MobileNets: 모바일 비전 애플리케이션을 위한 효율적인 컨볼루션 신경망

앤드류 G. 하워드 주 보 첸 칼레니첸코웨이준 왕 토비아스 웨이앤드
마르코 안드레토 하트비그 아담

Google Inc.
{howarda,menglong,bochen,dkalenichenko,weijunw,weyand,anm,hadam}@google.com

초록

Basler는 모바일 및 임베디드 비전 애플리케이션을 위한 효율적인 모델 클래스인 MobileNets를 소개합니다. 모바일넷은 깊이별 분리형 컨볼루션을 사용해 경량 심층 신경망을 구축하는 간소화된 아키텍처를 기반으로 합니다. 지연 시간과 정확도 사이에서 효율적으로 균형을 맞추는 두 가지 간단한 글로벌 하이퍼 파라미터를 도입했습니다. 이러한 하이퍼 파라미터를 통해 모델 빌더는 문제의 제약 조건에 따라 애플리케이션에 적합한 크기의 모델을 선택할 수 있습니다. 리소스 및 정확도 트레이드오프에 대한 광범위한 실험을 통해 이미지넷 분류에서 널리 사용되는 다른 모델과 비교하여 강력한 성능을 보여줍니다. 그런 다음 객체 감지, 세분화된 분류, 얼굴 속성, 대규모 지리적 위치 파악 등 다양한 애플리케이션과 사용 사례에서 모바일넷의 효율성을 입증합니다.

1. 소개

컨볼루션 신경망은 AlexNet [19]이 Ima- geNet Challenge: ILSVRC 2012 [24]에서 우승하여 심층 컨볼루션 신경망을 대중화한 이후 컴퓨터 비전 분야에서 보편화되었습니다. 더 높은 정확도를 달성하기 위해 더 깊고 복잡한 네트워크를 만드는 것이 일반적인 추세였습니다 [27, 31, 29, 8]. 하지만 정확도 향상을 위한 이러한 발전이 반드시 네트워크의 크기와 속도 측면에서 더 효율적인 네트워크를 만드는 것은 아닙니다. 로봇 공학, 자율 주행 자동차, 증강 현실과 같은 많은 실제 애플리케이션에서 인식 작업은 연산 능력이 제한된 플랫폼에서 적시에 수행되어야 합니다.

이 백서에서는 모바일 및 임베디드 비전 애플리케이션의 설계 요구 사항에 쉽게 맞출 수 있는 매우 작고 지연 시간이 짧은 모델을 구축하기 위한 효율적인 네트워크 아키텍처와 두 가지 하이퍼파라미터 세트에 대해 설명합니다. 섹션 2에서는 소규모 네트워크 구축에 대한 이전 작업을 검토합니다.

모델에 대해 설명합니다. 섹션 3에서는 더 작고 효율적인 모바일넷을 정의하기 위한 모바일넷 아키텍처와 두 가지 하이퍼파라미터 폭 승수 및 해상도 승수에 대해 설명합니다. 섹션 4에서는 이미지넷에 대한 실험과 다양한 애플리케이션 및 사용 사례에 대해 설명합니다. 섹션 5는 요약과 결론으로 마무리합니다.

2. 이전 작업

최근 문헌에서 작고 효율적인 신경망을 구축하는 것에 대한 관심이 높아지고 있습니다(예: [16, 34, 12, 36, 22]). 다양한 접근 방식은 일반적으로 사전 훈련된 네트워크를 압축하거나 작은 네트워크를 직접 훈련하는 것으로 분류할 수 있습니다. 이 백서에서는 모델 개발자가 애플리케이션의 리소스 제한(지연 시간, 크기)에 맞는 소규모 네트워크를 구체적으로 선택할 수 있는 네트워크 아키텍처 클래스를 제안합니다. 모바일넷은 주로 지연 시간 최적화에 초점을 맞추지만 소규모 네트워크도 생성할 수 있습니다. 소규모 네트워크에 관한 많은 논문은 크기에만 초점을 맞추고 속도를 고려하지 않습니다.

모바일넷은 주로 [26]에서 처음 도입된 깊이 분리형 컨볼루션으로 구축되며, 이후 Inception 모델 [13]에서 처음 몇 개의 레이어에서 계산을 줄이기 위해 사용되었습니다. 평탄화된 네트워크[16]는 완전히 인수분해된 컨볼루션으로 네트워크를 구축하고 극도로 인수분해된 네트워크의 잠재력을 보여주었습니다. 이 논문과는 별개로, 팩터링된 네트워크[34]는 유사한 팩터화된 컨볼루션과 위상 연결의 사용을 소개합니다. 그 후, Xception 네트워크[3]는 심층적으로 분리 가능한 필터를 확장하여 Inception V3 네트워크보다 뛰어난 성능을 발휘하는 방법을 시연했습니다. 또 다른 소규모 네트워크는 병목현상 접근법을 사용하여 매우 작은 네트워크를 설계하는 Squeezenet [12]입니다. 다른 축소 계산 네트워크에는 구조화된 변환 네트워크[28]와 딥 프라이드 컨브넷[37]이 있습니다.

작은 네트워크를 얻기 위한 다른 접근 방식은 미리 훈련된 네트워크를 축소, 인수분해 또는 압축하는 것입니다. 곱 양자화[36], 해싱에 기반한 압축

그림 1. 모바일넷 모델은 다양한 인식 작업에 적용하여 디바이스 인텔리전스를 효율적으로 활용할 수 있습니다.

[2], 프루닝, 벡터 양자화 및 허프만 코딩 [5]가 문헌에서 제안되었습니다. 또한 사전 학습된 네트워크의 속도를 높이기 위해 다양한 인수분해가 제안되었습니다[14, 20]. 작은 네트워크를 훈련하는 또 다른 방법은 증류[9]로, 더 큰 네트워크를 사용해 더 작은 네트워크를 훈련하는 것입니다. 이 방법은 우리의 접근 방식을 보완하며 섹션 4의 일부 사용 사례에서 다룹니다. 또 다른 새로운 접근법은 저비트 네트워크입니다[4, 22, 11].

