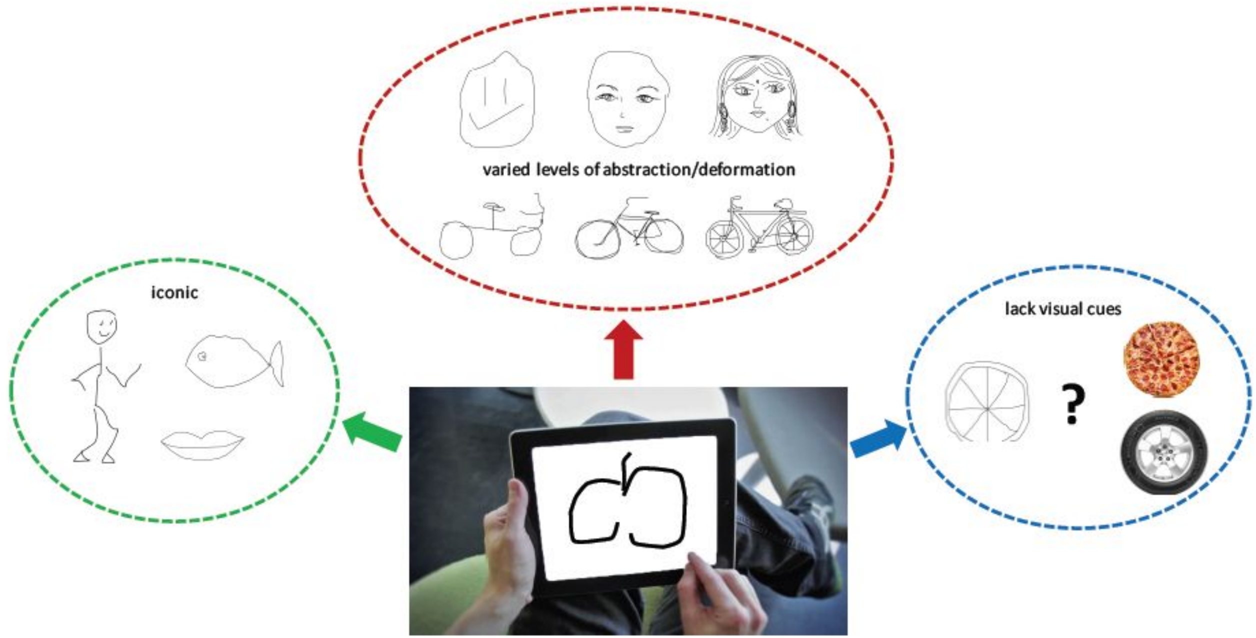


Fig. 1 Recognising a free-hand sketch is not easy due to a number of challenges

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

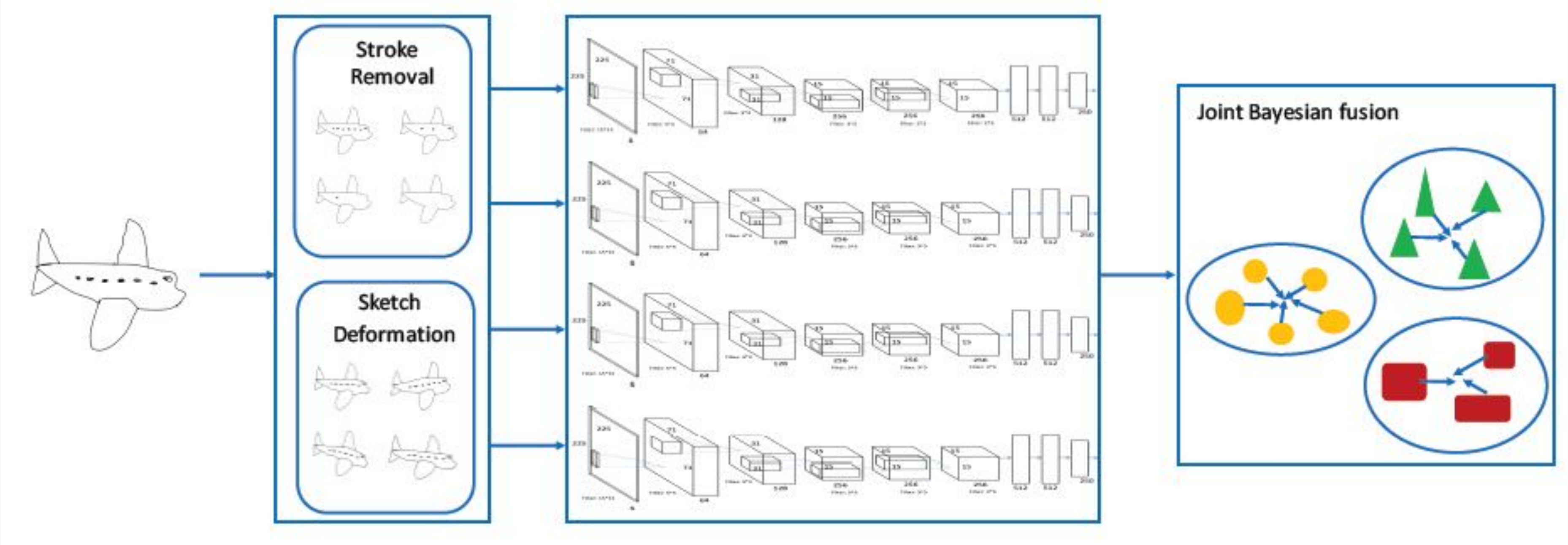
Unique characteristics of Sketch-a-Net Architecture

