ImageNet Classification with Deep Convolutional Neural Networks

- description: `AlexNet`
- year: `2012`
- paper
- implementation
- review date: `2023.02.20.`
- tags: #vision , #classification

Abstract

ImageNet의 1000개의 서로 다른 클래스를 가지는 고해상도 이미지 120만 장을 분류하기 위해 크고 깊은 CNN을 학습했고, 이전 sota 모델과 비교해 상당히 발전했다.

뉴럴네트워크는 다섯 개의 convolution layers로 구성되고 6,000만 개의 파라미터와 65만 개의 뉴런을 가진다. 다섯 개의 convolution layer는 max pooling layers와 마지막에 1000-way softmax를 가진 fully-connected layers로 이루어진다.

학습을 빠르게 하기 위해 non-saturating neurons를 사용했고, convolution 연산의 매우 효과적으로 GPU를 구현했다.

fully-connected layers의 overfitting을 줄이기 위해 dropout이라는 regularization 방법을 사용했고 매우 효과가 좋은 것을 입증했다.

1 Introduction

물체 인식을 위해선 머신러닝 방법이 필수인데, 이런 방법의 성능을 높이기 위해선 더 큰 데이터 셋, 더 강력한 모델, overfitting을 막기 위한 더 좋은 기술들이 필요하다.

최근까지 라벨링 된 이미지 데이터 셋은 비교적 작았다.(수만 개) 간단한 인식은 이 정도 데이터 셋으로 풀렸다. 특히 label-preserving transformation을 사용하면 됐다.

- label-preserving transformation: 원본 데이터의 특성을 그대로 보존하면서 augmentation. 예를 들어 6을 뒤집는 방식은 원본과 의미가 달라지므로 preserving이 안 된다.

하지만 현실에서의 객체는 상당히 다양하다. 그렇기 때문에 훨씬 더 큰 데이터 셋을 사용해야 한다. 최근에서야 수백만 개의 라벨링 된 이미지를 수집하는 것이 가능해졌다.

수백만 개의 이미지에서 수천 개의 물체에 대해 배우기 위해선 학습 능력이 큰 모델이 필요하다. 하지만 객체 인식 작업의 엄청난 복잡성은 ImageNet 같은 큰 데이터 셋으로도 커버하기 어렵다.

그렇기 때문에 우리가 가지고 있지 않은 모든 데이터를 커버할 수 있는 많은 사전 지식을 가져야 한다.

모델의 capacity는 깊이와 폭을 변화시켜 제어할 수 있으며, 이미지의 특성에 대해 대부분 정확한 예측을 한다.

하지만 너무 비싸다. 다행히 현재 GPU는 2D convolution 연산에 최적화되어 있어서 큰 CNN의 학습에 충분하고, ImageNet과 같은 데이터 셋에는 심한 overfitting 없이 모델을 학습시킬 수 있는 라벨링 된 예제가 충분히 들어있다.

결국 네트워크의 크기는 사용 가능한 메모리 양과 학습 시간에 의해 제한된다. 더 빠른 GPU와 더 큰 데이터 셋이 사용 가능해지는 것만으로도 결과를 개선할 수 있음을 시사한다.

2 The Dataset

ImageNet은 약 22,000 categories에 속하는 1,500만 개 이상의 라벨이 지정된 고해상도 이미지 데이터 셋이다. 이미지는 웹에서 수집되었고, 사람 라벨러들에 의해 라벨링 되었다.

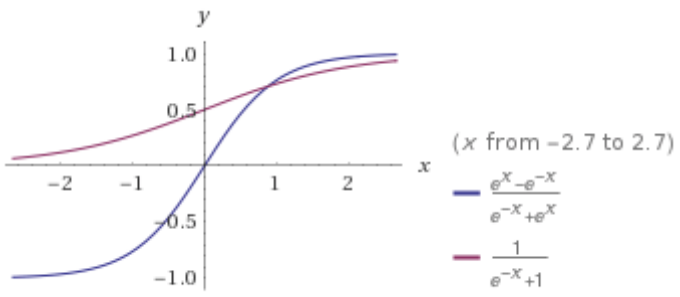
ImageNet은 여러 해상도의 이미지로 구성되어 있지만 일정한 input dimensionality가 필요하기 때문에 256 x 256으로 다운 샘플링했다. 직사각형 이미지가 주어졌을 때 짧은 쪽이 256이 되도록 이미지 크기를 조정한 뒤, 이미지 중앙에서 256 x 256 패치를 잘라냈다.