3. 모바일넷 아키텍처

이 섹션에서는 먼저 Mo- bileNet이 깊이별로 분리 가능한 필터인 핵심 레이어에 대해 설명합니다.
그런 다음 MobileNet 네트워크 구조를 설명하고 두 가지 모델 축소 하이퍼 파라미터 폭 배율과 해상도 배율에 대한 설명으로 마무리합니다.

3.1. 깊이 분리형 컨볼루션

모바일넷 모델은 표준 컨볼루션을 깊이 방향 컨볼루션과 점 방향 컨볼루션이라고 하는 1×1 컨볼루션으로 분해하는 인수분해 컨볼루션의 한 형태인 깊이 방향 분리 가능 컨볼루션을 기반으로 합니다. 모바일넷의 경우 깊이 방향 컨볼루션은 각 입력 채널에 단일 필터를 적용합니다. 그런 다음 점 단위 컨볼루션은 1 × 1 컨볼루션을 적용하여 깊이 단

$D_F \times M$ 피처 맵 **F**는 $D_F \times D_F \times N$ 피처 맵 **G**를 생성합니다. 여기서 D_F 는 정사각형 입력 피처 맵의 공간 폭과 높이입니다.[1]는 입력 채널 수(입력 깊이), D_G 는 정사각형 출력 피처 맵의 공간 폭과 높이, N 은 출력 채널 수(출력 깊이)입니다.

표준 컨볼루션 레이어는 $D_K \times D_K \times M \times N$ 크기의 컨볼루션 커널 K로 파라미터화되며, 여기서 D_K 는 정사각형으로 가정한 커널의 공간 차원이고 M은 입력 채널 수, N은 앞서 정의한 대로 출력 채널 수입니다.

보폭 1과 패딩을 가정한 표준 컨볼루션의 출력 피처 맵은 다음과 같이 계산됩니다:

$$\mathbf{Gk}_{n}^{,l,} = \sum_{\substack{\mathbf{K} \\ i,j,m}} {}_{N}^{I,J,M,} \text{ - Fk+i-1,l+j-1,}m \qquad (1)$$

위 컨볼루션의 출력을 결합합니다. 표준 컨볼루션은 한 번에 입력을 필터링하고 새로운 출력 세트로 결합합니다. 깊이 분리형 컨볼루션은 이를 필터링을 위한 별도의 레이어와 결합을 위한 별도의 레이어, 두 개의 레이어로 나눕니다. 이러한 인수분해는 계산과 모델 크기를 크게 줄이는 효과가 있습니다. 그림 2는 표준 컨볼루션 2(a)가 깊이 방향 컨볼루션 2(b)와 1 × 1 점 방향 컨볼루션 2(c)로 인수분해되는 과정을 보여줍니다.

표준 컨볼루션 계층은 $D_F \times$를 입력으로 받습니다.

표준 컨볼루션의 계산 비용은 다음과 같습니다:

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F \quad (2)$$

여기서 계산 비용은 입력 채널 수 M, 출력 채널 수 N, 커널 크기 $D_k \times D_k$, 피처 맵 크기 $D_F \times D_F$ 에 따라 곱셈으로 달라집니다. MobileNet 모델은 이러한 각 용어와 그 상호 작용을 다룹니다. 먼저 깊이별 세분화 컨볼루션을 사용하여 출력 채널 수와 커널 크기 간의 상호 작용을 분리합니다.

표준 컨볼루션 연산은 새로운 표현을 생성하기 위해 컨볼루션 커널을 기반으로 피처를 필터링하고 피처를 결합하는 효과가 있습니다. 필터링 및 결합 단계는 깊이별로라고 하는 인수분해 컨볼루션을 사용하여 두 단계로 나눌 수 있습니다.

[1] 출력 피처 맵의 공간 크기가 입력과 같고 두 피처 맵이 모두 정사각형이라고 가정합니다. 모델 축소 결과는 임의의 크기와 종횡비를 가진 특징 맵으로 일반화됩니다.

분리 가능한 컨볼루션을 통해 컴퓨팅 비용을 크게 절감할 수 있습니다.

깊이별로 분리 가능한 컨볼루션은 깊이별 컨볼루션과 점별 컨볼루션의 두 가지 레이어로 구성됩니다. 깊이별 컨볼루션을 사용하여 각 입력 채널(입력 깊이)당 단일 필터를 적용합니다. 그런 다음 단순한 1×1 컨볼루션인 점 단위 컨볼루션을 사용하여 깊이별 레이어의 출력의 선형 조합을 생성합니다. 모바일넷은 두 레이어 모두에 배치노름과 ReLU 비선형성을 모두 사용합니다.

입력 채널당 하나의 필터(입력 깊이)를 사용한 깊이별 컨볼루션은 다음과 같이 쓸 수 있습니다:

$$\hat{G}_{k,l,m} = \sum_{i,j} \hat{K}_{i,j,m} \cdot F_{k+i-1,l+j-1,m} \quad (3)$$

여기서 $\hat{K}$는 깊이 방향 컨볼루션 커널 크기입니다.
$D_K \times D_K \times M$ 여기서 $\hat{K}$의 m_{th} 필터는 다음에 적용됩니다.
에 m_{th} 채널을 추가하여 필터링된 출력 피처 맵 $\hat{G}$ 의 m_{th} 채널을 생성합니다.

깊이 컨볼루션의 계산 비용은 다음과 같습니다:

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F \quad (4)$$

깊이 방향 컨볼루션은 다음과 비교했을 때 매우 효율적입니다. 표준 컨볼루션입니다. 그러나 입력 채널만 필터링할 뿐, 채널들을 결합하여 새로운 특징을 생성하지는 않습니다. 따라서 이러한 새로운 특징을 생성하려면 1×1 컨볼루션을 통해 깊이별 컨볼루션 출력의 선형 조합을 계산하는 추가 레이어가 필요합니다.

깊이 방향 컨볼루션과 1 × 1(점 방향) 컨볼루션의 조합을 깊이 방향 분리 가능 컨볼루션이라고 하며, 이는 원래 [26]에서 소개된 바 있습니다.

