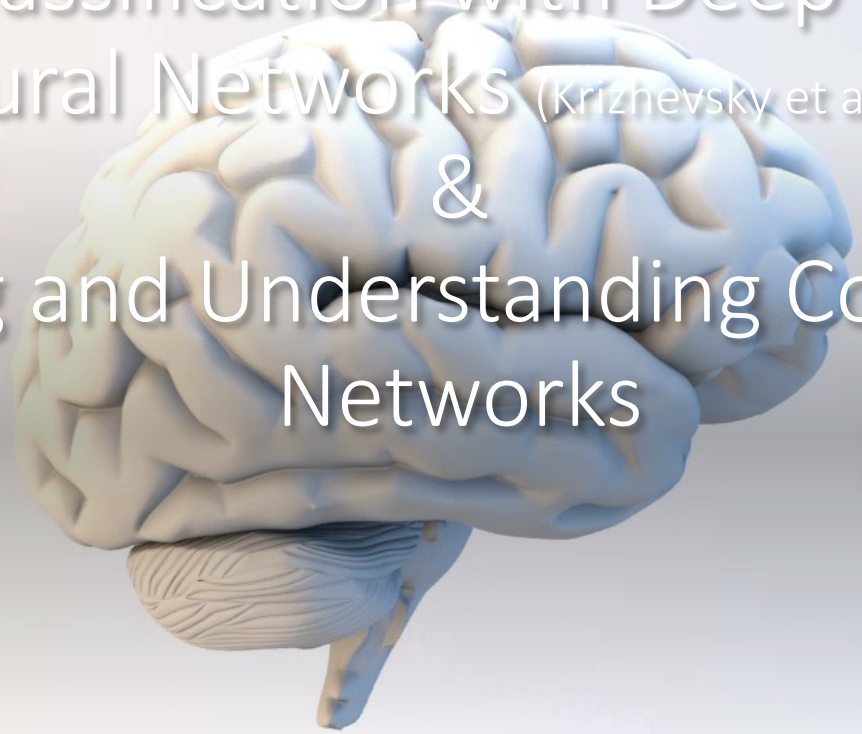
Algorithm Club:
ImageNet Classification with Deep Convolutional Neural Networks (Krizhevsky et al., 2012).
&
Visualizing and Understanding Convolutional Networks



Starting with ImageNet Classification with Deep Convolutional Neural Networks

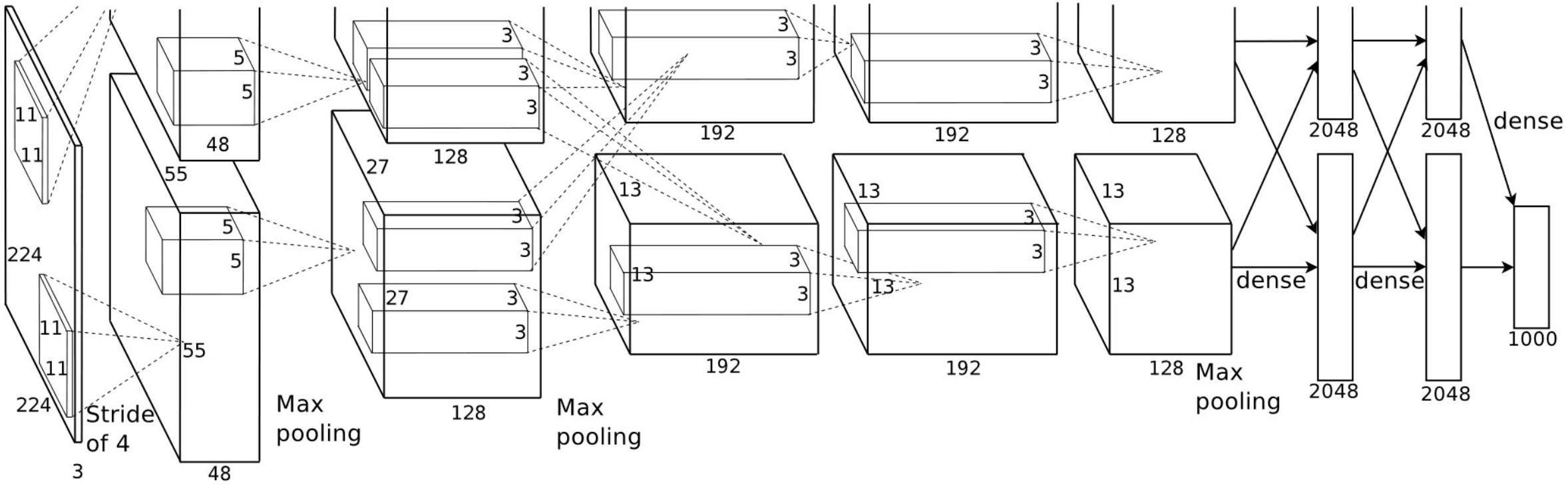
(Krizhevsky et al., 2012).

Introduction

- To learn about thousands of objects from **millions of images**, we need a model with a **large learning capacity**. Deep networks required.
- Deep networks prohibitively expensive to apply in large scale to high-resolution images. (2012 hardware and software)
- **CNNs** make strong and mostly correct assumptions about the nature of images, therefore **easier to train vs** feedforward neural networks (fewer connections and parameters).
- High power GPUs (2012) + optimized convolution facilitate **large/deep** CNNs training on large datasets (ImageNet) without severe overfitting.

Paper contributions

- Trained one of the **largest** convolutional neural networks to date on the subsets of ImageNet (2012)
- Achieved by far the best results ever reported on these datasets.
- Wrote a highly-optimized GPU implementation of 2D convolution.
- Used existing techniques to avoid overfitting in large networks (Drop out).



