2022.01.24

Visualizing and Understanding Convolutional Networks

-Matthew D. Zeiler , Rob Fergus

Abstract

큰 Convolutional Network 모델은 최근에 이미지넷 벤치마킹에서 훌륭한 분류 성능을 말해준다. 하지만 왜 성능이 좋은지에 대해 이해를 알 수가 없어서 이 논문에서 말하고자 한다. 중간 특성층과 분류기의 작동에 insight를 주는 새로운 시각화 기술을 도입한다. 분석에서 사용되는 이것은 이미지넷 분류기 벤치마크를 능가하는 모델 구조를 찾게 해준다. 또한 다른 모형 층에서의 모형 공헌도를 발견하는 ablation 학습(모델이나 알고리즘의 특성을 제거해가면서 그게 퍼포먼스에 어떠한 영향을 미치는지 확인하는 것)을 수행한다. ImageNet이 다른 데이터셋에 대해서도 잘 일반화된다는 것을 보여준다. 소프트맥스 분류기가 유지될 때, 이 모델은 확실히 현재의 결과를 능가할 수 있다.

1. Introduction

Convolution network는 숫자 손글씨를 분류하는 것이나 face-detection을 매우 잘 수행한다. 과거에 몇몇 논문들은 더 어려워진 시각적으로 분류하는 일에 더 좋은 성능을 보일 수 있다고 보여왔다. 그중에서 ImageNet에서 좋은 성능을 낸 모델이 주목할 만하다. 두 가지 요점들이 관심 대상이다. (1) 더 큰 훈련 데이터셋을 이용가능한가? (2) 강력한 GPU 사용 (3) 더 나은 모델 정규화 전략. 이러한 좋은 점들에도 불가하고 몇 가지 주안점이 있다. 어떻게 AlexNet 모델이 정확도를 잘 확보하였는지에 대한 설명이 확실하지 않으므로 visualization을 활용하여 설명을 해보겠다. 이러한 방법은 feature activations를 input으로 돌려내는 Deconvolutional Network를 활용한다. 또한 sensirivity analysis를 하여 input 이미지의 어떤 부분이 분류기에서 중요하게 작용하는지 파악해보겠다.

1.1 Related Work

feature들을 시각화하는 것은 흔한 기법이지만 픽셀 공간으로 투영이 가능한 첫 번째 층까지가 한계이다. 깊은 층까지도 접근해보려는 시도들도 많았지만 불변성이 복잡하다는 것이었다. 반대로 저자는 불변성에 비모수적 관점을 제공하였고 이는 훈련 set에서의 패턴들이 피쳐맵을 활성화하는 것을 보여준다.

2. Approach

Standard fully supervised convnet model을 사용하겠다. Input image를 받고 C개의 다른 클래스들에 대해 확률 벡터를 출력한다. 각 층은 learned filters와 이전 layer ouput과의 합성곱과 ReLU 함수를 통과한다. 그리고 max pooling을 시행하고 local response normalize를 시행한다. 이미지와 라벨을 이용해 학습하고 크로스 엔트로피가 손실 함수로 들어간다. 모수들의 최적값은 역전파 알고리즘을 통해 구해진다.

2.1 Visualization with a Deconvnet

Convnet의 작동을 이해하는 것은 중간 층에서의 feature activity를 해석하는 것을 요구한다. 이러한 activities로부터 input 이미지를 다시 구성하는 방법을 제시하였고 input 이미지의 어떤 부분이 feature map에서 activation을 유발하는지 보여준다. 이러한 것을 Deconvolution Net과 함께 수행하였다. Deconvnet은 convnet과 구성 요소는 같지만 반대로 일을 한다고 보면 된다. 비지도 학습을 수행하는 방식으로 제공되었다.

Convnet을 시험하기 위해 deconvent은 각가의 layers에 붙어있어 이미지 픽셀로 연속적인 path를 다시 제공한다. 주어진 convnet activation을 시험하기 위해 다른 모든 activation을 0으로 세팅하고 feature 맵을 통과한다. 이후 unpool과 rectify, filter를 통과하여 activity를 재구성한다. 인푹 픽셀 공간에 도달할 때까지 반복한다.