각 픽셀에 대해 학습 데이터 셋에 대한 평균을 뺀 것을 제외하고는 다른 사전 처리는 하지 않았다.

3 The Architecture

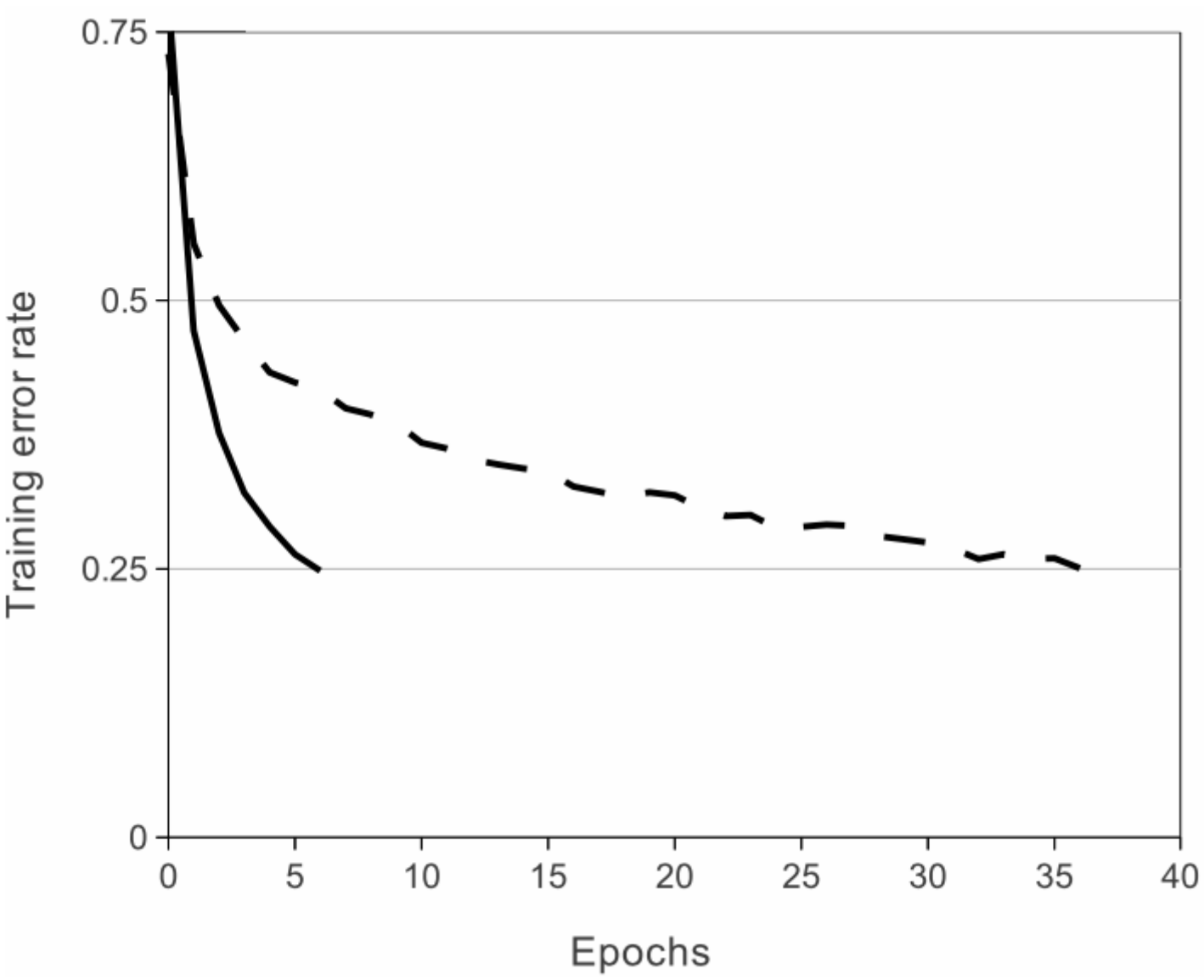
중요도 순으로 아키텍처의 기능 중 일부를 설명한다.