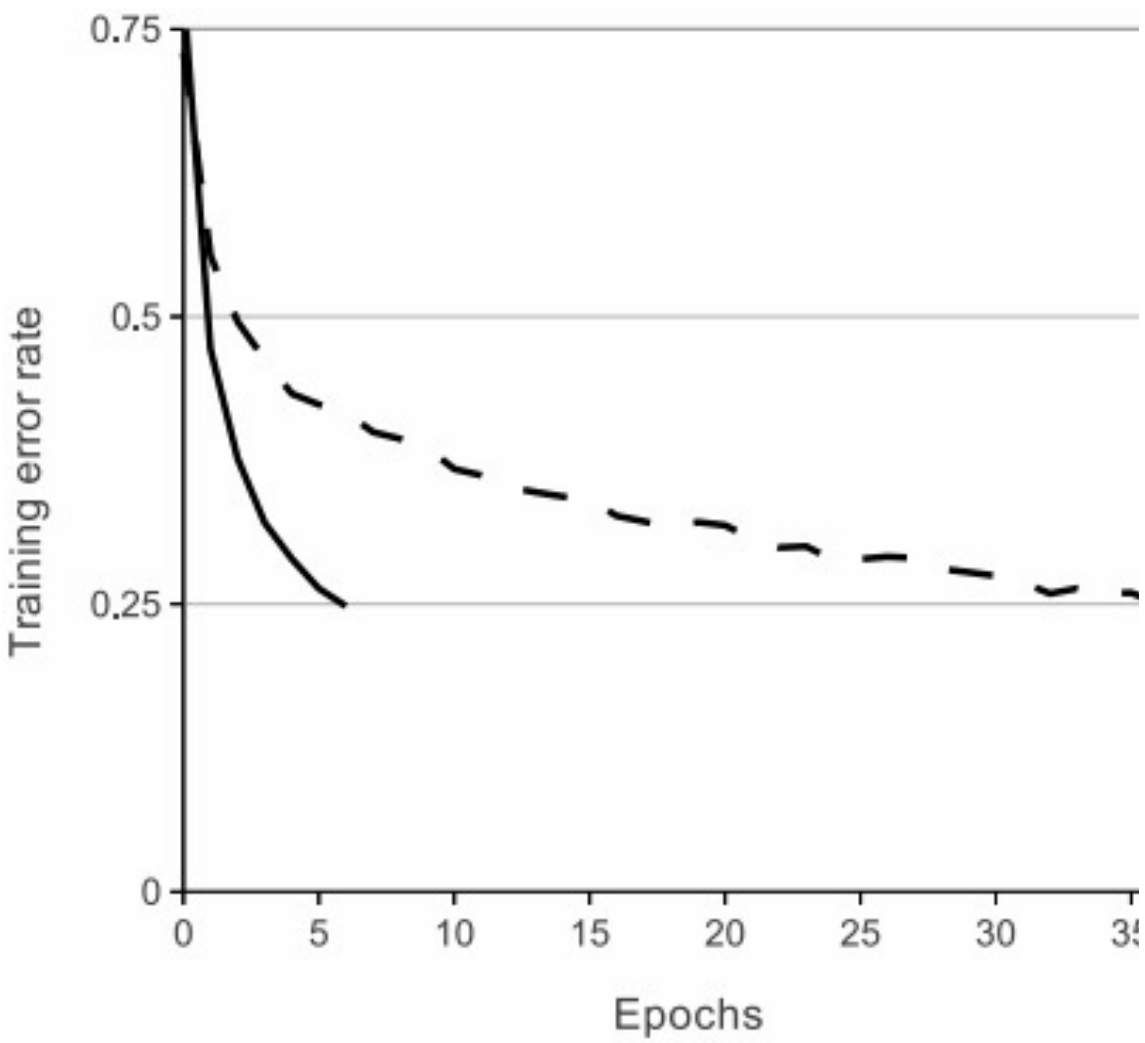
The Model

The net contains eight layers with weights; the first five are convolutional and the remaining three are fully connected.

The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

The Model: ReLU vs tanh

- ReLU: solid line
- tanh: dashed line
- 25% training error rate on CIFAR-10 six times faster.
- Faster learning has a great influence on the performance of large models trained on large datasets.
- They would not have been able to experiment with such large neural networks if they used traditional saturating neuron models.

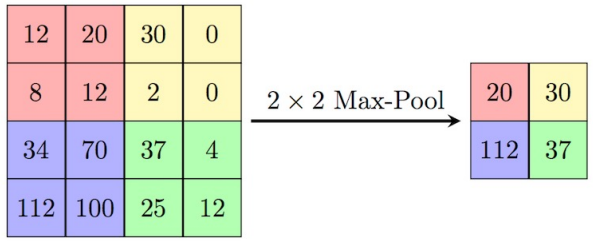


The Model continued

- Half of the kernels (or neurons) on each of 2 GPUs due to GPU memory constraints.
- Local Response Normalization

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^{\beta}$$

- Overlapping Max Pooling



Reducing Overfitting

- Data Augmentation:
  - Translations and horizontal reflections. Increases the training set by a factor of 2048.
  - Altering the intensities of the RGB (object identity is invariant to changes in the intensity and colour of the illumination).
- Dropout: setting to zero the output of each hidden neuron with probability 0.5. (reduces complex co-adaptations of neurons)

Results

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**

ILSVRC2010 test set.

Model	Top-1	Top-5
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

ILSVRC-2012 validation and test sets.

In italics are best results achieved by others.
Models with an asterisk* were “pre-trained”.

Finally

- Depth is important: removing any convolutional layer resulted in inferior performance.
- Deep convolutional neural network is capable of achieving record-breaking results on a highly challenging dataset using purely supervised learning.
- Results improve as we have made our network larger and trained it longer. Suggested improvements: faster GPUs and bigger datasets.

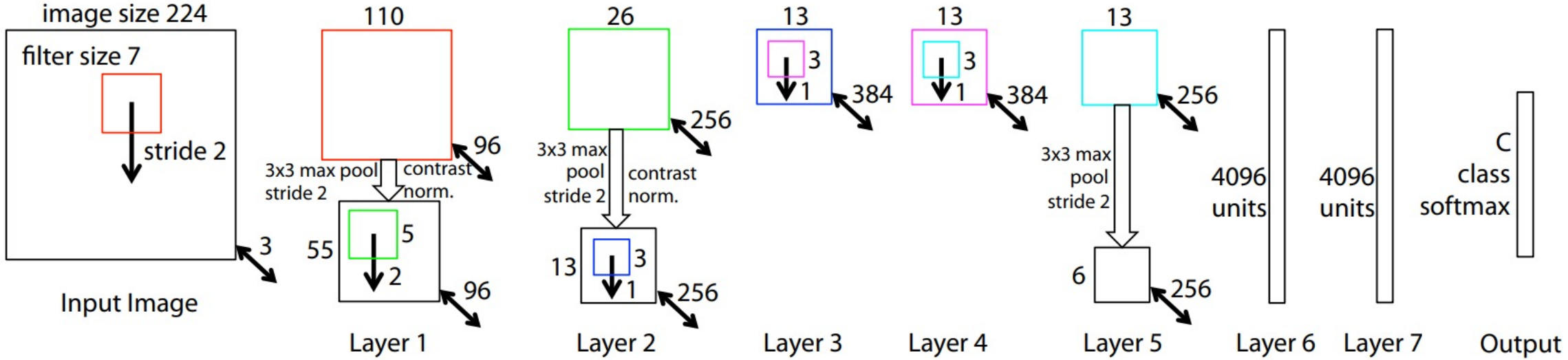
Moving on to…Visualizing and Understanding Convolutional Networks

Introduction

- Renewed interest in convnet models since large models practical because of newly available:
  - Larger training sets with millions of labels.
  - Powerful GPU implementations.
  - Better regularization strategies, such as Dropout (reduce overfitting).
- (Krizhevsky et al., 2012) => demonstrated that large Convolutional Network models have impressive classification performance on the ImageNet benchmark.
- Need to **understand why they perform so well**, or how they might be improved.
- Deconvolutional Network (**deconvnet**) to project feature activations back to the input pixel space.

Paper contributions

- Novel visualization technique that gives insight into the function of intermediate feature layers and the operation of the classifier.
- Used in a diagnostic role, visualization allow us to find model architectures that outperform Krizhevsky et al.
- Provide a lot of analysis.



Changed from Krizhevsky et al.:

1. Reduced the 1st layer filter size from 11x11 to 7x7
2. Made the stride of the convolution 2, rather than 4.
3. Single GPU vs dual GPU in Krizhevsky et al.: Dense rather than sparce connections.

The Model

8-layer convnet model.

A 224 by 224 crop of an image (with 3 color planes) is presented as the input. Convolved with 96 different 1st layer filters (red), each of size 7 by 7, using a stride of 2.