- Unpooling: max pooling은 역변환이 안 된다. 하지만 switch 변수를 활용하여 높은 값을 보인 위치를 기억함으로써 근사적 역변환을 할 수 있다. deconvent에서, unpooling은 switch를 사용해 자극을 보존하면서 layer에서 특정한 위치 위로 reconstruction을 위치한다.

- Rectification: Convnet은 relu 비선형을 사용하는데 feature map이 언제 양의 값을 갖도로 한다. 유효한 feature reconstruction을 얻기 위해 reconstructed signal을 relu 비선형에 통과시킨다.

-Filtering: Convnet은 learned filter를 사용해 feature map을 이전 층으로부터 convolve시킨다. 이 과정을 거꾸로 하면 deconvnet은 같은 필터의 transpose된 버전을 사용한다. 하지만 이는 rectified map에 적용되고 layer 아래의 output에는 적용되지 않는다. 이것은 filter를 vertical하고 horizontal하게 flipping하는 것을 의미한다.

Higher층으로부터 투영하는 것은 max pooling으로 인해 생성된 switch setting을 사용한다. Switch setting이 주어진 이미지에 대해 독특하기에 single activation으로 인해 얻어진 reconstruction은 원래 input 이미지의 일부와 닮는다. 구조들은 feature activation에 기여한 것을 토대로 가중치되어 있다. 모델이 차별적으로 훈련되어 있기에 input 이미지의 부분들도 차별적임을 보여준다. 생

성과정이 포함되어 있지 않기에 이러한 투영은 모델로부터의 샘플이 아닌 것을 알아두자.

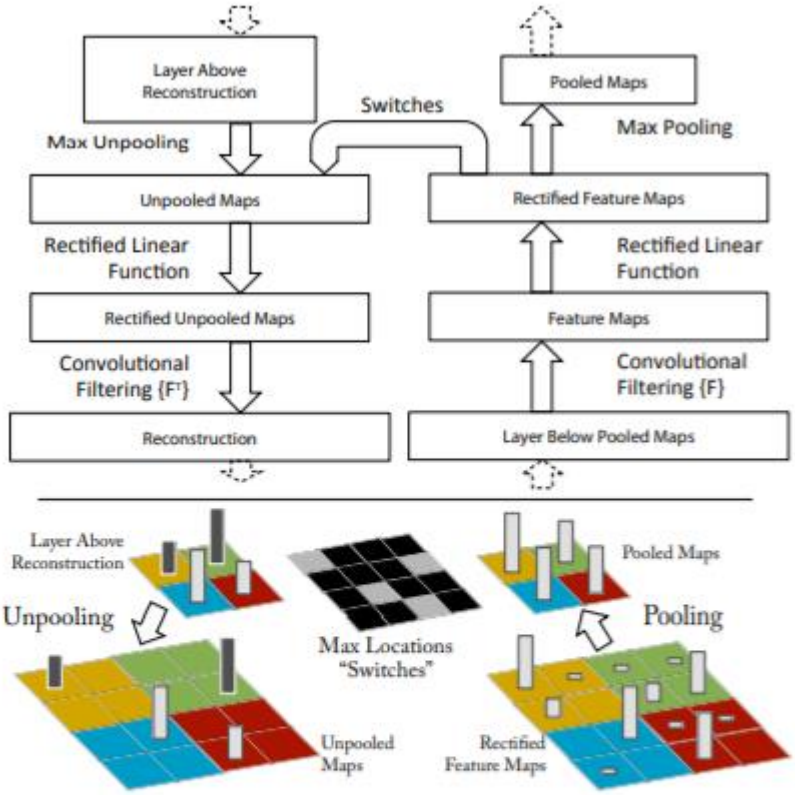


Figure 1. Top: A deconvnet layer (left) attached to a convnet layer (right). The deconvnet will reconstruct an approximate version of the convnet features from the layer beneath. Bottom: An illustration of the unpooling operation in the deconvnet, using *switches* which record the location of the local max in each pooling region (colored zones) during pooling in the convnet.

