**Squeeze-and-Excitation Networks**

Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu

**Abstract**

The central building block of convolutional neural networks (CNNs) is the convolution operator, which enables networks to construct informative features by fusing both spatial and channel-wise information within local receptive fields at each layer. A broad range of prior research has investigated the spatial component of this relationship, seeking to strengthen the representational power of a CNN by enhancing the quality of spatial encodings throughout its feature hierarchy. In this work, we focus instead on the channel relationship and propose a novel architectural unit, which we term the “Squeeze-and-Excitation” (SE) block, that adaptively recalibrates channel-wise feature responses by explicitly modelling interdependencies between channels. We show that these blocks can be stacked together to form SENet architectures that generalise extremely effectively across different datasets. We further demonstrate that SE blocks bring significant improvements in performance for existing state-of-the-art CNNs at slight additional computational cost. Squeeze-and-Excitation Networks formed the foundation of our ILSVRC 2017 classification submission which won first place and reduced the top-5 error to 2.251%, surpassing the winning entry of 2016 by a relative improvement of ∼25%. Models and code are available at <https://github.com/hujie-frank/SENet>.

컨볼루션 뉴럴 네트워크(CNN)의 중심 구성 요소는 컨볼루션 연산자로, 각 레이어의 로컬 수용 필드 내에서 공간 정보와 채널 정보 모두를 융합함으로써 네트워크가 유익한 기능을 구성할 수 있도록 합니다. 광범위한 사전 연구는 특징 계층 전반에 걸쳐 공간 인코딩의 품질을 향상시킴으로써 CNN의 표현력을 강화하기 위해 이 관계의 공간 구성요소를 조사했다. 본 연구에서는 대신 채널 관계에 초점을 맞추고 채널 간의 상호의존성을 명시적으로 모델링하여 채널별 특징 응답을 적응적으로 재보정하는 새로운 아키텍처 유닛을 제안한다. 이러한 블록을 함께 쌓아 서로 다른 데이터 세트에 걸쳐 매우 효과적으로 일반화하는 SENet 아키텍처를 형성할 수 있음을 보여 줍니다. 또한 SE 블록이 약간의 추가 계산 비용으로 기존 최첨단 CNN의 성능을 크게 향상시킨다는 것을 입증한다. 스퀴즈 앤 들뜸 네트워크는 ILSVRC 2017 분류 제출의 토대를 형성하여 상위 5위 오차를 2.251%로 줄여 2016년도 입상작을 약 25% 개선하였습니다. 모델 및 코드는 https://github.com/hujie-frank/SENet에서 구할 수 있습니다.

Convolutional neural networks (CNNs) have proven to be useful models for tackling a wide range of visual tasks. At each convolutional layer in the network, a collection of filters expresses neighbourhood spatial connectivity patterns along input channels—fusing spatial and channel-wise information together within local receptive fields. By interleaving a series of convolutional layers with non-linear activation functions and downsampling operators, CNNs are able to produce image representations that capture hierarchical patterns and attain global theoretical receptive fields. A central theme of computer vision research is the search for more powerful representations that capture only those properties of an image that are most salient for a given task, enabling improved performance. As a widely-used family of models for vision tasks, the development of new neural network architecture designs now represents a key frontier in this search. Recent research has shown that the representations produced by CNNs can be strengthened by integrating learning mechanisms into the network that help capture spatial correlations between features. One such approach, popularised by the Inception family of architectures, incorporates multi-scale processes into network modules to achieve improved performance. Further work has sought to better model spatial dependencies and incorporate spatial attention into the structure of the network.