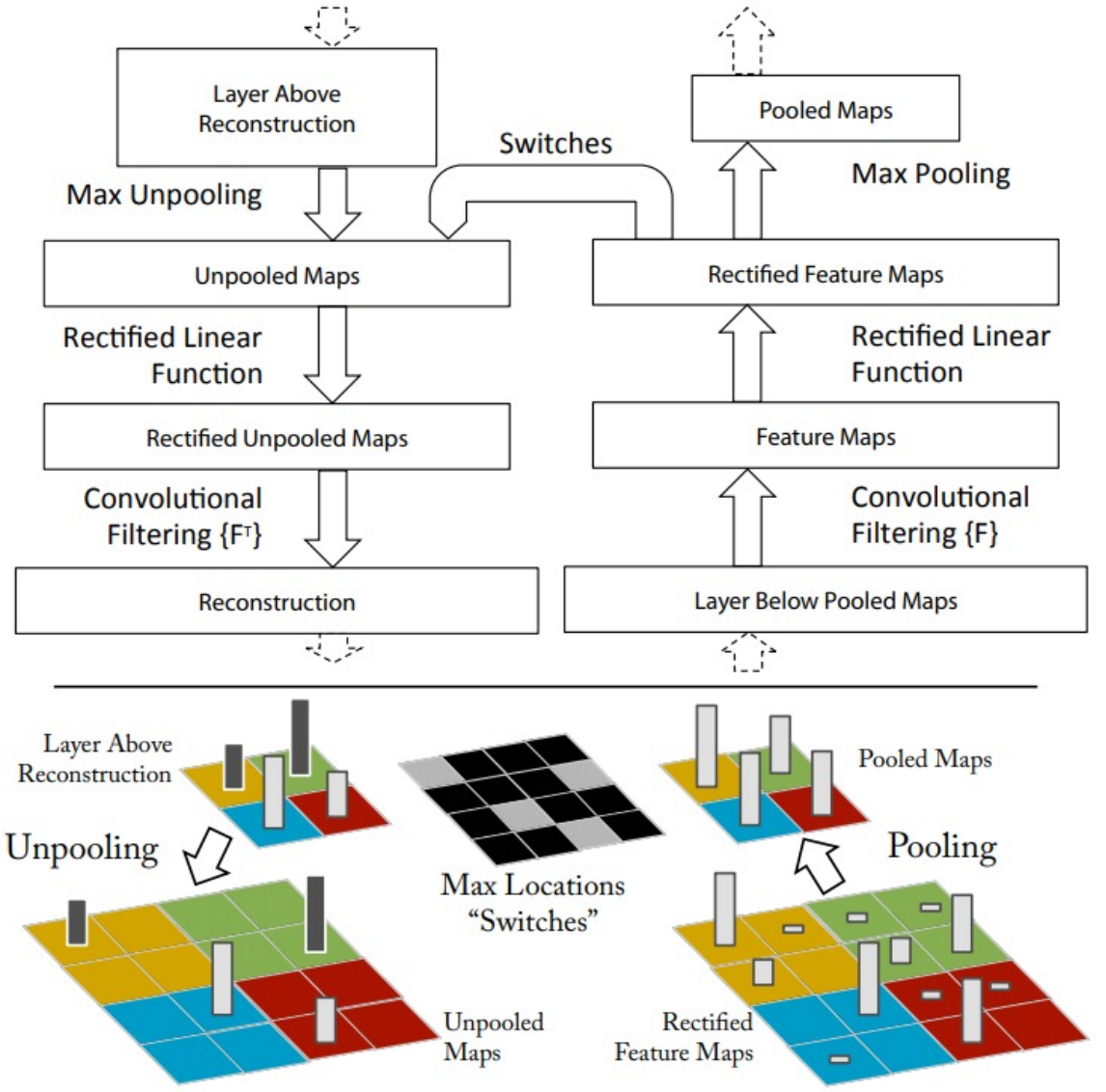
Then, passed through a rectified linear function (not shown), pooled (max within 3x3 regions, using stride 2) and contrast normalized across feature maps to give 96 different 55 by 55 element feature maps.

Similar operations are repeated in layers 2,3,4,5.

The last two layers are fully connected, taking features from the top convolutional layer as input in vector form ($6 \cdot 6 \cdot 256 = 9216$ dimensions). The final layer is a C-way SoftMax function, C being the number of classes. All filters and feature maps are square in shape.

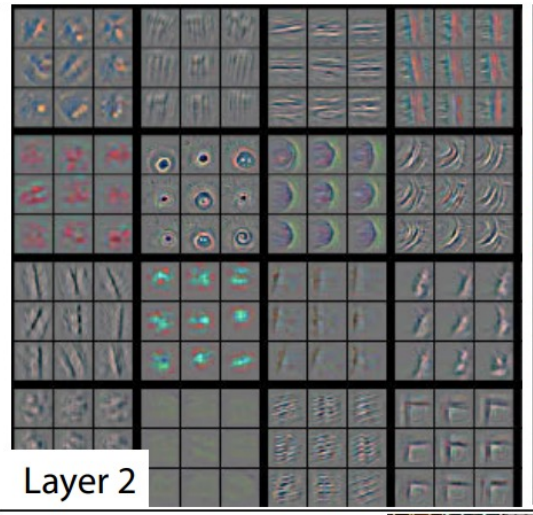
Deconvnet: Approach

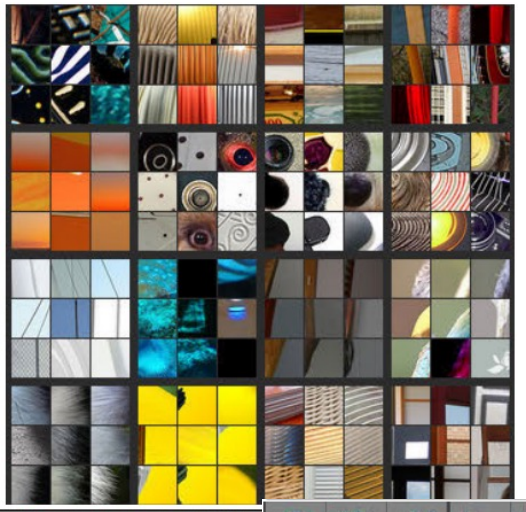
- Unpooling: max pooling operation is non-invertible so we need to recording the locations of the maxima (“Switches” in diagram) for an approximate inverse.
- Rectification: pass through ReLU.
- Filtering: transposed versions of the same filters. (flipping each filter vertically and horizontally)