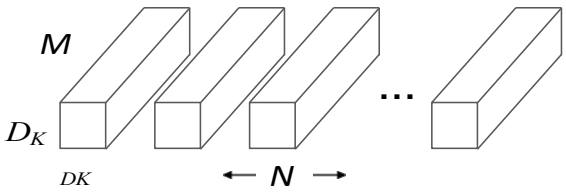
깊이별로 분리 가능한 컨볼루션 비용입니다:

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F \quad (5)$$

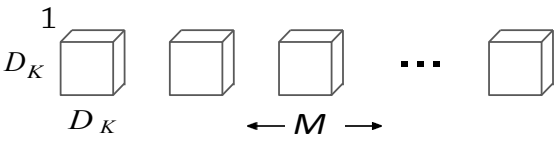
는 깊이 방향과 1 × 1 포인트 방향의 비율을 합한 값입니다.

컨볼루션을 필터링과 결합의 두 단계 프로세스로 표현하면 다음과 같은 계산을 줄일 수 있습니다:

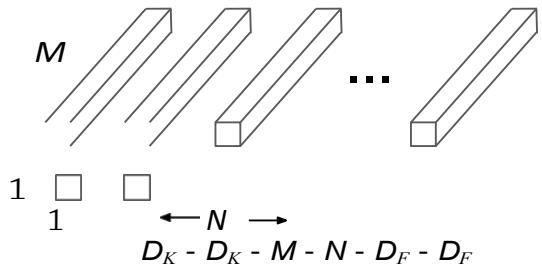
$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{}$$



(a) 표준 컨볼루션 필터



(b) 깊이별 컨볼루션 필터



(c) 1 × *1* 컨볼루션 필터는 깊이 분리형 컨볼루션의 컨텍스트에서 점 단위 컨볼루션이라고 합니다.

그림 2. (a)의 표준 컨볼루션 필터는 (b)의 깊이 방향 컨볼루션과 (c)의 점 방향 컨볼루션이라는 두 개의 레이어로 대체되어 깊이 방향으로 분리 가능한 필터를 구축합니다.

3.2. 네트워크 구조 및 교육

MobileNet 구조는 전체 컨볼루션인 첫 번째 레이어를 제외하고는 이전 섹션에서 언급한 것처럼 깊이별로 분리

$$= \frac{1}{N} + \frac{1}{D2_K}$$

모바일넷은 3 × 3 깊이로 분리 가능한 컨볼루션을 사용하는데, 이 컨볼루션은 섹션 4에서 볼 수 있듯이 정확도는 약간 떨어지지만 일반 컨볼루션보다 8~9배 적은 계산을 사용합니다.

16, 31]에서와 같이 공간 차원에서의 추가 인수분해는 심층 컨볼루션에 사용되는 계산이 거의 없기 때문에 추가 계산을 많이 절약할 수 없습니다.

가능한 컨볼루션을 기반으로 구축됩니다. 이렇게 간단한 용어로 네트워크를 정의하면 네트워크 토폴로지를 쉽게 탐색하여 좋은 네트워크를 찾을 수 있습니다. MobileNet 아키텍처는 표 1에 정의되어 있습니다. 비선형성이 없고 분류를 위해 소프트맥스 레이어에 공급되는 최종 완전 연결 레이어를 제외한 모든 레이어에는 배치노름[13]과 ReLU 비선형성이 뒤따릅니다. 그림 3은 일반 컨볼루션, 배치노름, ReLU 비선형성을 적용한 레이어와 인수분해 레이어를 깊이별로 대조한 것입니다.

컨볼루션, 1 × 1 포인트 컨볼루션 및 배치-
노름과 ReLU를 각 컨볼루션 레이어 뒤에 추가합니다. 하향 샘플링은 첫 번째 레이어뿐만 아니라 깊이별 컨볼루션에서도 보폭 컨볼루션으로 처리됩니다. 최종 평균 풀링은 완전히 연결된 레이어 이전에 공간 해상도를 1로 줄입니다. 깊이 방향 컨볼루션과 점 방향 컨볼루션을 별도의 레이어로 계산하면 MobileNet에는 28개의 레이어가 있습니다.

단순히 소수의 멀티 애드만으로 네트워크를 정의하는 것만으로는 충분하지 않습니다. 이러한 작업을 효율적으로 구현할 수 있는지 확인하는 것도 중요합니다. For

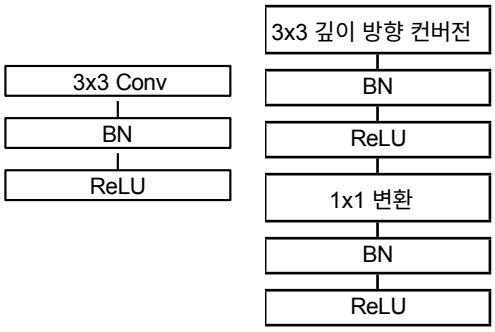


그림 3. 왼쪽: 배치노름과 ReLU가 포함된 표준 컨볼루션 레이어. 오른쪽: 깊이별 및 점별 레이어와 배치 노름 및 ReLU를 사용한 깊이별 분리형 컨볼루션.

인스턴스의 구조화되지 않은 희소 행렬 연산은 매우 높은 수준의 희소성이 아니면 일반적으로 조밀한 행렬 연산보다 빠르지 않습니다. 우리의 모델 구조는 거의 모든 연산을 고밀도 1 × 1 컨볼루션으로 처리합니다. 이는 고도로 최적화된 일반 행렬 곱하기(GEMM) 함수로 구현할 수 있습니다. 컨볼루션은 GEMM으로 구현되는 경우가 많지만, 이를 GEMM에 매핑하기 위해서는 im2col이라는 메모리에서 초기 순서를 다시 지정해야 합니다. 예를 들어, 이 접근 방식은 널리 사용되는 Caffe 패키지에서 사용됩니다[15]. 1 × 1 컨볼루션은 메모리에서 이러한 재정렬이 필요하지 않으며, 가장 최적화된 수치 선형 대수 알고리즘 중 하나인 GEMM으로 직접 구현할 수 있습니다. MobileNet은 표 2 에서 볼 수 있듯이 전체 계산 시간의 95%를 1×1 컨볼루션에 소비하며, 이 컨볼루션에는 75%의 파라미터가 사용됩니다. 거의 모든 추가 파라미터는 완전히 연결된 레이어에 있습니다.