컨볼루션 뉴럴 네트워크(CNN)는 광범위한 시각적 작업을 처리하는 데 유용한 모델임이 입증되었다. 네트워크의 각 컨볼루션 층에서 필터 집합은 입력 채널을 따라 인접 공간 연결 패턴을 표현하며, 로컬 수용 필드 내에서 공간 및 채널별 정보를 함께 결합합니다. 일련의 컨볼루션 레이어를 비선형 활성화 함수와 다운샘플링 연산자와 함께 인터리브함으로써 CNN은 계층 패턴을 포착하고 글로벌 이론 수용 필드를 달성하는 이미지 표현을 생성할 수 있다. 컴퓨터 비전 연구의 중심 테마는 특정 작업에서 가장 중요한 이미지의 속성만을 포착하여 성능 향상을 가능하게 하는 보다 강력한 표현을 찾는 것입니다. 비전 작업에 널리 사용되는 모델 제품군으로서, 새로운 뉴럴 네트워크 아키텍처 설계의 개발은 이제 이 검색의 핵심 프런티어를 나타냅니다. 최근 연구에 따르면 CNN에 의해 생산된 표현은 특징 사이의 공간적 상관관계를 포착하는 데 도움이 되는 학습 메커니즘을 네트워크에 통합함으로써 강화될 수 있다. Inception 제품군에 의해 널리 보급된 이러한 접근법 중 하나는 네트워크 모듈에 멀티스케일 프로세스를 통합하여 성능을 향상시킵니다. 공간 의존성을 더 잘 모델링하고 네트워크 구조에 공간적 관심을 통합하기 위한 추가 작업이 모색되고 있다.

In this paper, we investigate a different aspect of network design - the relationship between channels. We introduce a new architectural unit, which we term the Squeeze-andExcitation (SE) block, with the goal of improving the quality of representations produced by a network by explicitly modelling the interdependencies between the channels of its convolutional features. To this end, we propose a mechanism that allows the network to perform feature recalibration, through which it can learn to use global information to selectively emphasise informative features and suppress less useful ones.

이 문서에서는 네트워크 설계의 다른 측면, 즉 채널 간의 관계를 조사합니다. 우리는 컨볼루션 기능의 채널 간의 상호의존성을 명시적으로 모델링함으로써 네트워크에 의해 생산된 표현의 품질을 향상시킨다는 목표로 우리가 Squeeze-and Excitation (SE) 블록이라고 부르는 새로운 아키텍처 유닛을 도입한다. 이를 위해 네트워크가 기능 재보정을 실행할 수 있는 메커니즘을 제안합니다.이를 통해 글로벌 정보를 사용하여 정보 기능을 선택적으로 강조하고 덜 유용한 기능을 억제하는 방법을 배울 수 있습니다.

The structure of the SE building block is depicted in Fig. 1. For any given transformation Ftr mapping the input X to the feature maps U where U ∈ R H×W×C, e.g. a convolution, we can construct a corresponding SE block to perform feature recalibration. The features U are first passed through a squeeze operation, which produces a channel descriptor by aggregating feature maps across their spatial dimensions (H × W). The function of this descriptor is to produce an embedding of the global distribution of channel-wise feature responses, allowing information from the global receptive field of the network to be used by all its layers. The aggregation is followed by an excitation operation, which takes the form of a simple self-gating mechanism that takes the embedding as input and produces a collection of per-channel modulation weights. These weights are applied to the feature maps U to generate the output of the SE block which can be fed directly into subsequent layers of the network.

SE 구성 블록의 구조는 그림 1에 나와 있다. 입력 X를 U r R H×W×C의 특징 지도 U에 매핑하는 변환 Ftr에 대해 예를 들어 컨볼루션에 해당하는 SE 블록을 구성하여 특징 재보정을 수행할 수 있다. 기능 U는 먼저 스퀴즈 연산을 통해 전달됩니다.스퀴즈 연산은 공간 치수(H × W)에 걸친 기능 맵을 집약함으로써 채널 기술자를 생성합니다. 이 디스크립터의 기능은 채널별 기능 응답의 글로벌 배포를 생성하여 네트워크의 글로벌 수신 필드로부터의 정보를 모든 레이어에서 사용할 수 있도록 하는 것입니다. 집계에 이어 들뜸 연산이 뒤따른다. 들뜸 연산은 임베딩을 입력으로 받아 채널별 변조 가중치 컬렉션을 생성하는 단순한 자기 게이트 메커니즘의 형태를 취한다. 이러한 가중치는 피처 맵 U에 적용되어 네트워크의 후속 레이어에 직접 공급될 수 있는 SE 블록의 출력을 생성합니다.