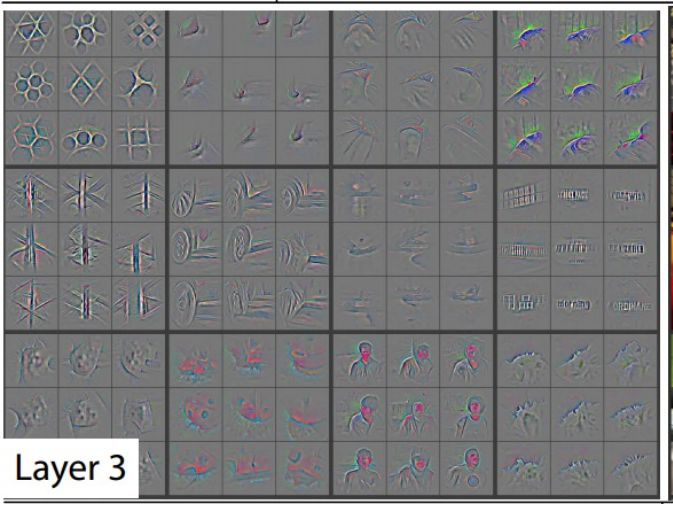


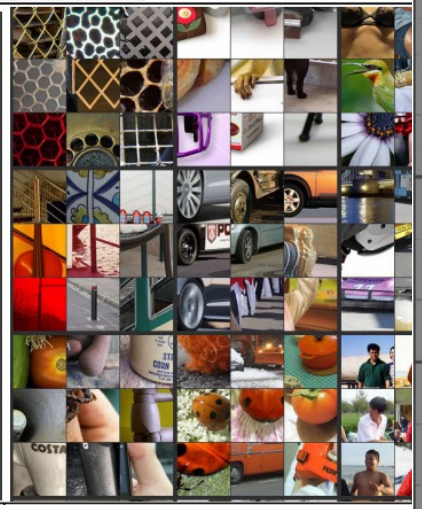
Deconvnet: Visualisations

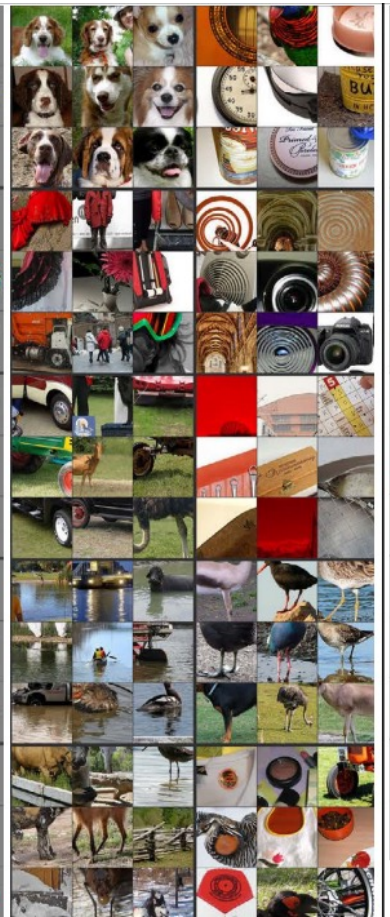
Top 9 activations in a random subset of feature maps projected down to pixel space.

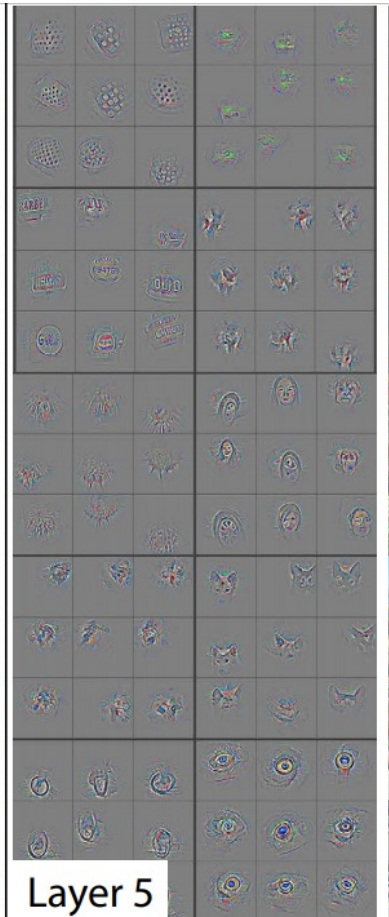








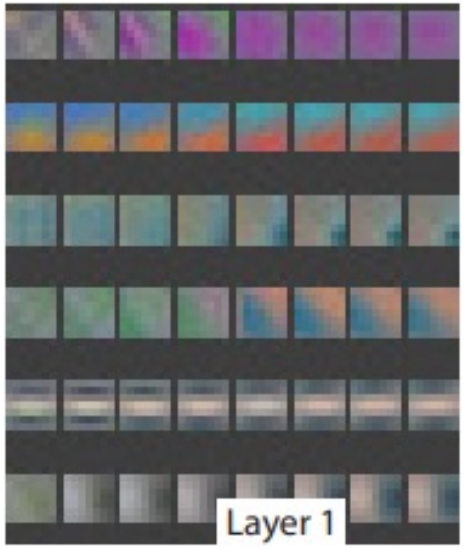


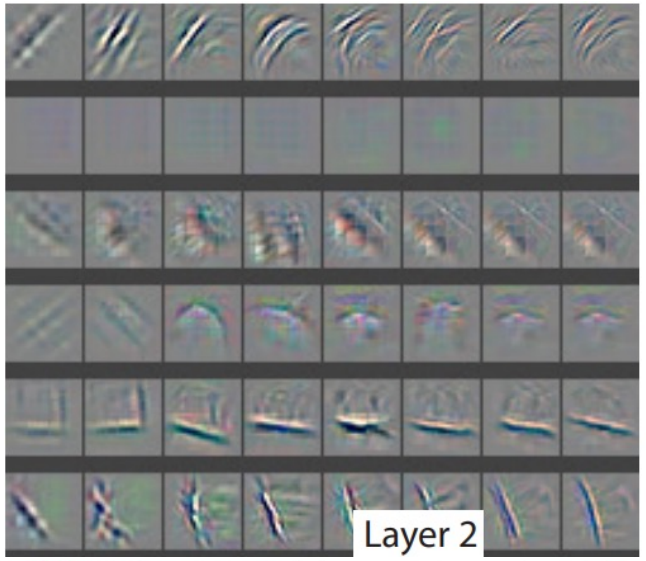


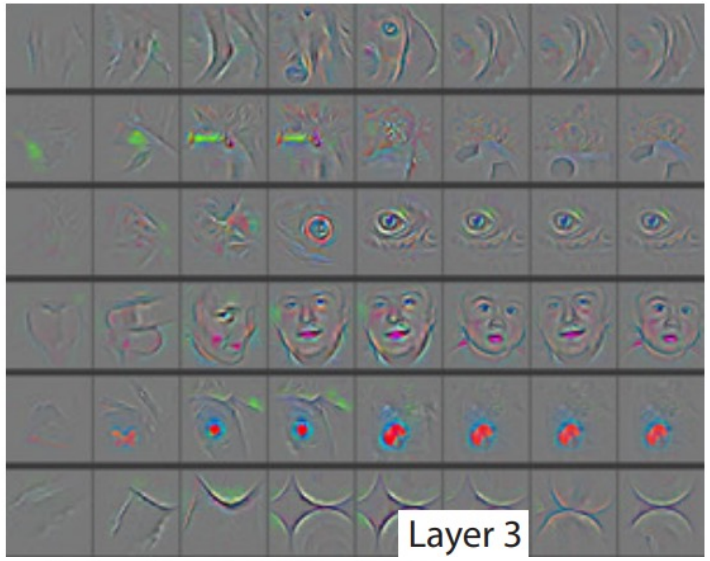
Feature map example: layer 5 r1c2 focuses on background grass.