3. Training Details

AlexNet과 구조가 비슷한데 하나의 차이는 layer 3, 4, 5에서 쓰이는 희소 연결이 dense connections로 대체된다는 것이다. 모델은 2012 ImagNet에서 사용된 훈련용 set를 사용한다. 각 RGB 이미지는 가장 작은 차원인 256으로 정제된다. 128개의 미니 배치와 함께 확률적 경사 하강법이 사용된다. 처음 learning rate는 0.01로 셋팅된다. Validation error가 정체될 때까지 학습률을 갱신시키면서 반복한다. Dropout이 FC layers에서 0.5로 사용된다. 모든 가중치들은 0.01로 절편들은 0으로 초기화된다. 훈련 도중 첫 번째 계층의 시각화는 그들 중 조금이 지배한다. 이것을 해결하기 위해 우리는 convolutional layers에 있는 필터 각각을 재정규화한다. 이것은 인풋 이미지가 대강 [-128,128] 범위에 있는 모델의 첫 번째 층에서 특히 중요하다. 70 epochs를 진행했고 하나의 GPU로 12일 정도가 걸렸다.

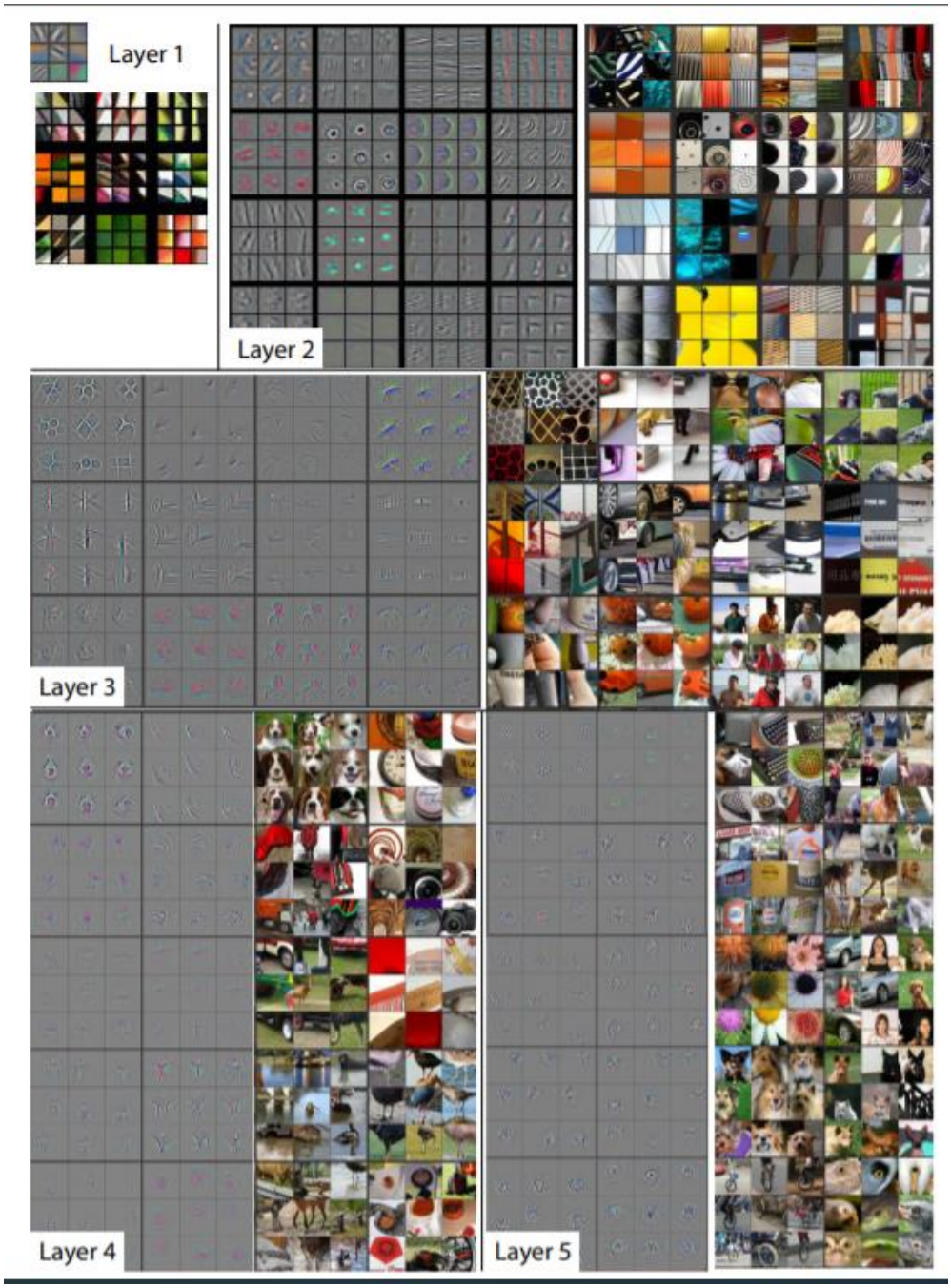


Figure 2. Visualization of features in a fully trained model. For layers 2-5 we show the top 9 activations in a random subset of feature maps across the validation data, projected down to pixel space using our deconvolutional network approach. Our reconstructions are *not* samples from the model: they are reconstructed patterns from the validation set that cause high activations in a given feature map. For each feature map we also show the corresponding image patches. Note: (i) the the strong grouping within each feature map, (ii) greater invariance at higher layers and (iii) exaggeration of discriminative parts of the image, e.g. eyes and noses of dogs (layer 4, row 1, cols 1). Best viewed in electronic form.

4. Convnet Visualization