3.1 ReLU Nonlinearity



이미지 출처: https://sebastianraschka.com/faq/docs/tanh-sigmoid-relationship.html

뉴런의 출력을 모델링 하는 표준 방법은 위 이미지에 나온 $tanh$(파란 그래프) 또는 $sigmoid$(빨간 그래프)이다. 경사하강법을 사용한 학습 시간 측면에서 이러한 saturating nonlinearity는 non-saturating nonlinearity $f(x) = max(0, x)$(ReLU)보다 훨씬 느리다. 아래 이미지는 CIFAR-10 데이터 셋에서 25% training error을 도달하는 데 필요한 epoch 수를 보여준다.(solid line: ReLU 사용, dashed line: $tanh$ 사용)



이미지 출처: paper

3.2 Training on Multiple GPUs

두 개의 GPU를 사용했다. 각 GPU에 특정한 layer만 할당함으로써 각각에 절반의 뉴런을 할당했다.

3.3 Local Response Normalization

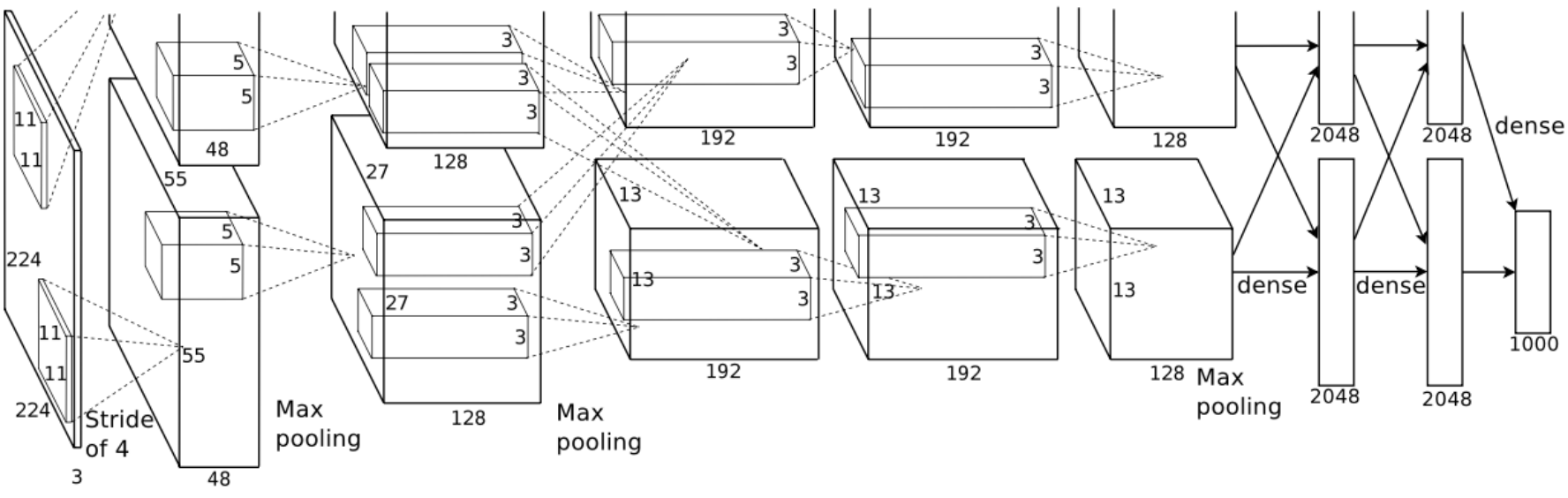
ReLU는 saturating을 방지하기 위해 input normalization을 할 필요가 없다. 일부 학습 데이터에서 양의 input이 들어오면 그 뉴런에서는 학습이 일어날 것이다. 하지만 local normalization이 generalization에 도움이 된다는 것을 발견했다.