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

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

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

Ensemble Fusion



- Joint Bayesian (JB) - Feature Level Fusion
- 4 x 512 = 2048D concatenated feature vector

Comparing Different Networks

Models	Accuracy
HOG-SVM (Eitz et al, 2012)	56%
Ensemble (Li et al, 2013)	61.5%
MKL-SVM (Li et al, 2015)	65.8%
FV-SP (Schneider and Tuytelaars, 2014)	68.9%
AlexNet-SVM (Krizhevsky et al, 2012)	67.1%
AlexNet-Sketch (Krizhevsky et al, 2012)	68.6%
LeNet (LeCun et al, 2012)	55.2%
SN1.0 (Yu et al, 2015)	74.9%
Our Full Model	**77.95%**
Humans (Eitz et al, 2012)	73.1%

Table 2 Comparitive results on sketch recognition

Visualizing Sketch Nets

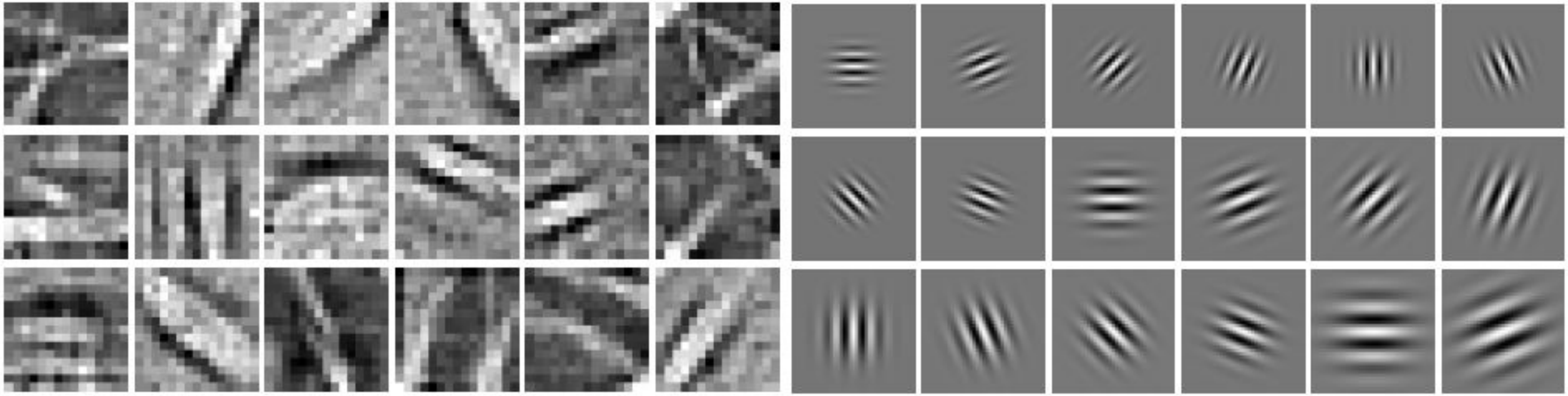
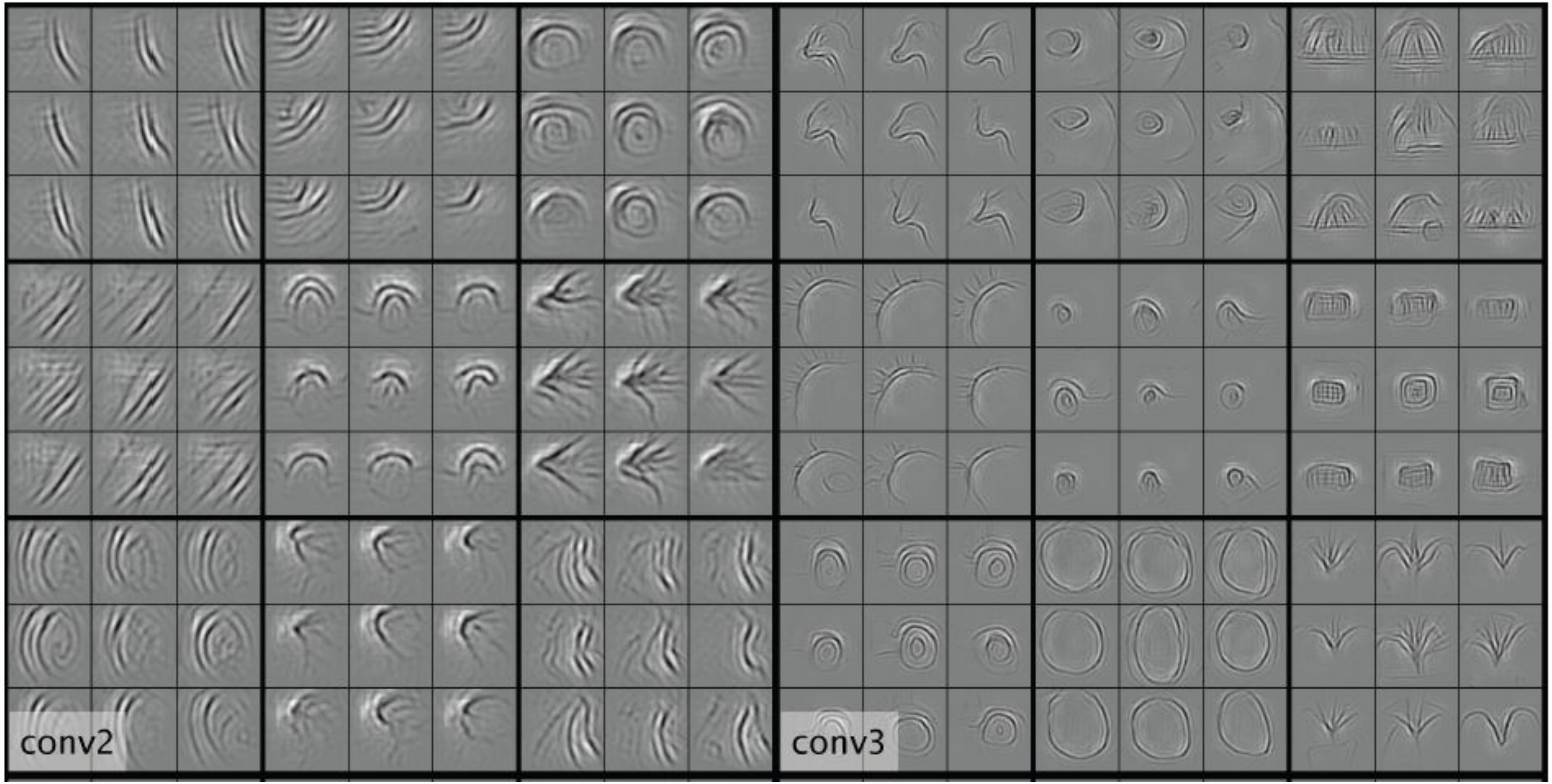