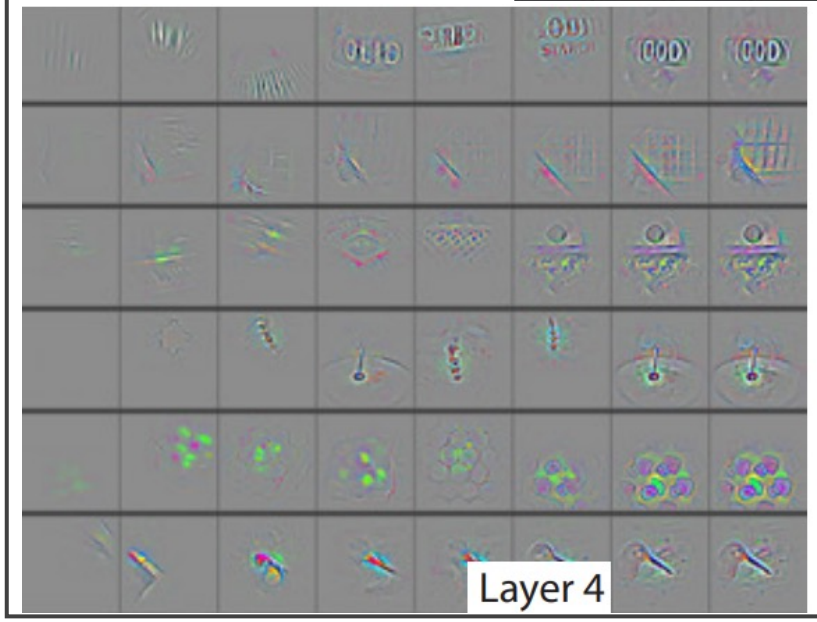
The deeper layers find more complex features.

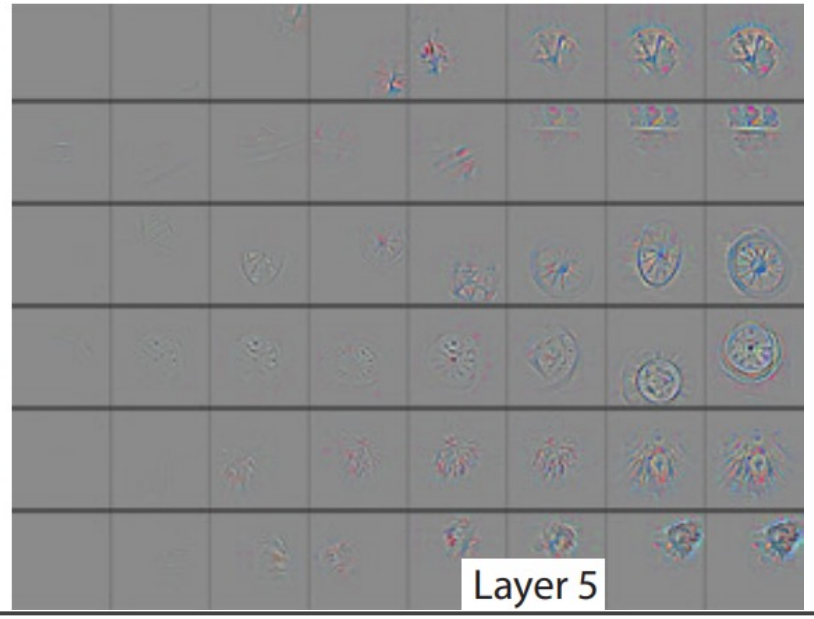
Feature Evolution during Training











Columns =

epochs [1,2,5,10,20,30,40,64]

Upper layers develop later in training (e.g., layer 5)

Its important to train enough for convergence.

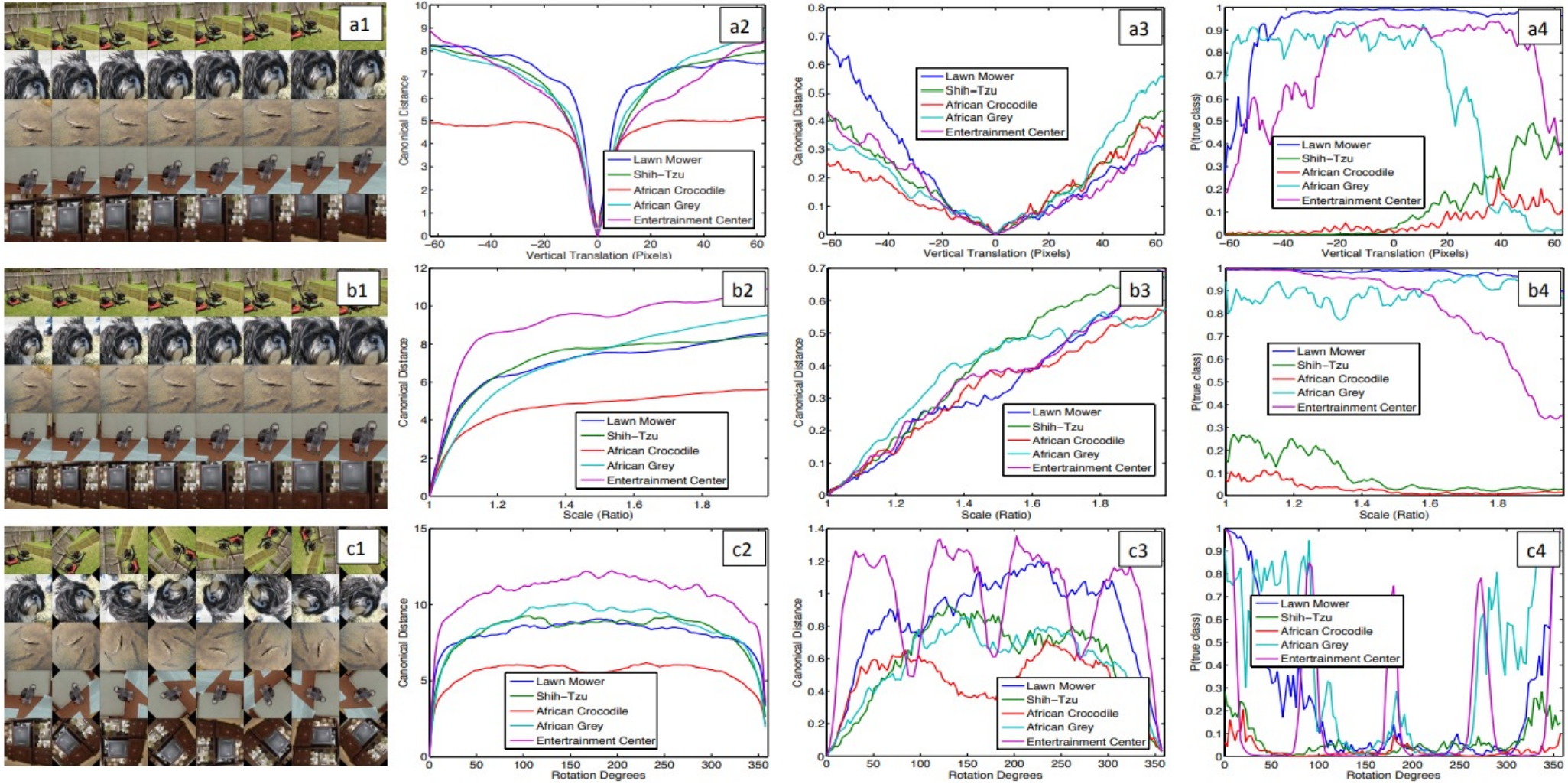
Feature Invariance

Analysis of:

a) Vertical translation

b) Scale

c) Rotation invariance



Example images undergoing the transformations

Euclidean distance between feature vectors from the original and transformed images in:

Layer 2 (more distance)

Layer 7 (less distance)