- 양수인 입력일 경우 ReLU는 값을 그대로 사용한다. 이때 값이 큰 양수 값이 있다면 주변에 영향을 줄 수 있다. 이를 막기 위해 측면 억제(lateral inhibition)를 사용한다.
  - Inter-channel LRN: 서로 다른 채널의 같은 위치에 있는 픽셀들의 정규화
  - Intra-channel LRN: 같은 채널의 이웃한 픽셀들의 정규화

3.4 Overlapping Pooling

커널이 움직이는 폭이 커널 사이즈보다 작은(stride = 1, kernel size = 2 x 2) 중복이 허용되는 overlapping pooling을 진행했다. overlapping pooling이 사용된 모델은 overfitting 되기 어렵다는 것을 관찰했다. 또한 이는 max pooling이다.

3.5 Overall Architecture



이미지 출처: paper

5개의 convolution layers + 3개의 fully-connected layers = 총 8개의 learned layers로 구성된다.

fully-connected layer의 출력은 1000-way softmax로 들어가 1000개의 클래스 라벨에 대한 분포를 생성한다.

이 네트워크는 multinomial logistic regression의 objective를 최대화하는데, 이는 예측 분포에서 학습 데이터에서 올바른 라벨의 로그 확률의 평균을 최대화하는 것과 같다.(AlexNet의 loss function = Cross Entropy)

두 번째, 네 번째, 다섯 번째 convolution layer의 kernel은 동일 GPU에 있는 이전 layer의 kernel map에만 연결된다.

세 번째 convolution layer의 kernel은 두 번째 layer의 모든 kernel map에 연결된다.(이를 통해 더 복잡한 패턴을 인식하고 이미지를 더 정확하게 분류할 수 있다.)

response normalization은 첫 번째, 두 번째 convolution layer에 있다.

max pooling은 첫 번째, 두 번째, 다섯 번째 layer에 있다.

ReLU는 모든 convolution layer와 fully-connected layer에 적용된다.

4 Reducing Overfitting

이 신경망 아키텍처는 6,000만 개의 파라미터를 가지고 있다. overfitting과 싸우는 두 가지 주요 방법을 설명한다.

4.1 Data Augmentation

가장 쉽고 일반적인 방법은 label-preserving transformation이다. 논문에서는 두 가지 다른 형태의 data augmentation을 사용하는데, 두 가지 모두 매우 적은 계산으로 가능하다. GPU에서 이전 이미지 배치에 대해 학습하는 동안 CPU에서 변환 이미지가 생성된다. 따라서 이러한 확대는 자유롭다.

data augmentation 첫 번째로 horizontal reflection image를 생성하는 것이다. 256 x 256 이미지에서 무작위 224 x 224 패치(+ 패치의 horizontal reflection image)를 추출하여 학습에 사용한다.(학습 데이터 32x32x2배 증가)

test 시에는 네 모서리 패치와 중앙 패치 crop(총 다섯 종류), horizontal reflection(두 종류)을 하여 한 이미지 당 10개의 패치를 추출하고 이들 예측값의 평균을 결과로 낸다.

data augmentation 두 번째로 훈련 이미지에서 RGB 채널의 값을 변경시키는 것이다.

4.2 Dropout

많은 다른 모델의 예측을 결합하는 것은 test error를 줄이는 매우 좋은 방법이지만 대규모 신경망에서는 비용이 너무 많이 든다. 그러나 훈련 비용이 약 2배 밖에 들지 않는 매우 효율적인 모델 조합이 있다. dropout은 50% 확률로 hidden 뉴런의 출력을 0으로 설정한다. 이렇게 drop된 뉴런은 forward pass와 back propagation에 참여하지 않는다. 이 기술은 뉴런이 다른 특정 뉴런에 의존할 수 없기 때문에 뉴런의 복잡한 co-adaptation(공동 적응: 둘 이상이 한 쌍 또는 그룹으로 적응하는 과정)을 감소시킨다.

test 시에는 모든 뉴런을 사용한다. 하지만 뉴런들의 output 들에 0.5를 곱한다. 이는 많은 dropout 네트워크에 의해 생성된 예측 분포의 기하학적 평균을 취하는 것에 대한 합리적인 근사치이다.(0.5를 곱하는 이유: overfitting을 줄임으로써 새로운 데이터에 대한 일반화 성능을 높이기 위함)

처음 두 개의 fully-connected layer에서 dropout을 사용한다. dropout이 없는 이 네트워크는 상당한 overfitting을 보인다. dropout은 수렴에 필요한 iteration 횟수를 대략 두 배 늘린다.