It is possible to construct an SE network (SENet) by simply stacking a collection of SE blocks. Moreover, these SE blocks can also be used as a drop-in replacement for the original block at a range of depths in the network architecture (Section 6.4). While the template for the building block is generic, the role it performs at different depths differs throughout the network. In earlier layers, it excites informative features in a class-agnostic manner, strengthening the shared low-level representations. In later layers, the SE blocks become increasingly specialised, and respond to different inputs in a highly class-specific manner (Section 7.2). As a consequence, the benefits of the feature recalibration performed by SE blocks can be accumulated through the network.

SE 블록을 쌓아 올리는 것만으로 SE 네트워크(SENet)를 구축할 수 있습니다. 더욱이, 이러한 SE 블록은 네트워크 아키텍처의 깊이 범위에서 원래 블록의 드롭인 대체품으로 사용될 수도 있습니다 (섹션 6.4). 빌딩 블록의 템플릿은 범용이지만, 이 템플릿이 수행하는 역할은 네트워크 전체에 따라 다릅니다. 이전 계층에서는 클래스에 구애받지 않는 방식으로 유용한 기능을 활성화하여 공유 하위 수준 표현을 강화합니다. 이후 계층에서, SE 블록은 점점 더 전문화되며, 매우 세분화된 방식으로 다양한 입력에 대응합니다 (섹션 7.2). 그 결과 SE 블록에 의해 실행되는 기능 재보정의 이점을 네트워크를 통해 축적할 수 있다.

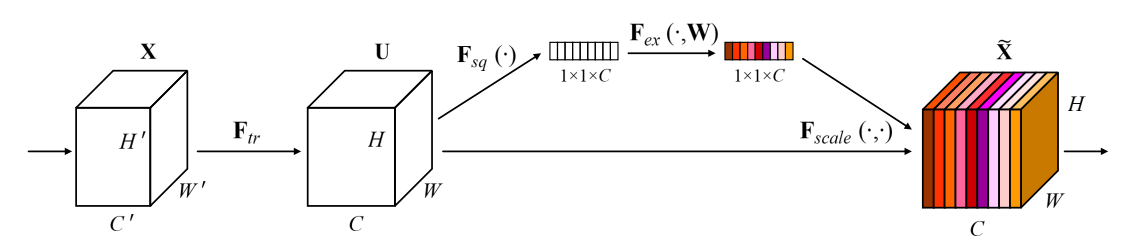


Fig. 1. A Squeeze-and-Excitation block

The design and development of new CNN architectures is a difficult engineering task, typically requiring the selection of many new hyperparameters and layer configurations. By contrast, the structure of the SE block is simple and can be used directly in existing state-of-the-art architectures by replacing components with their SE counterparts, where the performance can be effectively enhanced. SE blocks are also computationally lightweight and impose only a slight increase in model complexity and computational burden.

새로운 CNN 아키텍처의 설계와 개발은 어려운 엔지니어링 작업이며, 일반적으로 많은 새로운 하이퍼 파라미터와 레이어 구성을 선택해야 한다. 반면 SE 블록의 구조는 단순하고 기존 최첨단 아키텍처에서 구성 요소를 SE 블록으로 교체하여 직접 사용할 수 있으므로 성능이 효과적으로 향상됩니다. SE 블록은 또한 계산적으로 가볍고 모델 복잡성과 계산 부하가 약간 증가했을 뿐입니다.

To provide evidence for these claims, we develop several SENets and conduct an extensive evaluation on the ImageNet dataset. We also present results beyond ImageNet that indicate that the benefits of our approach are not restricted to a specific dataset or task. By making use of SENets, we ranked first in the ILSVRC 2017 classification competition. Our best model ensemble achieves a 2.251% top-5 error on the test set1. This represents roughly a 25% relative improvement when compared to the winner entry of the previous year (top-5 error of 2.991%).