The probability of the true label

Feature Invariance

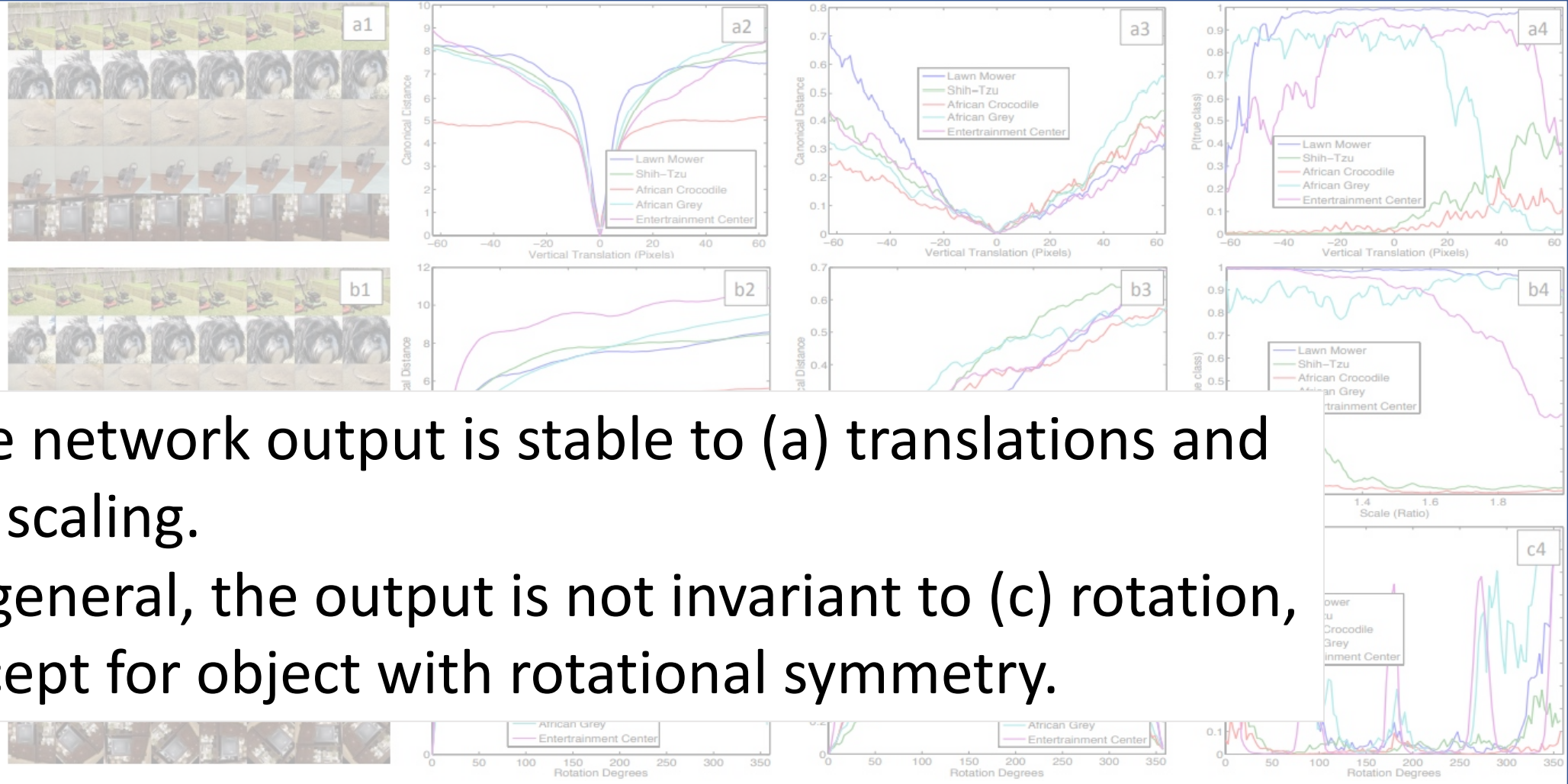
Analysis of:

a) Vertical translati

b) Scale

c) Rotatio invarian

The network output is stable to (a) translations and (b) scaling.
In general, the output is not invariant to (c) rotation, except for object with rotational symmetry.



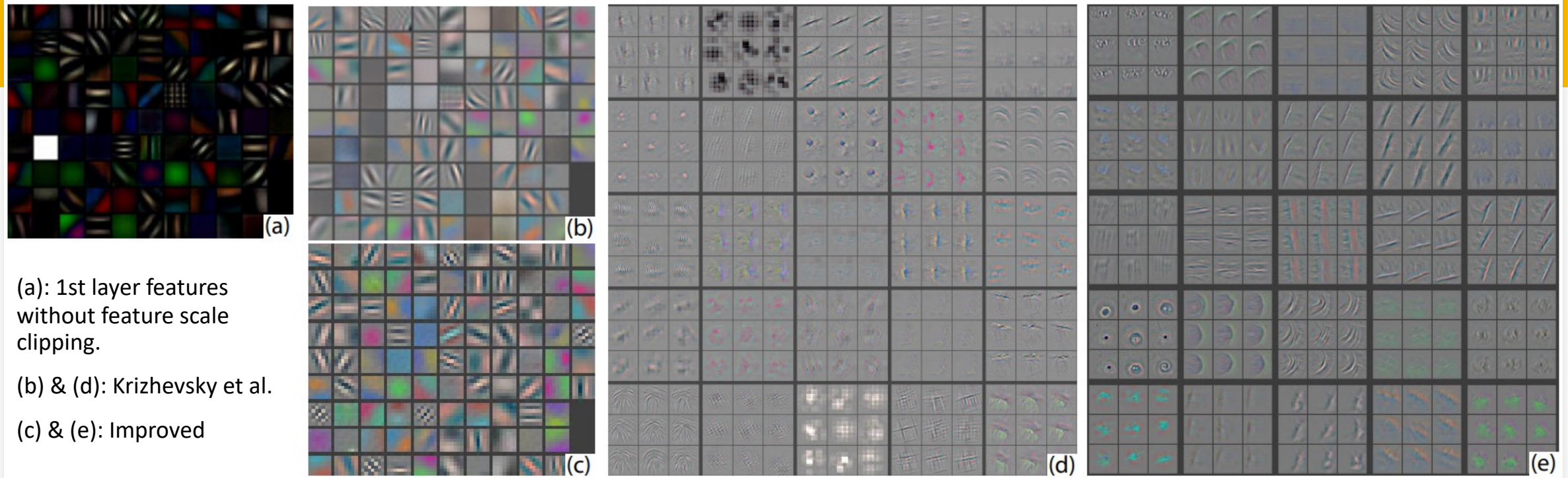
Example images undergoing the transformations

Euclidean distance between feature vectors from the original and transformed images in:

Layer 2 (more distance)

Layer 7 (less distance)

The probability of the true label



(a): 1st layer features without feature scale clipping.

(b) & (d): Krizhevsky et al.