5 Details of learning

이 모델에는 SGD가 사용되었다.

- batch size = 128
- momentum = 0.9
- weight decay = 0.0005
  작은 값의 weight decay는 모델 학습에 중요하다는 것을 발견했다. 다시 말해, 여기서 weight decay는 단순한 정규화가 아니라 모델의 training error를 감소시킨다.

논문에서는 표준 편차가 0.01인 zero-mean Gaussian distribution에서 각 layer의 weight을 초기화했다. 그리고 두 번째, 네 번째, 다섯 번째 convolution layer와 fully-connected hidden layer에서 bias를 1로 초기화했다. 이 초기화는 ReLU에 양의 input을 제공함으로써 학습의 초기 단계를 가속화한다. 나머지 layer의 bias는 0으로 초기화했다.

논문에서는 학습을 하는 동안 수동으로 조정한 learning rate을 사용했고, 이는 모든 layer에서 동일했다. 논문에서 사용한 heuristic(불충분한 시간이나 정보로 인하여 합리적인 판단을 할 수 없거나, 체계적이면서 합리적인 판단이 굳이 필요하지 않은 상황에서 사람들이 빠르게 사용할 수 있게 보다 용이하게 구성된 간편 추론의 방법)은 validation error rate가 개선되지 않을 때 learning rate을 10으로 나누는 것이었다. learning rate은 0.01로 초기화하고 종료 전 세 번 감소했다.

6 Results

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

Table 1: Comparison of results on ILSVRC-2010 test set. In *italics* are best results achieved by others.

이미지 출처: paper

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%**

Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets. In *italics* are best results achieved by others. Models with an asterisk* were "pre-trained" to classify the entire ImageNet 2011 Fall release. See Section 6 for details.

이미지 출처: paper

6.1 Qualitative Evaluations

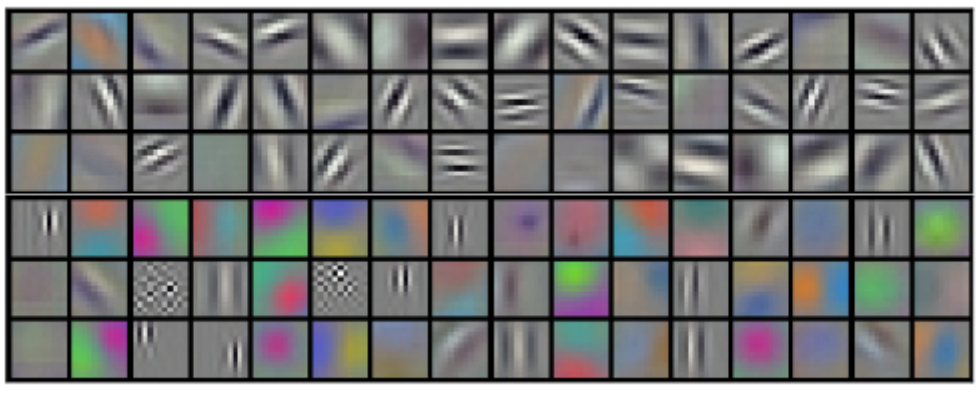


Figure 3: 96 convolutional kernels of size $11 \times 11 \times 3$ learned by the first convolutional layer on the $224 \times 224 \times 3$ input images. The top 48 kernels were learned on GPU 1 while the bottom 48 kernels were learned on GPU 2. See Section 6.1 for details.

이미지 출처: paper

위 이미지를 보면 GPU 1의 kernel들(위 48 kernels)과 GPU 2의 kernel들(아래 48 kernels)이 다르다. (각 GPU는 자체 파라미터 집합을 가지고 있기 때문에 디테일이 다름)