이러한 주장에 대한 증거를 제공하기 위해 NAT은 몇 가지 SENet을 개발하고 ImageNet 데이터 세트에 대한 광범위한 평가를 실시합니다. 또, 델의 어프로치의 메리트가 특정의 데이터 세트나 태스크에 한정되지 않는 것을 나타내는, ImageNet 이외의 결과도 제시합니다. SENets를 활용하여 ILSVRC 2017 등급 공모에서 1위를 차지하였습니다. 최고의 모델 앙상블은 테스트 세트1에서 2.251%의 상위 5위 오차를 달성합니다. 이는 전년도 수상자 응모(상위 5위 오차 2.991%)와 비교하여 약 25%의 상대적 개선을 나타낸 것입니다.

**2. Related Work**

**Deeper architectures.** VGGNets and Inception models showed that increasing the depth of a network could significantly increase the quality of representations that it was capable of learning. By regulating the distribution of the inputs to each layer, Batch Normalization (BN) added stability to the learning process in deep networks and produced smoother optimisation surfaces. Building on these works, ResNets demonstrated that it was possible to learn considerably deeper and stronger networks through the use of identity-based skip connections. Highway networks introduced a gating mechanism to regulate the flow of information along shortcut connections. Following these works, there have been further reformulations of the connections between network layers, which show promising improvements to the learning and representational properties of deep networks.

**더 깊은 아키텍처.** VGGNets 및 Inception 모델은 네트워크의 깊이를 높이면 학습할 수 있는 표현의 품질이 크게 향상될 수 있음을 보여 주었습니다. BN(Batch Normalization)은 각 계층에 대한 입력의 분포를 조절함으로써 심층 네트워크의 학습 프로세스에 안정성을 더하고 보다 매끄러운 최적화 표면을 생성했습니다. 이러한 작업을 바탕으로 ResNets는 아이덴티티 기반 스킵 접속을 사용하여 훨씬 더 깊고 강력한 네트워크를 학습할 수 있음을 입증했습니다. 고속도로 네트워크는 바로 가기 연결을 따라 정보의 흐름을 조절하는 게이트 메커니즘을 도입했습니다. 이러한 작업에 따라, 네트워크 계층 간 연결에 대한 추가적인 재구성이 있었으며, 이는 딥 네트워크의 학습 및 표현 특성에 대한 유망한 개선을 보여준다.

An alternative, but closely related line of research has focused on methods to improve the functional form of the computational elements contained within a network. Grouped convolutions have proven to be a popular approach for increasing the cardinality of learned transformations. More flexible compositions of operators can be achieved with multi-branch convolutions which can be viewed as a natural extension of the grouping operator. In prior work, cross-channel correlations are typically mapped as new combinations of features, either independently of spatial structure or jointly by using standard convolutional filters with 1 × 1 convolutions. Much of this research has concentrated on the objective of reducing model and computational complexity, reflecting an assumption that channel relationships can be formulated as a composition of instance-agnostic functions with local receptive fields. In contrast, we claim that providing the unit with a mechanism to explicitly model dynamic, non-linear dependencies between channels using global information can ease the learning process, and significantly enhance the representational power of the network.

대체적이지만 밀접하게 관련된 연구 라인은 네트워크 내에 포함된 계산 요소의 기능적 형태를 개선하는 방법에 초점을 맞추고 있다. 그룹화된 컨볼루션은 학습된 변환의 카디널리티를 높이기 위한 일반적인 접근 방식임이 입증되었습니다. 그룹화 연산자의 자연스러운 확장으로 볼 수 있는 다중 분기 컨볼루션을 통해 연산자의 보다 유연한 구성을 달성할 수 있습니다. 이전 연구에서 교차 채널 상관관계는 일반적으로 공간 구조와 독립적으로 또는 1 × 1 컨볼루션을 가진 표준 컨볼루션 필터를 사용하여 기능의 새로운 조합으로 매핑된다. 이 연구의 대부분은 모델 및 계산의 복잡성을 줄이는 목적에 초점을 맞추고 있으며, 이는 채널 관계가 로컬 수용 분야와 인스턴스 독립 함수의 구성으로서 공식화될 수 있다는 가정을 반영하고 있습니다. 이와는 대조적으로, 우리는 글로벌 정보를 사용하여 채널 간의 동적 비선형 종속성을 명시적으로 모델링하는 메커니즘을 장치에 제공하는 것이 학습 과정을 용이하게 하고 네트워크의 대표력을 크게 향상시킬 수 있다고 주장한다.