모바일넷 모델은 텐서플로우[1]로 학습되었습니다. Inception V3 [31]와 유사한 비동기 경사 하강을 사용하는 RMSprop [33]을 사용합니다. 그러나 대규모 모델을 훈련할 때와는 달리, 작은 모델은 과적합 문제가 적기 때문에 정규화 및 데이터 증강 기법을 덜 사용합니다. 모바일넷을 훈련할 때는 사이드 헤드나 라벨 평활화를 사용하지 않으며, 대규모 Inception 훈련에 사용되는 작은 크롭의 크기를 제한하여 왜곡된 이미지의 양을 추가로 줄입니다[31]. 또한, 깊이 필터에는 파라미터가 매우 적기 때문에 가중치 감쇠(l2 정규화)를 거의 또는 전혀 적용하지 않는 것이 중요하다는 사실을 발견했습니다. 다음 섹션의 ImageNet 벤치마크에서는 모델의 크기에 관계없이 모든 모델을 동일한 훈련 파라미터로 훈련했습니다.

3.3. 너비 배율: 더 얇은 모델

기본 MobileNet 아키텍처는 이미 작고 지연 시간이 짧지만, 특정 사용 사례나 애플리케이션에 따라 더 작고 빠른 모델이 필요할 수 있습니다. 이러한 더 작고 계산 비용이 적게 드는 모델을 구축하기 위해 폭 승수라는 매우 간단한 매개변수 α를 도입했습니다. 폭 승수 α의 역할은 각 레이어에서 네트워크를 균일하게 얇게 만드는 것입니다. 주어진 레이어에 대해

표 1 . 모바일넷 본체 아키텍처

유형 / 보폭	필터 모양	입력 크기
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 DW	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 DW	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 DW	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 DW	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 DW	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 DW	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5× Conv dw / s1 Conv / s1	3 × 3 × 512 dw 1 × 1 × 512 × 512	14 × 14 × 512 14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
평균 풀 / s1	풀 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
소프트맥스/S1	분류기	1 × 1 × 1000

표 2 . 레이어 유형별 리소스

유형	멀티 애드	매개변수
Conv 1 × 1	94.86%	74.59%
Conv DW 3 × 3	3.06%	1.06%
Conv 3 × 3	1.19%	0.02%
완벽한 연결	0.18%	24.33%

와 폭 승수 α*를* 곱하면 입력 채널 수 M*은* αM*이 되고* 출력 채널 수 N*은* αN*이* 됩니다.

너비 승수 α*가* 있는 깊이 분리 가능한 컨볼루션의 계산 비용은 다음과 같습니다:

$$D_K \cdot D_K \cdot \alpha M \cdot D_F \cdot D_F + \alpha M \cdot \alpha N \cdot D_F \cdot D_F \quad (6)$$

여기서 $\alpha \in (0, 1]$, 일반적인 설정은 1, 0.75, 0.5 및 0.25. $\alpha = 1$은 기준 모바일넷, $\alpha < 1$은 축소된 모바일넷입니다. 폭 승수는 계산 비용과 매개변수 수를 약 α*만큼* 감소시키는 효과가 있습니다[2] . 폭 승수는 모든 모델 구조에 적용하여 합리적인 정확도, 지연 시간 및 크기 절충을 통해 새로운 더 작은 모델을 정의할 수 있습니다. 이 는 처음부터 학습해야 하는 새로운 축소 구조를 정의하는 데 사용됩니다.

3.4. 해상도 승수: 대표성 감소

신경망의 계산 비용을 줄이기 위한 두 번째 하이퍼 파라미터는 해상도 승수 ρ*입니다*.

표 3. 표준 컨볼루션 수정 시 리소스 사용량. 각 행은 이전 행 위에 추가되는 누적 효과라는 점에 유의하세요. 이 예는 $D_K = 3$, $M = 512$, $N = 512$, $D_F = 14$인 내부 MobileNet 레이어에 대한 것입니다.

레이어/수정	백만 멀티 애드	백만 매개변수
컨볼루션	462	2.36
깊이 분리형 컨버전	52.3	0.27
$\alpha = 0.75$	29.6	0.15
$\rho = 0.714$	15.1	0.15

를 입력 이미지에 곱하면 모든 레이어의 내부 표현이 동일한 승수만큼 감소합니다. 실제로는 입력 해상도를 설정하여 ρ를 암시적으로 설정합니다.

이제 네트워크의 핵심 레이어에 대한 계산 비용을 폭 승수 α와 해상도 승수 ρ를 사용하여 깊이별로 분리 가능한 컨볼루션으로 표현할 수 있습니다:

$$D_K \cdot D_K \cdot \alpha M \cdot \rho D_F \cdot \rho D_F + \alpha M \cdot \alpha N \cdot \rho D_F \cdot \rho D_F \quad (7)$$

여기서 $\rho \in (0, 1]$은 일반적으로 네트워크의 입력 해상도가 224, 192, 160 또는 128이 되도록 암시적으로 설정됩니다. $\rho = 1$은 기준 모바일 네트워크이고 $\rho < 1$은 계산이 줄어든 모바일 네트워크입니다. 해상도 승수는 계산 비용을 ρ만큼 줄이는 효과가 있습니다[2] .

예를 들어, 모빌넷의 일반적인 레이어를 살펴보고 깊이별로 분리 가능한 컨볼루션, 너비 승수, 해상도 승수가 어떻게 비용과 파라미터를 줄이는지 살펴볼 수 있습니다. 표 3은 아키텍처 축소 방법이 레이어에 순차적으로 적용될 때 레이어의 계산 및 매개변수 수를 보여줍니다. 첫 번째 행은 입력 피처 맵 크기가 $14 \times 14 \times 512$이고 케르넬 K가 $3 \times 3 \times 512 \times 512$인 전체 컨볼루션 레이어에 대한 멀티-애드 및 파라미터를 보여줍니다. 다음 섹션에서는 리소스와 정확도 간의 트레이드오프에 대해 자세히 살펴보겠습니다.

4. 실험

이 섹션에서는 먼저 심층 컨볼루션의 효과와 레이어 수가 아닌 네트워크의 폭을 다시 줄여 축소하는 방법을 살펴봅니다. 그런 다음 폭 승수와 해상도 승수라는 두 가지 하이퍼파라미터에 따라 네트워크 작업을 줄일 때의 장단점을 보여주고 여러 인기 모델과 결과를 비교합니다. 그런 다음 다양한 애플리케이션에 적용된 모바일넷을 조사합니다.