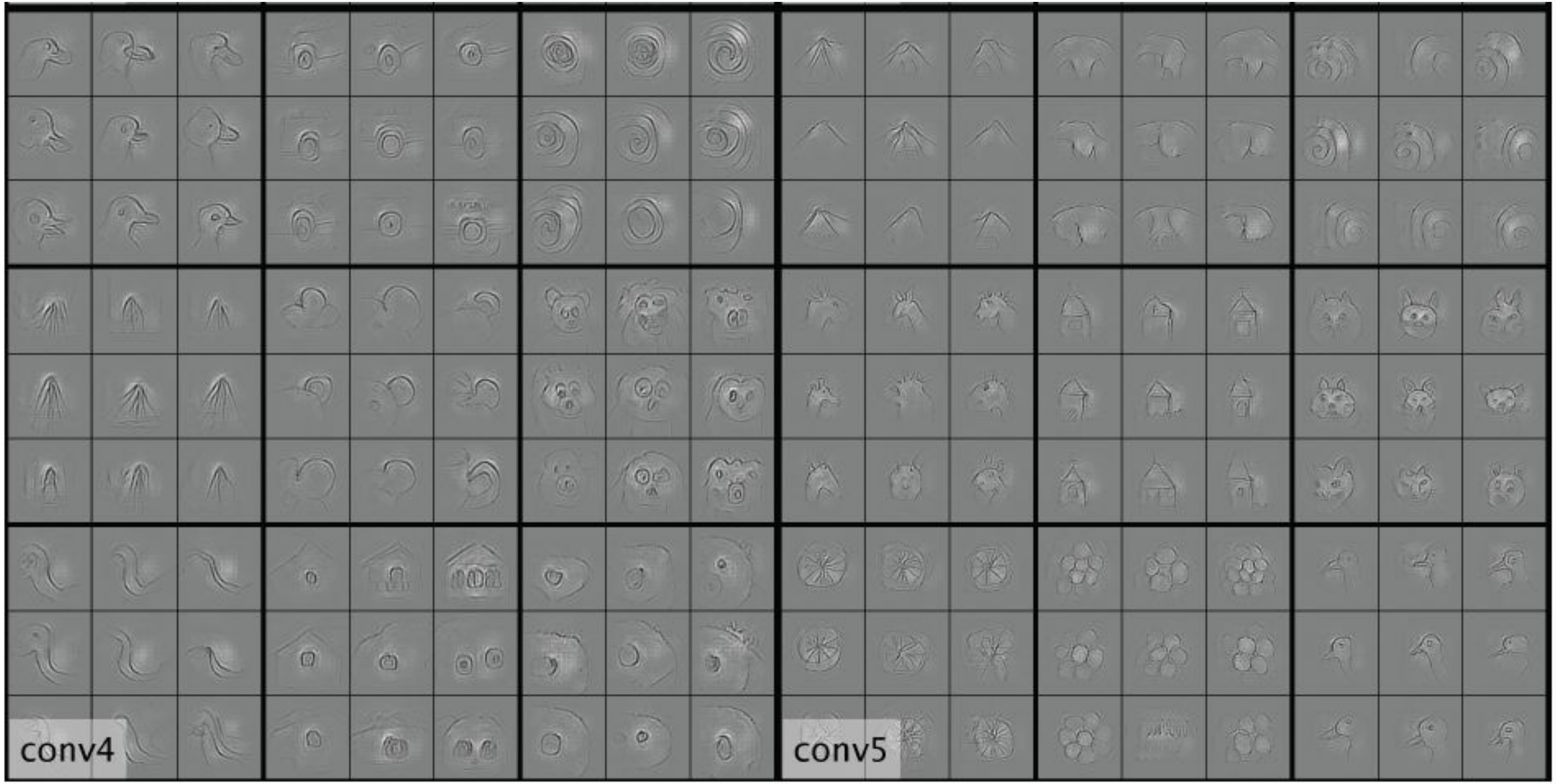
Fig. 8 Visualisation of the learned filters. Left: randomly selected filters from the first layer in our model; right: the real parts of some Gabor filters