**Algorithmic Architecture Search**. Alongside the works described above, there is also a rich history of research that aims to forgo manual architecture design and instead seeks to learn the structure of the network automatically. Much of the early work in this domain was conducted in the neuro-evolution community, which established methods for searching across network topologies with evolutionary methods. While often computationally demanding, evolutionary search has had notable successes which include finding good memory cells for sequence models and learning sophisticated architectures for largescale image classification. With the goal of reducing the computational burden of these methods, efficient alternatives to this approach have been proposed based on Lamarckian inheritance and differentiable architecture search.

알고리즘 아키텍처 검색. 위에서 설명한 작업과 함께, 수동 아키텍처 설계를 포기하고 네트워크의 구조를 자동으로 학습하는 것을 목표로 하는 풍부한 연구 역사도 있습니다. 이 영역의 초기 작업의 대부분은 진화적 방법을 사용하여 네트워크 토폴로지를 검색하는 방법을 확립한 신경 진화 커뮤니티에서 수행되었습니다. 진화적 검색은 종종 계산 부하가 높지만 시퀀스 모델에 적합한 메모리 셀을 찾고 대규모 이미지 분류를 위한 정교한 아키텍처를 학습하는 등 주목할 만한 성공을 거두었습니다. 이러한 방법의 계산 부담을 줄이기 위해, 라마르크식 유전과 차별화된 아키텍처 검색을 기반으로 이 접근법에 대한 효율적인 대안이 제안되었습니다.

By formulating architecture search as hyperparameter optimisation, random search and other more sophisticated model-based optimisation techniques can also be used to tackle the problem. Topology selection as a path through a fabric of possible designs and direct architecture prediction have been proposed as additional viable architecture search tools. Particularly strong results have been achieved with techniques from reinforcement learning. SE blocks can be used as atomic building blocks for these search algorithms, and were demonstrated to be highly effective in this capacity in concurrent work.

아키텍처 검색을 하이퍼 파라미터 최적화로 공식화함으로써 랜덤 검색 및 기타 보다 정교한 모델 기반 최적화 기술을 사용하여 문제를 해결할 수 있습니다. 가능한 설계의 패브릭을 통한 경로로서의 토폴로지 선택과 직접적인 아키텍처 예측이 실행 가능한 추가 아키텍처 검색 도구로 제안되었습니다. 특히, 강화 학습의 기법으로 강한 성과를 얻고 있습니다. SE 블록은 이러한 검색 알고리즘을 위한 원자 구성 블록으로 사용될 수 있으며, 동시 작업에서 이 용량에서 매우 효과적인 것으로 입증되었다.

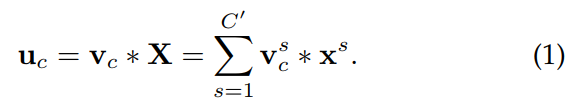
**Attention and gating mechanisms.** Attention can be interpreted as a means of biasing the allocation of available computational resources towards the most informative components of a signal. Attention mechanisms have demonstrated their utility across many tasks including sequence learning, localisation and understanding in images, image captioning and lip reading. In these applications, it can be incorporated as an operator following one or more layers representing higher-level abstractions for adaptation between modalities. Some works provide interesting studies into the combined use of spatial and channel attention. Wang et al. introduced a powerful trunk-and-mask attention mechanism based on hourglass modules that is inserted between the intermediate stages of deep residual networks. By contrast, our proposed SE block comprises a lightweight gating mechanism which focuses on enhancing the representational power of the network by modelling channel-wise relationships in a computationally efficient manner.