Feature Visualization: fig2는 한 번 훈련이 되었을 때 feature visualization을 보여준다. 하지만 주어진 피쳐 맵에서 가장 강력한 하나의 activation을 보여주는 대신, top 9개의 activation을 보여준다. 각각을 분리하여 픽셀 공간으로 투영하는 것은 주어진 피쳐맵을 excite 시키는 다른 구조들을 밝힌다. 따라서 인풋 변형의 불변성을 보여준다. 이러한 시각화에서 대응되는 이미지 패치들을 보여준다. 이러한 것들은 나중의 것들이 패치안에서 차별적은 구조에 홀로 집중하기에 시각화보다 분산이 크다. 각각의 층에서의 투영은 네트워크에서 특징의 계층적 본질을 보여준다. Layer 2는 코너와 다른 edge/color conjunctions에 대응하고 layer 3는 비슷한 재질을 잡아내면서 더 복잡한 불변성을 보여준다.

Feature Evolution during Training: Fig 4는 훈련 도중 가장 강한 activation의 진행과정을 보여준다. 모습에서 급격한 변화가 가장 강한 activation을 초래하는 이미지에서의 변화로부터 발생한다. 낮은 층에서는 약간의 epoch가 필요하다. 하지만 더 깊은 층에서는 40-50 epoch 후에나 수렴한다. 이는 모델이 완전히 수렴하도록 훈련시켜야 하는 필요성을 의미한다.

Feature Invariance: Fig 5는 5개의 이미지가 translate, rotate, scale 된 것을 보여준다. 그리고 이는 변형되지 않은 feature들과 비교하여 모델을 모두 통과시켰을 때의 feature vector의 차이점에 주목한다. 조금의 변형은 모형의 첫 번째 층에서 확실한 효과를 보이지만 더 깊은 층에서는 그렇게 큰 차이가 없다. 망의 output은 translation과 scaling에 불변하지만 rotation에 따라서는 변할 수 있다.

4.1 Architecture Selection

훈련된 모형의 시각화가 운영에 insight를 주는 도중에 최적의 구조를 선택하는 데 도움을 준다. 첫 번째 층과 두 번째 층을 시각화해보면 다양한 문제들이 확실해진다. 첫 번째 층의 필터들은 극도로 높고 낮은 빈번한 정보들의 혼합이다. Mid frequency가 거의 없다. 게다가 두 번째 층의 시각화는 첫 번째 층 convolution에서 사용되는 큰 stride는 4로 인해 aliasing artifacts나 나타난다. 이러한 문제를 해결하기 위해 첫 번째 층의 filter 사이즈를 7x7로 축소하였고 stride를 4로 하였다. 이러한 새로운 구조는 기존의 첫 번째, 두 번째 층의 feature들을 모두 보존한다. 물론 성능도 더 좋았다.