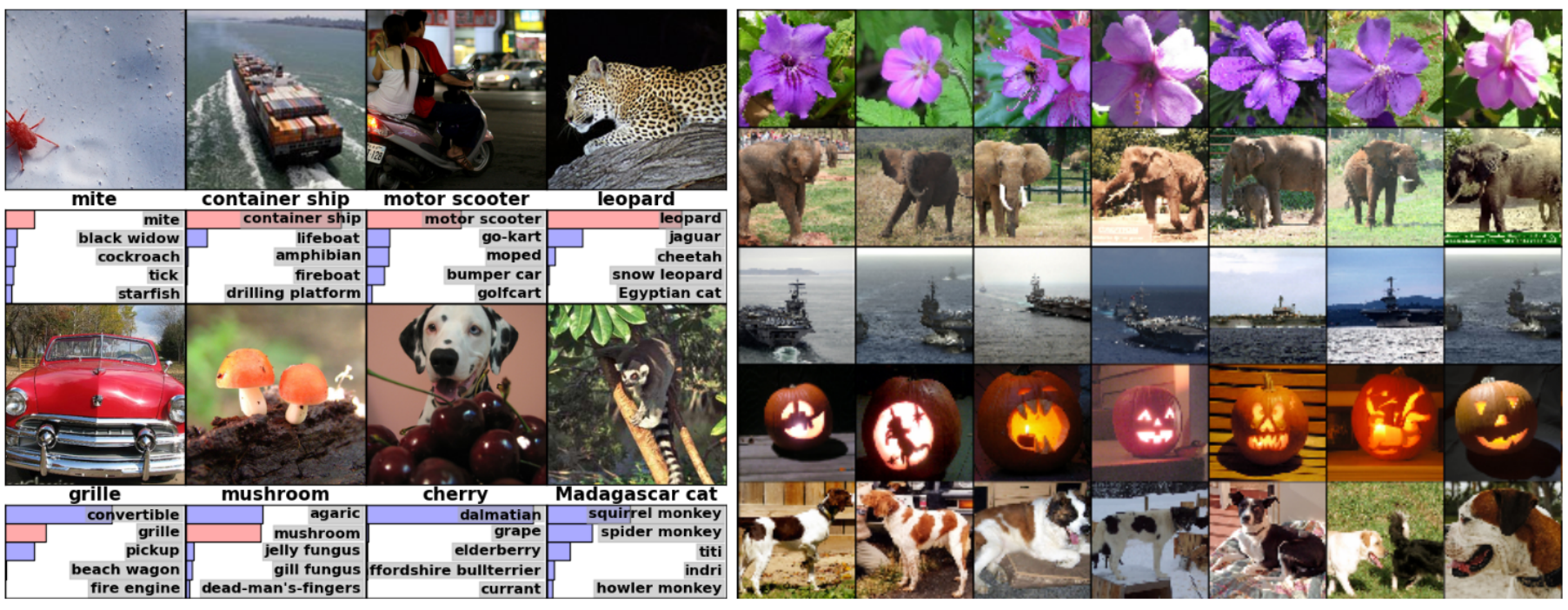


Figure 4: **(Left)** Eight ILSVRC-2010 test images and the five labels considered most probable by our model. The correct label is written under each image, and the probability assigned to the correct label is also shown with a red bar (if it happens to be in the top 5). **(Right)** Five ILSVRC-2010 test images in the first column. The remaining columns show the six training images that produce feature vectors in the last hidden layer with the smallest Euclidean distance from the feature vector for the test image.

이미지 출처: paper

위 이미지의 왼쪽 패널에서는 8개의 test 이미지에 대한 상위 5개의 예측의 정성적 평가를 보여준다.

오른쪽 패널에서는 첫 번째 열에 test 이미지가 있고, 나머지 열의 이미지 6개는 test 이미지의 feature vector에서 가장 작은 Euclidean distance를 가지는 마지막 4096차원의 hidden layer에서 feature vector를 생성하는 학습 이미지를 보여준다. test 이미지와 train 이미지가 매우 비슷하다는 것을 보여준다. 픽셀 수준에서는 비슷하지 않다. 개와 코끼리를 다양한 포즈로 찾았다.

두 개의 4096차원의 벡터의 Euclidean distance를 계산하는 것은 비효율적이지만 autoencoder(인코더를 통해 입력을 신호로 변환한 다음 다시 디코더를 통해 레이블 따위를 만들어내는 비지도 학습 기법)를 이용해 벡터를 압축하여 학습한다면 효율적으로 만들 수 있다.

7 Discussion

논문에서는 deep convolution neural network의 효과를 보여주었다. layer를 제거하면 성능이 저하되기 때문에 효과적인 결과를 얻기 위해서는 깊이가 중요하다고 말한다.