**주의 및 게이트 메커니즘.** 주의는 신호의 가장 유익한 구성요소에 사용 가능한 계산 자원을 할당하는 수단으로 해석될 수 있다. 주의 메커니즘은 이미지에서의 시퀀스 학습, 위치 파악 및 이해, 이미지 캡션 및 입술 판독을 포함한 많은 작업에 걸쳐 유용성을 입증했다. 이러한 어플리케이션에서는, 양식간의 적응을 위해서, 상위 레벨의 추상화를 나타내는 하나 이상의 레이어에 이은 연산자로서 통합될 수 있다. 몇몇 작품들은 공간과 채널 주의의 결합 사용에 대한 흥미로운 연구를 제공한다. Wang 등은 심층 잔류 네트워크의 중간 단계 사이에 삽입되는 모래시계 모듈에 기초한 강력한 트렁크 앤 마스크 주의 메커니즘을 도입했다. 대조적으로, 우리가 제안한 SE 블록은 계산적으로 효율적인 방식으로 채널별 관계를 모델링함으로써 네트워크의 표현력을 향상시키는 데 초점을 맞춘 경량 게이트 메커니즘으로 구성됩니다.

**3. Squeeze-And-Excitation Blocks**

A Squeeze-and-Excitation block is a computational unit which can be built upon a transformation Ftr mapping an input X ∈ R H0×W0×C0 to feature maps U ∈ R H×W×C . In the notation that follows we take Ftr to be a convolutional operator and use V = [v1, v2, . . . , vC ] to denote the learned set of filter kernels, where vc refers to the parameters of the c-th filter. We can then write the outputs as U = [u1, u2, . . . , uC ], where

Squeeze-and-Excitation 블록은 입력 X × R H0×W0×C를 특징 지도 U . R H×W×C에 매핑하는 변환 Ftr 위에 구축될 수 있는 계산 단위입니다. 다음 표기법에서는 Ftr을 컨볼루션 연산자로 간주하고 V = v1, v2를 사용합니다. c번째 필터 그런 다음 출력을 U = [u1, u2, ., uC]로 쓸 수 있습니다.



Here ∗ denotes convolution, vc = [v1c, v2c, . . . , vC0c], X =[x1, x2, . . . , xC0] and uc ∈ RH×W. vs c is a 2D spatial kernel representing a single channel of vc that acts on the corresponding channel of X. To simplify the notation, bias terms are omitted. Since the output is produced by a summation through all channels, channel dependencies are implicitly embedded in vc, but are entangled with the local spatial correlation captured by the filters. The channel relationships modelled by convolution are inherently implicit and local (except the ones at top-most layers). We expect the learning of convolutional features to be enhanced by explicitly modelling channel interdependencies, so that the network is able to increase its sensitivity to informative features which can be exploited by subsequent transformations. Consequently, we would like to provide it with access to global information and recalibrate filter responses in two steps, squeeze and excitation, before they are fed into the next transformation. A diagram illustrating the structure of an SE block is shown in Fig. 1.

여기서 θ는 컨볼루션, vc = [v1c, v2c, . , vC0c], X = [x1, x2, . , xC0] 및 uc δ RH×W를 나타냅니다.vs c는 X의 대응하는 채널에 작용하는 VC의 단일 채널을 나타내는 2D 공간 커널입니다. 표기법을 단순화하기 위해 치우침 항은 생략됩니다. 출력은 모든 채널의 합계에 의해 생성되기 때문에 채널 의존성은 암묵적으로 VC에 포함되어 있지만 필터에 의해 캡처된 로컬 공간 상관과 얽혀 있습니다. 컨볼루션에 의해 모델링된 채널 관계는 본질적으로 암묵적이고 로컬적입니다(최상단 레이어에 있는 관계는 제외). 우리는 네트워크가 후속 변환에 의해 이용될 수 있는 정보 기능에 대한 민감도를 높일 수 있도록 채널 상호의존성을 명시적으로 모델링함으로써 컨볼루션 기능의 학습이 향상될 것으로 기대한다. 따라서 글로벌 정보에 대한 접근을 제공하고 필터 응답을 다음 변환에 공급하기 전에 squeeze와 excitation의 2단계로 재보정하고 싶습니다. 그림 1에 SE 블록의 구조를 나타내는 그림을 나타낸다.

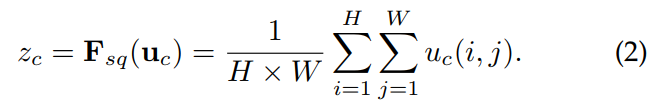
**3.1. Squeeze: Global Information Embedding**

In order to tackle the issue of exploiting channel dependencies, we first consider the signal to each channel in the output features. Each of the learned filters operates with a local receptive field and consequently each unit of the transformation output U is unable to exploit contextual information outside of this region.