4.1. 모델 선택

먼저 전체 컨볼루션으로 구축된 모델과 비교하여 깊이별 분리 컨볼루션을 사용한 MobileNet의 결과를 보여줍니다. 표 4에서 전체 컨볼루션과 비교하여 깊이별 분리 컨볼루션을 사용하면 다음과 같은 결과만 감소하는 것을 볼 수 있습니다.

표 4 . 깊이 분리형과 풀 컨볼루션 모바일넷 비교

모델	이미지넷 정확도	백만 다중 추가	백만 파라미터
Conv MobileNet	71.7%	4866	29.3
모바일넷	70.6%	569	4.2

표 5 . 좁은 모바일 네트워크와 얕은 모바일 네트워크

모델	이미지넷 정확도	백만 다중 추가	백만 파라미터
0.75 모바일넷	68.4%	325	2.6
얕은 모바일 네트워크	65.3%	307	2.9

표 6 . 모바일넷 폭 승수

너비 배율	이미지넷	백만 멀티 애드	백만 파라미터
1.0 MobileNet-224	70.6%	569	4.2
0.75 MobileNet-224	68.4%	325	2.6
0.5 MobileNet-224	63.7%	149	1.3
0.25 MobileNet-224	50.6%	41	0.5

표 7 . 모바일넷 해상도

해상도	이미지넷 정확도	백만 다중 추가	백만 파라미터
1.0 모바일넷-224	70.6%	569	4.2
1.0 MobileNet-192	69.1%	418	4.2
1.0 MobileNet-160	67.2%	290	4.2
1.0 MobileNet-128	64.4%	186	4.2

정확도를 1% 향상시킴으로써 멀티 애드 및 파라미터를 크게 절약할 수 있었습니다.

다음은 폭 승수를 적용한 더 얇은 모델과 더 적은 레이어를 사용한 더 얕은 모델을 비교한 결과를 보여줍니다. 모바일넷을 더 얇게 만들기 위해 표 1에서 피처 크기가 14 × 14 × 512인 분리 가능한 필터의 5개의 레이어를 제거했습니다. 표 5는 비슷한 계산과 매개변수 수에서 모바일넷을 더 얇게 만드는 것이 더 얕게 만드는 것보다 3% 더 낫다는 것을 보여줍니다.

4.2. 모델 축소 하이퍼파라미터

표 6은 폭 승수 α를 사용하여 MobileNet 아키텍처를 축소할 때의 정확도, 계산 및 크기 절충안을 보여줍니다. 아키텍처를 너무 작게 만들면 $\alpha = 0.25$가 될 때까지 정확도가 완만하게 떨어집니다.

표 7은 입력 해상도를 낮춘 모빌넷을 훈련하여 다양한 해상도 배율에 대한 정확도, 계산 및 크기 절충안을 보여줍니다. 정확도는 해상도에 따라 완만하게 떨어집니다.

그림 4는 폭 승수 $\alpha \in \{1, 0.75, 0.5, 0.25\}$와 해상도 {224, 192, 160, 128}의 교차 곱으로 만든 16개 모델에 대한 ImageNet 정확도와 계산 간의 트레이드 오프를 보여줍니다. 결과는 $\alpha = 0.25$에서 모델이 매우 작아질 때 점프가 있는 로그 선형입니다.

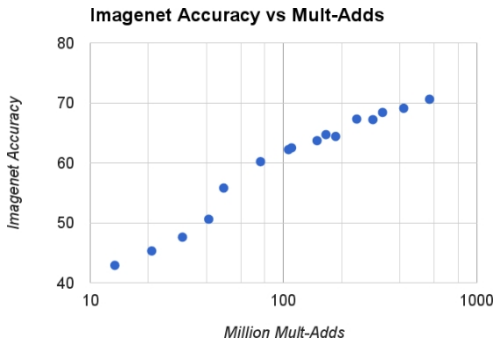


그림 4. 이 그림은 ImageNet 벤치마크에서 계산(멀티-애드)과 정확도 간의 절충점을 보여줍니다. 정확도와 계산 간의 로그 선형 의존성에 주목하세요.

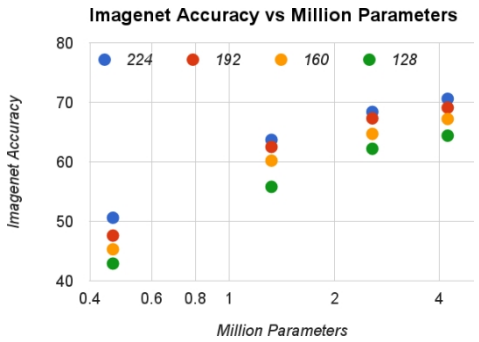


그림 5. 이 그림은 다음 두 가지의 절충점을 보여줍니다.
매개변수 및 ImageNet 벤치마크의 정확도. 색상
입력 해상도를 인코딩합니다. 매개변수 수는 변하지 않습니다.
입력 해상도를 기준으로 합니다.

그림 5는 너비 승수 $\alpha \in$의 교차 곱으로 만든 16개 모델에 대한 이미지넷 성능과 매개변수 수 간의 절충점을 보여줍니다.
{1, 0.75, 0.5, 0.25} 및 해상도 {224, 192, 160, 128}입니다
.

표 8은 전체 모바일넷과 기존 구글넷[30] 및 VGG16[27]을 비교한 것입니다. MobileNet은 VGG16과 거의 비슷하면서도 크기가 32배 더 작고 컴퓨팅 집약도가 27배 더 낮습니다. 더 작고 계산량이 2.5배 이상 적으면서도 구글넷보다 더 정확합니다.