(c) & (e): Improved

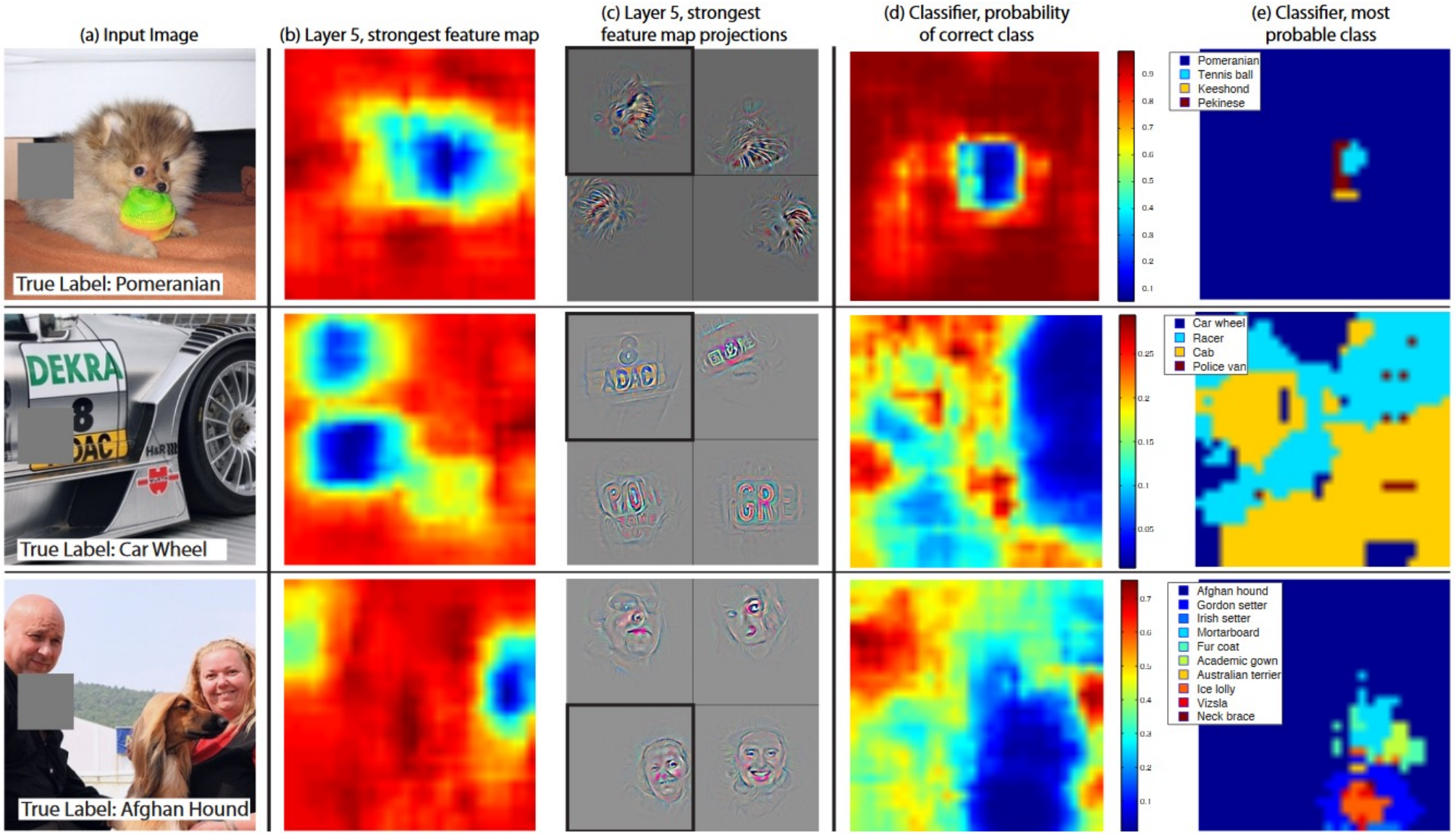
Using visualisations to improve Krizhevsky et al.

- In Krizhevsky et al., the first layer filters are a mix of extremely high and low frequency information, with little coverage of the mid frequencies (b).
- Additionally, the 2nd layer visualization shows aliasing artifacts caused by the large stride 4 used in the 1st layer convolutions.
- Remedied by (i) reducing the 1st layer filter size from 11x11 to 7x7 and (ii) making the stride of the convolution 2, rather than 4.
- The improved architecture (c&e) retain much more information in the 1st and 2nd layer features.
- Importantly, improved classification performance as shown later.

Occlusion Sensitivity

Test examples systematically covered by a grey square

The examples show the model is localising objects, as probability of the correct class drops significantly when the object is occluded.

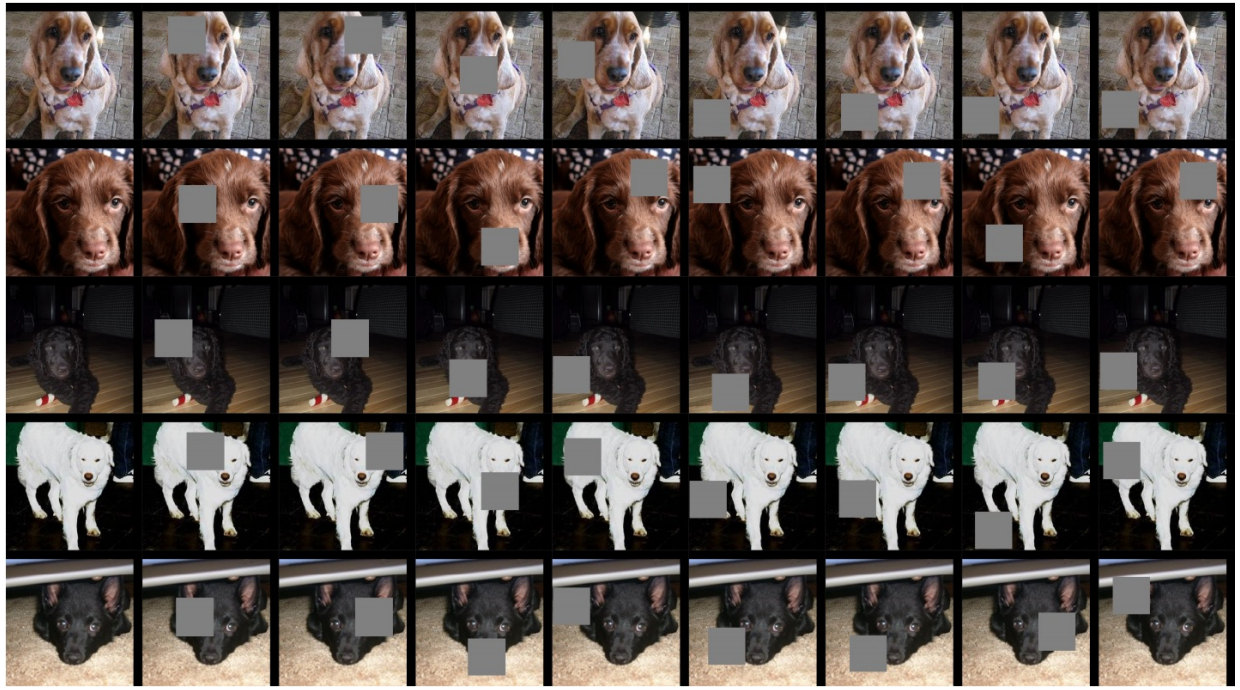


Layer 5 feature maps ((b) & (c))
Black square = strongest activation feature's projection.

Occlusion causes classifier probability to drop

Correspondence Analysis: correspondence between specific object parts in different images

(e.g. faces have a particular spatial configuration of the eyes and nose).



Col 1: Original image. Col 2,3,4: Occlusion of the right eye, left eye, and nose respectively. Other columns show examples of random occlusions.

Occlusion Location	Mean Feature Sign Change Layer 5	Mean Feature Sign Change Layer 7
Right Eye	0.067 ± 0.007	0.069 ± 0.015
Left Eye	0.069 ± 0.007	0.068 ± 0.013
Nose	0.079 ± 0.017	0.069 ± 0.011
Random	0.107 ± 0.017	0.073 ± 0.014

The lower scores for the eyes and nose (compared to random object parts) show the model implicitly establishing some form of correspondence.