4.2 Occlusion Sensitivity

이미지 분류에 접근할 때 자연스러운 질문은 모형이 정말로 물체의 위치를 잘 포착하는 것인지 주변의 context를 이용하는 것인지에 대한 것이다. Fig 7은 이러한 문제를 체계적으로 이미지의 다른 부분들을 갈색 사각형으로 가리면서 output을 관찰해본다. 이러한 예시들은 모형이 정말로 사진 안에서 물체의 위치를 잘 잡아낸다는 것을 보여준다. 또한, 강한 activation이 이러한 위치를 의미한다고 볼 수 있다. 갈색 사각형이 이러한 위치를 가린다면 feature map에서의 activation이 상당히 감소하였다.

4.3 Correspondence Analysis

깊은 모델들은 존재하는 인식 접근들과 다른 이미지들에서 특정한 물체 부분들이 사이의 대응을 설립하는 확실한 메커니즘이 없다는 점에서 다르다. 하지만, 흥미로운 가능성은 deep model들은 암암리에 그것을 계산할 수도 있다는 것이다. 이를 위해 우리는 정면을 보고 있고 체계적으로 같은 부위를 가리고 있는 5개의 랜덤한 개의 사진을 가져왔다. 그리고 원래 이미지의 피쳐 벡터와 가린 이미지의 피쳐 벡터를 뺀 값을 epsilon으로 할당하겠다.

$$\Delta_l = \sum_{i,j=1,i\neq j}^{5} \mathcal{H}(\text{sign}(\epsilon_i^l), \text{sign}(\epsilon_j^l)),$$

그 다음은 위 식을 계산한다. (Hamming distance) 낮은 값은 가려도 일관성이 있다는 걸 의미한다.

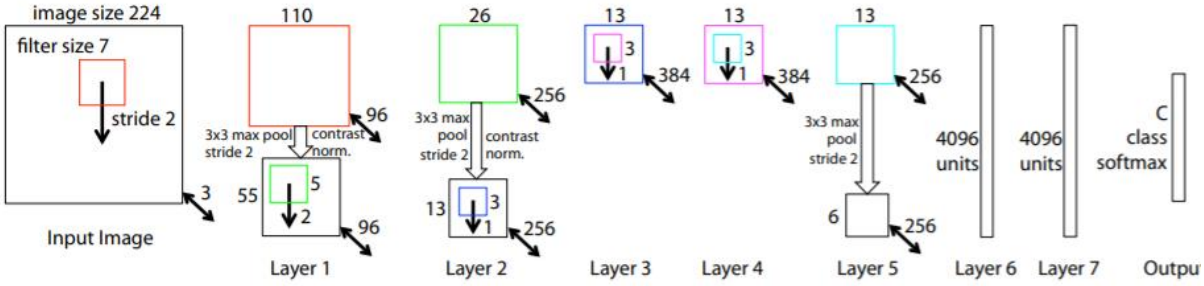


Figure 3. Architecture of our 8 layer convnet model. A 224 by 224 crop of an image (with 3 color planes) is presented as the input. This is convolved with 96 different 1st layer filters (red), each of size 7 by 7, using a stride of 2 in both x and y. The resulting feature maps are then: (i) passed through a rectified linear function (not shown), (ii) pooled (max within 3x3 regions, using stride 2) and (iii) 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.

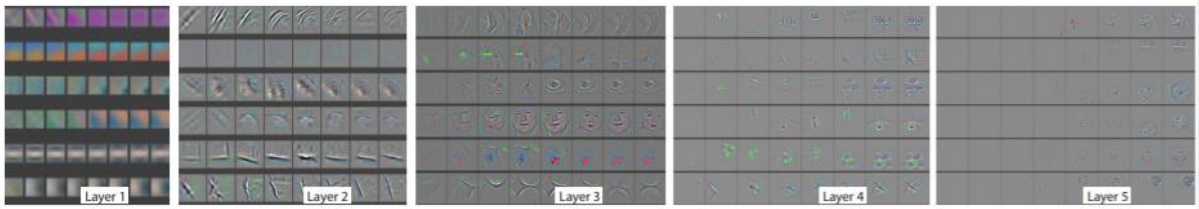


Figure 4. Evolution of a randomly chosen subset of model features through training. Each layer's features are displayed in a different block. Within each block, we show a randomly chosen subset of features at epochs [1,2,5,10,20,30,40,64]. The visualization shows the strongest activation (across all training examples) for a given feature map, projected down to pixel space using our deconvnet approach. Color contrast is artificially enhanced and the figure is best viewed in electronic form.