표 9는 폭을 0.5로 줄이고 해상도를 160 × 160으로 줄

표 8. 모바일넷과 인기 모델 비교

모델	이미지넷 정확도	백만 멀티 애드	백만 매개변수
1.0 MobileNet-224	70.6%	569	4.2
구글넷	69.8%	1550	6.8
VGG 16	71.5%	15300	138

표 9. 인기 모델과 소형 모바일넷 비교

모델	이미지넷 정확도	백만 멀티 애드	백만 매개변수
0.50 MobileNet-160	60.2%	76	1.32
스퀴즈넷	57.5%	1700	1.25
AlexNet	57.2%	720	60

표 10. 스탠포드 개를 위한 모바일넷

모델	Top-1 정확도	백만 멀티 애드	백만 매개변수
인셉션 V3 [18]	84%	5000	23.2
1.0 MobileNet-224	83.3%	569	3.3
0.75 MobileNet-224	81.9%	325	1.9
1.0 MobileNet-192	81.9%	418	3.3
0.75 MobileNet-192	80.5%	239	1.9

표 11. MobileNet 아키텍처를 사용한 PlaNet의 성능. 백분율은 Im2GPS 테스트 데이터 세트에서 기준점으로부터 특정 거리 내에 위치가 파악된 비율을 나타냅니다. 원래 PlaNet 모델의 수치는 아키텍처와 훈련 데이터 세트가 개선된 업데이트 버전을 기준으로 합니다.

Scale	Im2GPS [7]	PlaNet [35]	PlaNet 모바일넷
대륙(2500km)	51.9%	77.6%	79.3%
국가(750km)	35.4%	64.0%	60.3%
지역(200km)	32.1%	51.1%	45.2%
시내 (25km)	21.9%	31.7%	31.7%
거리 (1km)	2.5%	11.0%	11.4%

인 축소 MobileNet을 비교한 것입니다. 축소된 모바일넷은 알렉스넷[19]보다 45배 더 작고 9.4배 $\mathit{더}$ 적은 컴퓨팅을 사용하면서도 알렉스넷보다 4% 더 우수합니다. 또한 크기는 거의 같고 계산량은 22배 적은 Squeezenet[12]보다 4% 더 우수합니다.

4.3. 세분화된 인식

스탠퍼드 개 데이터 세트 [17]에 대해 세분화된 인식을 위해 MobileNet을 훈련합니다. 우리는 [18]의 접근 방식을 확장하여 웹에서 [18]보다 훨씬 더 크지만 노이즈가 많은 훈련 집합을 수집합니다. 노이즈가 많은 웹 데이터를 사용하여 세분화된 개 인식 모델을 사전 훈련한 다음, 스탠포드 독스 훈련 세트에서 모델을 미세 조정합니다. 스탠포드 독스 테스트 세트의 결과는 표 10에 나와 있습니다. MobileNet은 컴퓨팅과 크기를 크게 줄이면서 [18]의 최첨단 결과를 거의 달성할 수 있습니다.

4.4. 대규모 지오로컬라이제이션

PlaNet [35]은 사진이 지구상에서 어디에서 촬영되었는지를 결정하는 작업을 분류 문제로 제시합니다. 이 접근 방식은 지구를 목표 클래스 역할을 하는 지리적 셀 그리드로 나누고 컨볼루션 신경망을 훈련시킵니다.

를 사용하여 수백만 장의 사진에서 위치를 파악할 수 있습니다. PlaNet은 다양한 종류의 사진을 성공적으로 로컬라이즈하고 동일한 작업을 처리하는 Im2GPS[6, 7]를 능가하는 성능을 보여줬습니다.

동일한 데이터에 대해 MobileNet 아키텍처를 사용하여 PlaNet을 재학습합니다. 기본 V3 아키텍처[31]를 기반으로 하는 전체 PlaNet 모델에는 5,200만 개의 매개변수가 있고
57억 4천만 개의 멀티애드. 모바일넷 모델의 파라미터는 1,300만 개에 불과하며, 본문에 300만 개, 최종 레이어에 1,000만 개, 멀티 애드에 58만 개가 있습니다. 탭. 11에서 볼 수 있듯이, 모바일넷 버전은 훨씬 더 컴팩트함에도 불구하고 플라넷에 비해 성능이 약간만 떨어집니다. 또한 여전히 Im2GPS를 큰 차이로 능가합니다.

4.5. 얼굴 속성

MobileNet의 또 다른 사용 사례는 알려지지 않았거나 난해한 훈련 절차가 있는 대규모 시스템을 압축하는 것입니다. 얼굴 속성 분류 작업에서 우리는 MobileNet과 딥 네트워크의 지식 전달 기법인 증류[9] 사이의 시너지 효과를 입증했습니다. 우리는 7,500만 개의 파라미터와 1억 6,000만 개의 멀티 애드를 가진 대규모 얼굴 속성 분류기를 축소하고자 합니다. 이 분류기는 YFCC100M [32]과 유사한 다중 속성 데이터 세트에서 훈련됩니다.

MobileNet 아키텍처를 사용하여 얼굴 속성 분류기를 증류합니다. 증류[9]는 분류기가 더 큰 모델의 출력을 에뮬레이션하도록 훈련하는 방식으로 작동합니다.[2] 따라서 라벨이 없는 대규모(그리고 잠재적으로 무한한) 데이터 세트에서 학습할 수 있습니다. 증류 훈련의 확장성과 MobileNet의 간결한 매개변수화가 결합된 최종 시스템은 정규화(예: 가중치 붕괴 및 조기 중지)가 필요하지 않을 뿐만 아니라 향상된 성능을 보여줍니다. Tab. 12에서 MobileNet 기반 분류기가 공격적인 모델 축소에 다시 침묵하는 것을 알 수 있습니다. 이 분류기는 멀티 애드를 1%만 사용하면서 인하우스와 유사한 속성 평균 정밀도(평균 AP)를 달성합니다.

4.6. 물체 감지

MobileNet은 최신 객체 감지 시스템에서 효과적인 기본 네트워크로도 배포할 수 있습니다. 2016년 COCO 챌린지 [10]에서 우승한 최근 작업을 기반으로 COCO 데이터에서 객체 감지를 위해 훈련된 MobileNet의 결과를 보고합니다. 표 13에서 MobileNet은 Faster-RCNN [23] 및 SSD [21] 프레임워크 모두에서 VGG 및 Inception V2 [13]와 비교됩니다. 실험에서 SSD는 300개의 입력 해상도(SSD 300)로 평가하고, Faster-RCNN은 300개와 600개의 입력 해상도 (Faster- RCNN 300, Faster-RCNN 600) 모두로 비교했습니다. Faster-RCNN 모델은 이미지당 300개의 RPN 제안 상자를 평가합니다. 모델은 8k 미니벌 이미지를 제외한 COCO 트레인+밸로 훈련됩니다.