Learns similar to biologically plausible gabor filter

Deconvnet Visualization : Lower Layers

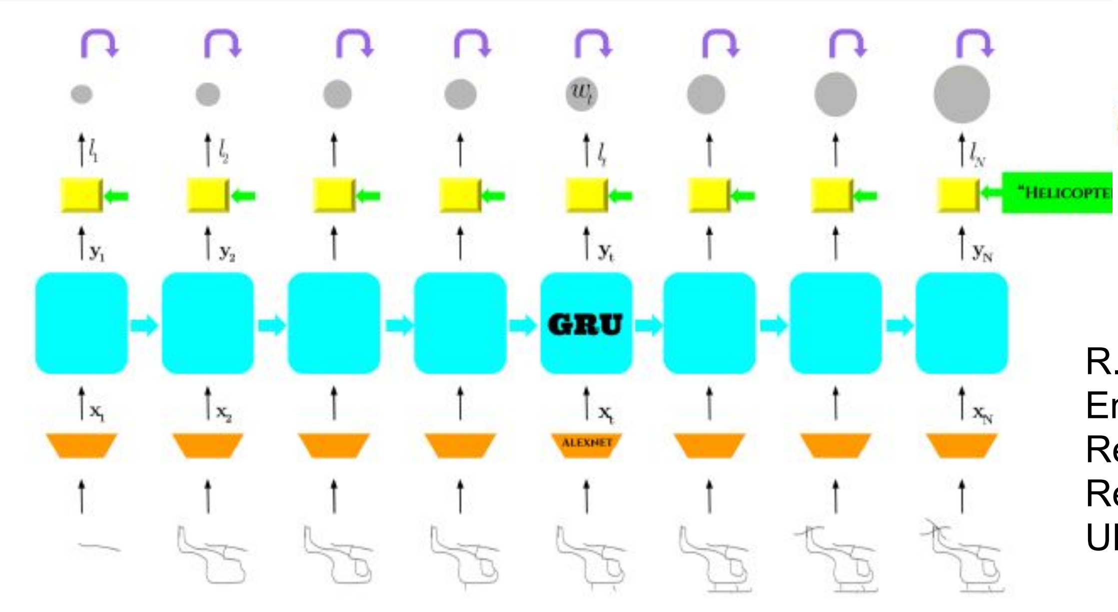


Deconvnet Visualization : Higher Layers



RNN for Sketch Recognition

- Sequential nature of stroke by stroke hand-sketching improves overall learning rate.
- GRU models sequential data in natural fashion.



$$r_t = \sigma(W_{xr}\mathbf{x_t} + W_{hr}h_{t-1} + b_r) \quad (1)$$
$$z_t = \sigma(W_{xz}\mathbf{x_t} + W_{hz}h_{t-1} + b_z) \quad (2)$$
$$\widetilde{h_t} = tanh(W_{xh}\mathbf{x_t} + U(r_t \odot h_{t-1}) + b_h) \quad (3)$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h_t} \quad (4)$$
$$\mathbf{y_t} = W_{hy}h_t \quad (5)$$

R.K. Sarvadevabhatla, J. Kundu & V. Babu R, Enabling My Robot To Play Pictionary: Recurrent Neural Networks For Sketch Recognition
URL: https://arxiv.org/pdf/1608.03369.pdf

Recognition Results for different networks

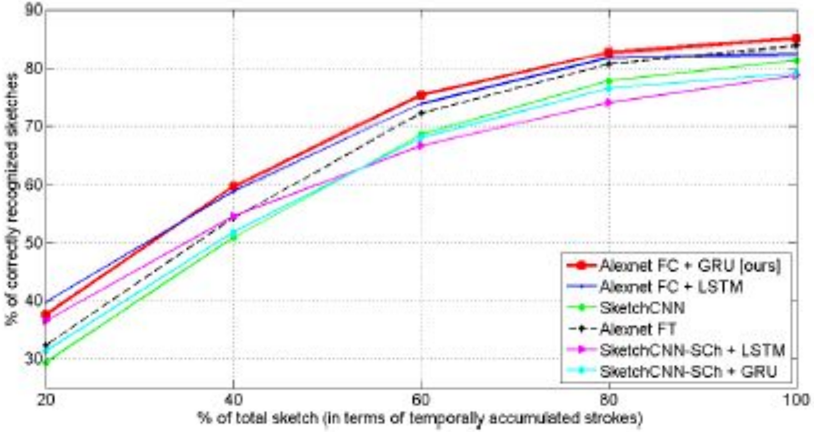


Figure 2: Comparison of online recognition performance for various classifiers. Our architecture recognizes the largest % of sketches at all levels of sketch completion. Best viewed in color.

CNN	Recurrent Network	#Hidden	Avg. Acc
Alexnet-FC	**GRU**	**3600**	**85.1%**
Alexnet-FC	LSTM	3600	82.5%
SketchCNN [23]	-	-	81.4%
Alexnet-FT	-	-	83.9%
SketchCNN-Sch-FC	LSTM	3600	78.8%
SketchCNN-Sch-FC	GRU	3600	79.1%

Table 1: Average recognition accuracy (rightmost column) for various architectures. #Hidden refers to the number of hidden units used in recurrent network. We obtain state-of-the-art results for sketch object recognition.

For next week ...

- Use existing visualization techniques (Zeiler et al, DeConv) to analyze sketch CNNs fine-tuned for sketches (AlexNet, VGG, GoogLeNet, ResNet) and Sketch-CNN (Matconvnet → Caffe)

Thank You