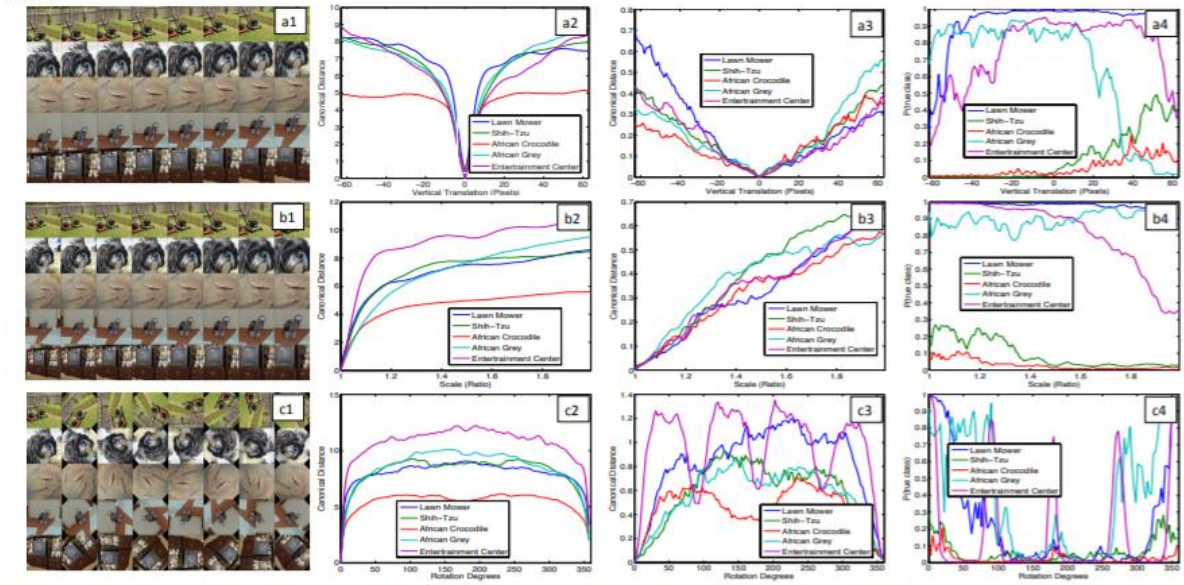


Figure 5. Analysis of vertical translation, scale, and rotation invariance within the model (rows a-c respectively). Col 1: 5 example images undergoing the transformations. Col 2 & 3: Euclidean distance between feature vectors from the original and transformed images in layers 1 and 7 respectively. Col 4: the probability of the true label for each image, as the image is transformed.

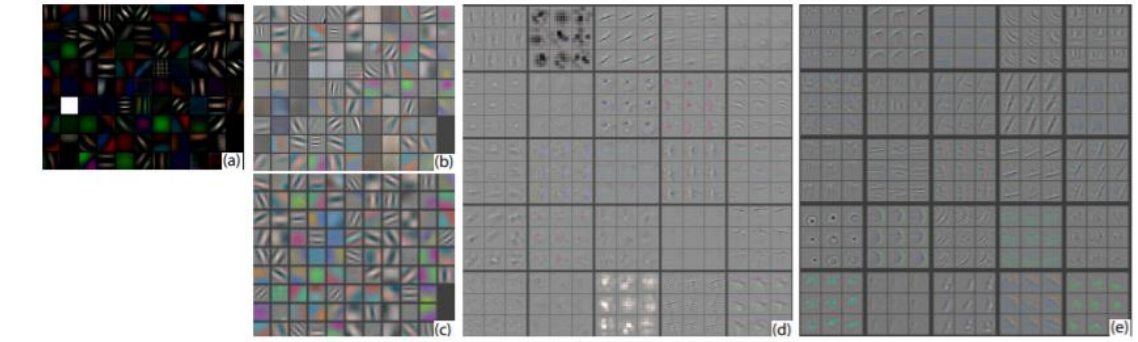


Figure 6. (a): 1st layer features without feature scale clipping. Note that one feature dominates. (b): 1st layer features from (Krizhevsky et al., 2012). (c): Our 1st layer features. The smaller stride (2 vs 4) and filter size (7x7 vs 11x11) results in more distinctive features and fewer "dead" features. (d): Visualizations of 2nd layer features from (Krizhevsky et al., 2012). (e): Visualizations of our 2nd layer features. These are cleaner, with no aliasing artifacts that are visible in (d).