[2] 에뮬레이션 품질은 모든 속성에 대한 속성별 교차 엔트로피의 평균을 구하여 측정합니다.

표 12. MobileNet 아키텍처를 사용한 얼굴 속성 분류. 각 행은 서로 다른 하이퍼파라미터 설정(폭 승수 α 및 이미지 해상도)에 해당합니다.

폭 배율 / 해상도	AP	평균 백만 멀티 애드	백만 매개변수
1.0 MobileNet-224	88.7%	568	3.2
0.5 MobileNet-224	88.1%	149	0.8
0.25 MobileNet-224	87.2%	45	0.2
1.0 MobileNet-128	88.1%	185	3.2
0.5 MobileNet-128	87.7%	48	0.8
0.25 MobileNet-128	86.4%	15	0.2
기준선	86.9%	1600	7.5

표 13. 서로 다른 프레임워크와 네트워크 아키텍처를 사용한 COCO 오브젝트 탐지 결과 비교. mAP는 COCO 기본 챌린지 지표(IoU=0.50:0.05:0.95에서 AP)로 보고됩니다.

프레임워크 해상도	모델	mAP	억 멀티 애드	백만 매개변수
SSD 300	deeplab-VGG	21.1%	34.9	33.1
	인셉션 V2	22.0%	3.8	13.7
	모바일넷	19.3%	1.2	6.8
Faster-RCNN 300	VGG	22.9%	64.3	138.5
	인셉션 V2	15.4%	118.2	13.3
	모바일넷	16.4%	25.2	6.1
Faster-RCNN 600	VGG	25.7%	149.6	138.5
	인셉션 V2	21.9%	129.6	13.3
	모바일넷	19.8%	30.5	6.1

그림 6. MobileNet SSD를 사용한 오브젝트 탐지 결과 예시.

로 설정하고 미니벌에서 평가했습니다. 두 프레임워크 모두에서 모바일넷은 계산 복잡성과 모델 크기의 일부만으로 다른 네트워크와 비슷한 결과를 달성합니다.

4.7. 얼굴 임베딩

FaceNet 모델은 최첨단 얼굴 인식 모델입니다[25]. 이 모델은 삼중 손실에 기반하여 얼굴 임베딩을 구축합니다. 모바일 FaceNet 모델을 구축하기 위해 증류법을 사용하여 출력의 제곱 차이를 최소화하여 훈련합니다.

표 14 . 모바일넷 디스틸 페이스넷에서 주도

Model1e-4	정확도 다중 추가	백만	백만 파 라 미터
페이스넷 [25]	83%	1600	7.5
1.0 MobileNet-160	79.4%	286	4.9
1.0 MobileNet-128	78.3%	185	5.5
0.75 MobileNet-128	75.2%	166	3.4
0.75 MobileNet-128	72.5%	108	3.8

의 훈련 데이터에 대한 FaceNet과 MobileNet의 비율입니다. 아주 작은 모바일넷 모델에 대한 결과는 표 14에서 확인할 수 있습니다.

5. 결론

저희는 심층적으로 분리 가능한 컨볼루션을 기반으로 하는 새로운 모델 아키텍처인 Mo- bileNets을 제안했습니다. 효율적인 모델로 이어지는 몇 가지 중요한 설계 결정을 조사했습니다. 그런 다음 크기와 지연 시간을 줄이기 위해 적정 수준의 정확도를 유지하면서 폭 배율과 해상도 배율을 사용해 더 작고 빠른 모바일넷을 구축하는 방법을 시연했습니다. 그런 다음 다양한 모바일넷을 여러 인기 모델과 비교하여 최고의 크기, 속도, 정확도 특성을 보여주었습니다. 다양한 작업에 적용했을 때 MobileNet의 효과를 입증하는 것으로 결론을 내렸습니다. 모바일넷의 도입과 탐색을 돕기 위한 다음 단계로 텐서플로우의 모듈을 출시할 계획입니다.

참조

[1] M. 아바디, A. 아가르왈, P. 바햄, E. 브레브도, Z. 첸, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin 외. Tensorflow: 이기종 시스템에 대한 대규모 기계 학습, 2015. *소프트웨어는 tensorflow. org에서 제공*, 1, 2015. 4

[2] W. 첸, J. T. 윌슨, S. 타이리, K. Q. 와인버거, 및 Y. Chen. 해싱을 이용한 신경망 압축 트릭. *CoRR, abs/1504.04788*, 2015. 2

[3] F. 콜레트. Xception: 깊이별 세파라블 컨볼루션을 이용한 딥 러닝. *arXiv preprint arXiv:1610.02357v2*, 2016. 1

[4] M. Courbariaux, J.-P. David, and Y. Bengio. 저정밀 곱셈을 이용한 심층 신경망 훈련. *arXiv 사전 인쇄본 arXiv:1412.7024*, 2014. 2

[5] S. 한, H. 마오, 및 W. J. 달리. 심층 압축: 가지치기, 훈련된 양자화 및 허프만 코딩을 사용한 심층 신경망 압축. *CoRR, abs/1510.00149*, 2, 2015. 2

[6] J. Hays와 A. Efros. IM2GPS: 단일 이미지에서 지리적 형성 추정. *IEEE 국제 컴퓨터 비전 및 패턴 컨퍼런스 논문집 인식*, 2008. 7

[7] J. Hays and A. Efros. 대규모 이미지 지오로컬라이제이션. J. Choi 및 G. Friedland, 편집자, *비디오 및 이미지의 다중 모드 위치 추정*. Springer, 2014. 6, 7

[8] K. He, X. Zhang, S. Ren, and J. Sun. 이미지 인식을 위한 심층 잔여 학습. *arXiv preprint arXiv:1512.03385*, 2015. 1

[9] G. 힌튼, O. 빈얄스, 및 J. 딘. 신경망에서 지식의 증류. *arXiv 사전 인쇄물 arXiv:1503.02531*, 2015. 2, 7

[10] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama 등. 최신 컨볼루션 객체 검출기의 속도/정확도 트레이드오프. *arXiv preprint arXiv:1611.10012*, 2016. 7

[11] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, 및 Y. 벤지오. 정량화된 신경망: 저정밀 가중치 및 활성화로 신경망 훈련. *arXiv 사전 인쇄본 arXiv:1609.07061*, 2016. 2