At layer 7, the scores are more similar, perhaps due to upper layers trying to discriminate between the different breeds of dog.

Error %	Val Top-1	Val Top-5	Test Top-5
(Gunji et al., 2012)	-	-	26.2
(Krizhevsky et al., 2012), 1 convnet	40.7	18.2	−−
(Krizhevsky et al., 2012), 5 convnets	38.1	16.4	16.4
(Krizhevsky et al., 2012)*, 1 convnets	39.0	16.6	−−
(Krizhevsky et al., 2012)*, 7 convnets	36.7	15.4	15.3
Our replication of (Krizhevsky et al., 2012), 1 convnet	40.5	18.1	−−
1 convnet as per Fig. 3	38.4	16.5	−−
5 convnets as per Fig. 3 – (a)	36.7	15.3	15.3
1 convnet as per Fig. 3 but with layers 3,4,5: 512,1024,512 maps – (b)	37.5	16.0	16.1
6 convnets, (a) & (b) combined	**36.0**	**14.7**	**14.8**

Experiments

- ImageNet 2012 dataset: 1.3M/50k/100k training/validation/test examples, spread over 1000 categories.
- Reproduced Krizhevskyet al., 2012 model => error rate within 0.1% of their reported value.
- Improved model => better by 1.7% (test top-5).
- Combine multiple models => 14.8%
- This error is almost half of the top non-convnet entry in the ImageNet 2012 classification challenge, which obtained 26.2% error.

Error %	Train Top-1	Val Top-1	Val Top-5
Our replication of (Krizhevsky et al., 2012), 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

Experiments Continued…

Removing both the middle convolution layers and the fully connected layers yields a model with only 4 layers whose performance is dramatically worse.

This would suggest that the overall depth of the model is important for obtaining good performance.

Changing the size of the fully connected layers makes little difference to performance. However, increasing the size of the middle convolution layers goes give a useful gain in performance.

Increasing size of both middle convolution layers & the fully connected layers **results in over-fitting**.

The experiments show the importance of the convolutional part of the model in obtaining state-of-the-art performance.

Supported by the visualizations which show the complex invariances learned in the convolutional layers.

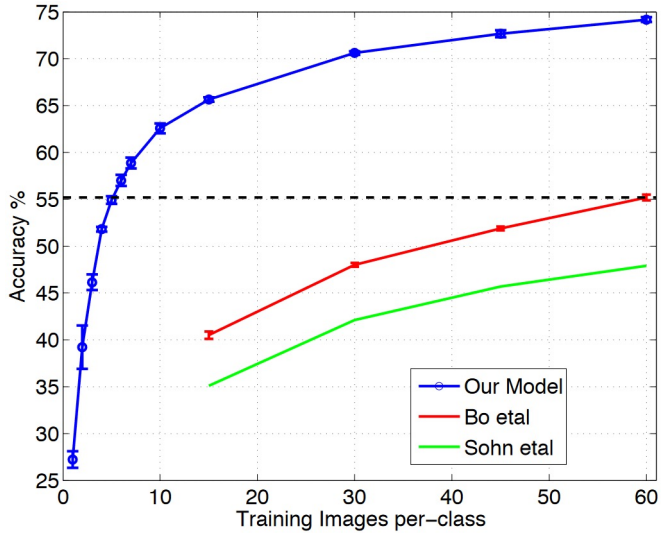


Figure 9. Caltech-256 classification performance as the number of training images per class is varied. Using only 6 training examples per class with our pre-trained feature extractor, we surpass best reported result by (Bo et al., 2013).

# Train	Acc % 15/class	Acc % 30/class	Acc % 45/class	Acc % 60/class
(Sohn et al., 2011)	35.1	42.1	45.7	47.9
(Bo et al., 2013)	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.	$\mathbf{65.7 \pm 0.2}$	$\mathbf{70.6 \pm 0.2}$	$\mathbf{72.7 \pm 0.4}$	$\mathbf{74.2 \pm 0.3}$

Table 5. Caltech 256 classification accuracies.

# Train	Acc % 15/class	Acc % 30/class
(Bo et al., 2013)	–	81.4 ± 0.33
(Jianchao et al., 2009)	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-101 classification accuracy for our convnet models, against two leading alternate approaches.

Acc %	[B]	Ours	Acc %	[B]	Ours
Airplane	**97.3**	96.0	Dining tab	**77.8**	67.7
Bicycle	**84.2**	77.1	Dog	83.0	**87.8**
Bird	80.8	**88.4**	Horse	**87.5**	86.0
Boat	85.3	**85.5**	Motorbike	**90.1**	85.1
Bottle	**60.8**	55.8	Person	**95.0**	90.9
Bus	**89.9**	85.8	Potted pl	**57.8**	52.2
Car	**86.8**	78.6	Sheep	79.2	**83.6**
Cat	89.3	**91.2**	Sofa	**73.4**	61.1
Chair	**75.4**	65.0	Train	**94.5**	91.8
Cow	**77.8**	74.4	Tv	**80.7**	76.1
Mean	**82.2**	79.0	# won	**15**	5

PASCAL 2012 classification results, comparing our ImageNet-pretrained convnet against the leading two methods ([A]= (Sande et al., 2012) and [B] = (Yan et al.,2012)).

Model Transfer

Explore the ability of the feature extraction layers to generalize to other datasets, namely Caltech-101, Caltech-256 and PASCAL VOC 2012.

To do this, layers 1-7 of the ImageNet-trained model are kept fixed while a new SoftMax classifier is trained on top using the training images of the new dataset.

	Cal-101 (30/class)	Cal-256 (60/class)
SVM (1)	44.8 ± 0.7	24.6 ± 0.4
SVM (2)	66.2 ± 0.5	39.6 ± 0.3
SVM (3)	72.3 ± 0.4	46.0 ± 0.3
SVM (4)	76.6 ± 0.4	51.3 ± 0.1
SVM (5)	$\mathbf{86.2 \pm 0.8}$	65.6 ± 0.3
SVM (7)	$\mathbf{85.5 \pm 0.4}$	$\mathbf{71.7 \pm 0.2}$
Softmax (5)	82.9 ± 0.4	65.7 ± 0.5
Softmax (7)	$\mathbf{85.4 \pm 0.4}$	$\mathbf{72.6 \pm 0.1}$

Table 7. Analysis of the discriminative information contained in each layer of feature maps within our ImageNet-pretrained convnet. We train either a linear SVM or softmax on features from different layers (as indicated in brackets) from the convnet. Higher layers generally produce more discriminative features.

Hierarchical feature structures

This supports the premise that as the feature hierarchies become deeper, they learn increasingly powerful features.

Finally

- Explored large convolutional neural network models, trained for image classification.
- Presented novel visualisation method for activity within the model.
- Visualisation reveals features are not random, they contain intuitively desirable properties such as compositionality, increasing invariance and class discrimination as we ascend the layers.
- Visualisation to debug problems with a model: obtain better results.
- Demonstrated through occlusions: model is highly sensitive to local structure, not just using broad scene context.
- Ablation study revealed minimum depth to the network, rather than any individual section of layers.
- Showed how the ImageNet trained model can generalise well to other datasets beating the best Caltech-101 and Caltech-256 reported results, in the latter case by a significant margin.
- Generalized less well to the PASCAL data, but might improve with a loss function that per-mitted multiple objects per image.