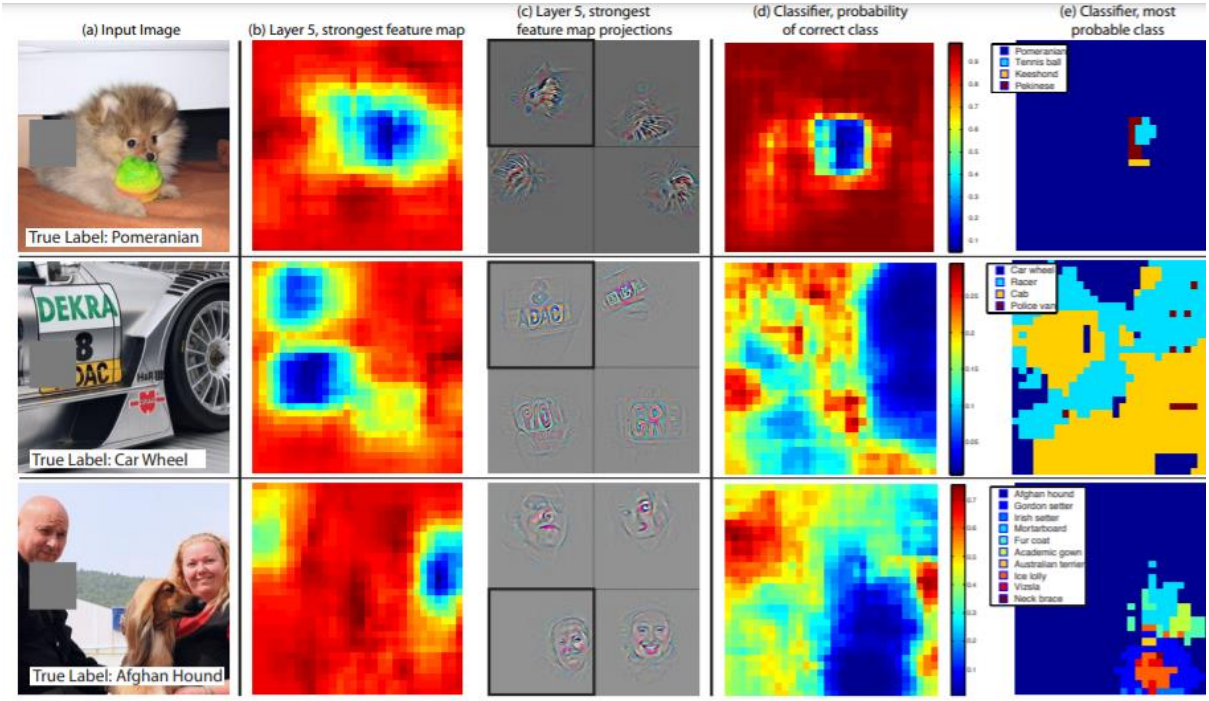


Figure 7. Three test examples where we systematically cover up different portions of the scene with a gray square (1st column) and see how the top (layer 5) feature maps ((b) & (c)) and classifier output ((d) & (e)) changes. (b): for each position of the gray scale, we record the total activation in one layer 5 feature map (the one with the strongest response in the unoccluded image). (c): a visualization of this feature map projected down into the input image (black square), along with visualizations of this map from other images. The first row example shows the strongest feature to be the dog's face. When this is covered-up the activity in the feature map decreases (blue area in (b)). (d): a map of correct class probability, as a function of the position of the gray square. E.g. when the dog's face is obscured, the probability for "pomeranian" drops significantly. (e): the most probable label as a function of occluder position. E.g. in the 1st row, for most locations it is "pomeranian", but if the dog's face is obscured but not the ball, then it predicts "tennis ball". In the 2nd example, text on the car is the strongest feature in layer 5, but the classifier is most sensitive to the wheel. The 3rd example contains multiple objects. The strongest feature in layer 5 picks out the faces, but the classifier is sensitive to the dog (blue region in (d)), since it uses multiple feature maps.