[12] F. N. 이안돌라, M. W. 모스케비츠, K. 아슈라프, S. 한, W. J. 달리, 그리고 K. 커처. Squeezenet: 50배 더 적은 매개변수와¡ 1mb 모델 크기로 Alexnet 수준의 정확도. *arXiv 사전 인쇄본 arXiv:1602.07360*, 2016. 1, 6

[13] S. 이오페와 C. 세게디. 배치 정규화: 내부 공변량 이동을 줄임으로써 심층 네트워크 학습 가속화. *arXiv preprint arXiv:1502.03167*, 2015. 1, 3, 7

[14] M. Jaderberg, A. Vedaldi, and A. Zisserman. 낮은 순위 확장으로 컨볼루션 신경망의 속도 향상. *arXiv preprint arXiv:1405.3866*, 2014. 2

[15] Y. 지아, E. 셸하머, J. 도나휴, S. 카라예프, J. 롱, R. 기르식, S. 구아다라마, T. 대럴. Caffe: 빠른 피처 임베딩을 위한 컨볼루션 아키텍처. *arXiv 사전 인쇄 arXiv:1408.5093*, 2014. 4

[16] J. Jin, A. Dundar, and E. Culurciello. 피드포워드 가속을 위한 평탄화된 컨볼루션 신경망. *arXiv 프리프린트 arXiv:1412.5474*, 2014. 1, 3

[17] A. 코슬라, N. 자야데바프라카시, B. 야오, 및 L. 페이페이. 세분화된 이미지 분류를 위한 새로운 데이터 세트. 세분화된 *시각적 분류에 관한 첫 번째 워크샵, IEEE 컴퓨터 비전 및 패턴 인식 컨퍼런스*, 콜로라도 스프링스, 콜로라도, 2011년 6월. 6

[18] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, 및 L. Fei-Fei. 세분화된 인식을 위한 노이즈 데이터의 불합리한 효율성. *arXiv 사전 인쇄본 arXiv:1511.06789*, 2015. 6

[19] A. 크리제프스키, I. 수츠케버, 및 G. E. 힌튼. 심층 컨볼루션 신경망을 이용한 이미지넷 분류. *신경 정보 처리 시스템의 발전*, 페이지 1097-1105, 2012. 1, 6

[20] V. 레베데프, Y. 가닌, M. 라쿠바, I. 오셀레데츠, 및 V. 렘피츠키. 미세 조정된 cp-분해를 이용한 컨볼루션 신경망 작업 속도 향상. *arXiv preprint arXiv:1412.6553*, 2014. 2

[21] W. 리우, D. 안젤로프, D. 에르한, C. 세게디, 및 S. 리드. *arXiv 사전 인쇄 arXiv:1512.02325*, 2015. 7

[22] M. 라스테가리, V. 오르도네즈, J. 레드몬, 및 A. 파르하디. Xnor- net: 이진 컨볼루션 신경망을 이용한 이미지넷 분류 ral 네트워크. *arXiv 사전 인쇄물 arXiv:1603.05279*, 2016. 1, 2

[23] S. Ren, K. He, R. Girshick, and J. Sun. 더 빠른 r-cnn: 영역 제안 네트워크를 통한 실시간 객체 감지를 향하여. *신경 정보 처리 시스템의 발전*, 91-99페이지, 2015. 7

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein 외. Imagenet 대규모 시각 인식 챌린지. *국제 컴퓨터 비전 저널*, 115(3):211-252, 2015. 1

[25] F. 슈로프, D. 칼레니첸코, 및 J. 필빈. 패싯: 얼굴 인식 및 클러스터링을 위한 단일 임베딩. *IEEE 컴퓨터 비전 컨퍼런스 프로시딩 및 패턴 인식*, 815-823페이지, 2015. 8

[26] L. Sifre. *이미지 분류를 위한 리지드 모션 산란*. 박사 학위 논문, 박사 학위 논문, 2014. 1, 3

[27] K. Simonyan and A. Zisserman. 대규모 이미지 인식을 위한 매우 심층적인 컨볼루션 네트워크. *arXiv preprint arXiv:1409.1556*, 2014. 1, 6

[28] V. Sindhwani, T. Sainath, 및 S. Kumar. 소형 풋프린트 딥러닝을 위한 구조화된 트랜스폼. *신경 정보 처리 시스템의 발전*, 3088-3096 페이지, 2015. 1

[29] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet 및 잔여 연결이 학습에 미치는 영향. *arXiv 사전 인쇄물 arXiv:1602.07261*, 2016. 1

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. 안젤로프, D. 에르한, V. 반호케, 및 A. 라비노비치. 컨볼루션으로 더 깊이 들어가기. *IEEE 컴퓨터 비전 및 패턴 인식 컨퍼런스 논문집*, , 1-9페이지, 2015. 6

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. 컴퓨터 비전을 위한 시작 아키텍처 재고. *arXiv 사전 인쇄본 arXiv:1512.00567*, 2015. 1, 3, 4, 7

[32] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. 폴란드, D. Borth, 및 L.-J. Li. Yfcc100m: 멀티미디어 연구의 새로운 데이터. *ACM 커뮤니케이션*, 59(2):64-73, 2016. 7

[33] T. 티엘레만과 G. 힌튼. 강의 6.5-rmsprop: 기울기를 최근 크기의 평균으로 나눕니다. *코스 세라: 기계 학습을 위한 신경망*, 4(2), 2012. 4

[34] M. Wang, B. Liu, and H. Foroosh. 인수분해 컨볼루션 신경망. *arXiv preprint arXiv:1608.04337*, 2016. 1

[35] T. Weyand, I. Kostrikov, and J. Philbin. PlaNet - 컨볼루션 신경망을 이용한 사진 위치 파악. In *European 컴퓨터 비전 컨퍼런스(ECCV)*, 2016. 6, 7

[36] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng. 모바일 장치를 위한 정량화된 컨볼루션 신경망. *arXiv 사전 인쇄본 arXiv:1512.06473*, 2015. 1

[37] Z. 양, M. 모줄스키, M. 데닐, N. 드 프레이타스, A. 스몰라,

L. 송, 그리고 Z. 왕. 튀긴 콘넷. *IEEE 국제 컴퓨터 비전 컨퍼런스 논문*집, 1476-1483페이지, 2015. 1