Figure 8. Images used for correspondence experiments. 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

Table 1. Measure of correspondence for different object parts in 5 different dog images. The lower scores for the eyes and nose (compared to random object parts) show the model implicitly establishing some form of correspondence of parts at layer 5 in the model. At layer 7, the scores are more similar, perhaps due to upper layers trying to discriminate between the different breeds of dog.

5. Experiments

5.1 ImageNet 2012

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**

Varying ImageNet Model Sizes

성능을 얻기 위해 모델의 깊이가 중요하다. FC layer의 크기를 변경하는 것은 성능에 큰 차이를 주지 않았다. 하지만 중간 convolution layer의 사이즈를 늘리는 것은 큰 gain을 줬다. (과적합의 위험은 존재한다.)

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

5.2 Feature Generalization

Caltech-101

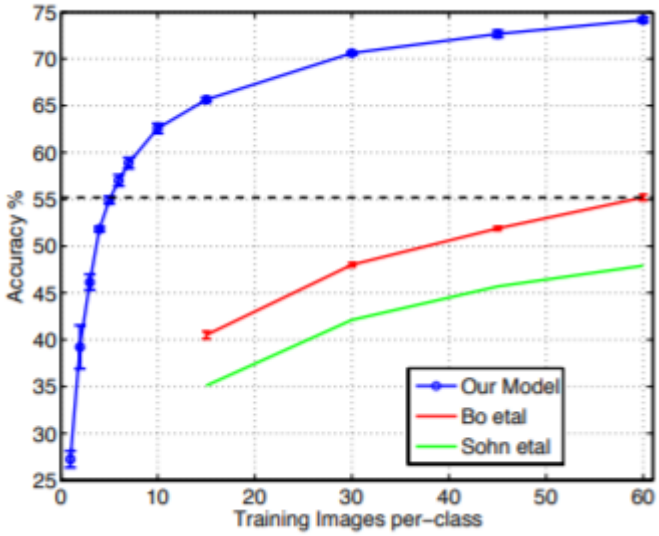
# Train	Acc % 15/class	Acc % 30/class
(Bo et al., 2013)	–	81.4 ± 0.33
(Jianchao et al., 2009)	73.2	84.3
Non-pretrained convnet	22.8 ± 1.5	46.5 ± 1.7
ImageNet-pretrained convnet	$\mathbf{83.8 \pm 0.5}$	$\mathbf{86.5 \pm 0.5}$

Table 4. Caltech-101 classification accuracy for our convnet models, against two leading alternate approaches.

Caltech-256

# 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.



PASCAL 2012

Acc %	[A]	[B]	Ours	Acc %	[A]	[B]	Ours
Airplane	92.0	**97.3**	96.0	Dining tab	63.2	**77.8**	67.7
Bicycle	74.2	**84.2**	77.1	Dog	68.9	83.0	**87.8**
Bird	73.0	80.8	**88.4**	Horse	78.2	**87.5**	86.0
Boat	77.5	85.3	**85.5**	Motorbike	81.0	**90.1**	85.1
Bottle	54.3	**60.8**	55.8	Person	91.6	**95.0**	90.9
Bus	85.2	**89.9**	85.8	Potted pl	55.9	**57.8**	52.2
Car	81.9	**86.8**	78.6	Sheep	69.4	79.2	**83.6**
Cat	76.4	89.3	**91.2**	Sofa	65.4	**73.4**	61.1
Chair	65.2	**75.4**	65.0	Train	86.7	**94.5**	91.8
Cow	63.2	**77.8**	74.4	Tv	77.4	**80.7**	76.1
Mean	74.3	**82.2**	79.0	# won	0	**15**	5

Table 6. 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)).

5.3 Feature Analysis

	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.

6. Discussion

Paper link: https://arxiv.org/pdf/1311.2901.pdf