채널 의존성 이용 문제에 대처하기 위해 먼저 출력 기능의 각 채널에 대한 신호를 고려합니다. 학습된 각 필터는 로컬 리셉티브 필드와 함께 동작하므로 변환출력 U의 각 유닛은 이 영역 밖에서 컨텍스트 정보를 이용할 수 없다.

To mitigate this problem, we propose to squeeze global spatial information into a channel descriptor. This is achieved by using global average pooling to generate channel-wise statistics. Formally, a statistic z ∈ R C is generated by shrinking U through its spatial dimensions H × W, such that the c-th element of z is calculated by:

이 문제를 완화하기 위해 글로벌 공간 정보를 채널 기술자로 압축할 것을 제안합니다. 이는 글로벌 평균 풀링을 사용하여 채널별 통계정보를 생성함으로써 실현됩니다. 공식적으로, z의 c번째 요소는 다음과 같이 계산되도록 공간 치수 H × W를 통해 U를 축소함으로써 통계 z δ R C가 생성된다.



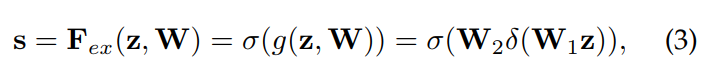
*Discussion.* The output of the transformation U can be interpreted as a collection of the local descriptors whose statistics are expressive for the whole image. Exploiting such information is prevalent in prior feature engineering work. We opt for the simplest aggregation technique, global average pooling, noting that more sophisticated strategies could be employed here as well.

*논의* 변환 U의 출력은 전체 이미지에 대해 통계 정보가 표현되는 로컬 기술자의 집합으로 해석할 수 있습니다. 이러한 정보의 이용은 이전의 기능 엔지니어링 작업에서는 널리 행해지고 있습니다. 델은 가장 단순한 집약 기법인 글로벌 평균 풀링을 선택하고 있으며, 여기서도 보다 정교한 전략을 사용할 수 있다는 점에 주목하고 있습니다.

**3.2. Excitation: Adaptive Recalibration**

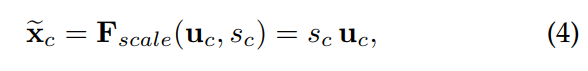
To make use of the information aggregated in the squeeze operation, we follow it with a second operation which aims to fully capture channel-wise dependencies. To fulfil this objective, the function must meet two criteria: first, it must be flexible (in particular, it must be capable of learning a nonlinear interaction between channels) and second, it must learn a non-mutually-exclusive relationship since we would like to ensure that multiple channels are allowed to be emphasised (rather than enforcing a one-hot activation). To meet these criteria, we opt to employ a simple gating mechanism with a sigmoid activation:

스퀴즈 조작으로 집약된 정보를 활용하기 위해 채널별 의존관계를 완전히 캡처하는 것을 목적으로 하는 두 번째 조작을 수행합니다. 이 목적을 달성하기 위해 함수는 두 가지 기준을 충족해야 한다. 첫째, 유연해야 한다(특히 채널 간의 비선형 상호작용을 학습할 수 있어야 한다). 둘째, 여러 채널이 강조되도록 (강제하는 대신) 보장되기를 원하기 때문에 상호 배타적 관계를 학습해야 한다. 원핫 액티베이션). 이러한 기준을 충족하기 위해 Sigmoid 활성화가 있는 단순한 게이트 메커니즘을 채택하기로 선택하였습니다.



where δ refers to the ReLU [63] function, W1 ∈ R Cr×C and W2 ∈ R C×Cr. To limit model complexity and aid generalisation, we parameterise the gating mechanism by forming a bottleneck with two fully-connected (FC) layers around the non-linearity, i.e. a dimensionality-reduction layer with reduction ratio r (this parameter choice is discussed in Section 6.1), a ReLU and then a dimensionality-increasing layer returning to the channel dimension of the transformation output U. The final output of the block is obtained by rescaling U with the activations s:

여기서 θ는 ReLU [63] 함수, W1 r R Cr×C 및 W2 r R C×Cr을 의미한다. 모델 복잡성을 제한하고 일반화를 지원하기 위해 비선형성 주위에 2개의 완전 연결(FC) 계층으로 병목 현상을 형성하여 게이트 메커니즘을 매개변수화한다. 즉, 감소 비율이 r인 치수 감소 계층, ReLU 및 차원 증가 계층 반환 t.o 변환 출력 U의 채널 치수. 블록의 최종 출력은 활성화로 U를 재스케일링하여 얻을 수 있습니다.



where Xe = [xe1, xe2, . . . , xeC ] and Fscale(uc, sc) refers to channel-wise multiplication between the scalar sc and the feature map uc ∈ R H×W.

여기서 Xe = [xe1, xe2, ., xeC] 및 Fscale(uc, sc)은 스칼라 sc와 피쳐 맵 uc r R H×W 사이의 채널별 곱셈을 나타냅니다.

*Discussion.* The excitation operator maps the inputspecific descriptor z to a set of channel weights. In this regard, SE blocks intrinsically introduce dynamics conditioned on the input, which can be regarded as a selfattention function on channels whose relationships are not confined to the local receptive field the convolutional filters are responsive to.

*논의* 여기 연산자는 입력 고유의 설명자 z를 채널 가중치 세트에 매핑합니다. 이와 관련하여 SE 블록은 본질적으로 입력에 조건부 역학을 도입하며, 이는 컨볼루션 필터가 응답하는 로컬 리셉티브 필드에 국한되지 않는 관계를 가진 채널에서의 자기 주의 기능으로 간주할 수 있다.

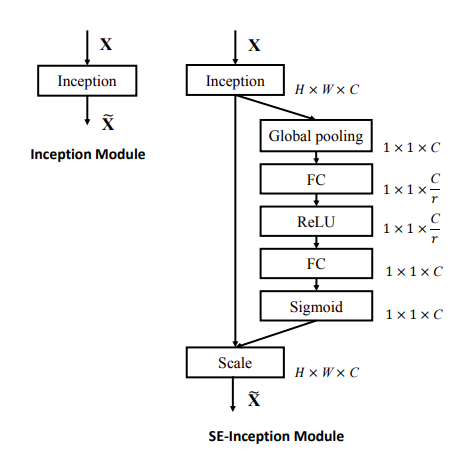


Fig. 2. The schema of the original Inception module (left) and the SEInception module (right).

그림 2. 원래 Inception 모듈(왼쪽)과 SEInception 모듈(오른쪽)의 스키마.

**3.3. Instantiations**

The SE block can be integrated into standard architectures such as VGGNet by insertion after the non-linearity following each convolution. Moreover, the flexibility of the SE block means that it can be directly applied to transformations beyond standard convolutions. To illustrate this point, we develop SENets by incorporating SE blocks into several examples of more complex architectures, described next.

SE 블록은 VGGNet과 같은 표준 아키텍처에 통합될 수 있습니다. 각 컨볼루션에 이은 비선형성 뒤에 삽입됩니다. 또한 SE 블록의 유연성은 표준 컨볼루션 이상의 변환에 직접 적용할 수 있음을 의미합니다. 이 점을 설명하기 위해 SE 블록을 좀 더 복잡한 아키텍처의 몇 가지 예에 통합하여 SEets를 개발했습니다.

We first consider the construction of SE blocks for Inception networks. Here, we simply take the transformation Ftr to be an entire Inception module (see Fig. 2) and by making this change for each such module in the architecture, we obtain an SE-Inception network. SE blocks can also be used directly with residual networks (Fig. 3 depicts the schema of an SE-ResNet module). Here, the SE block transformation Ftr is taken to be the non-identity branch of a residual module. Squeeze and Excitation both act before summation with the identity branch. Further variants that integrate SE blocks with ResNeXt, Inception-ResNet, MobileNet and ShuffleNet can be constructed by following similar schemes. For concrete examples of SENet architectures, a detailed description of SE-ResNet-50 and SE-ResNeXt-50 is given in Table 1.

먼저 Inception 네트워크를 위한 SE 블록 구축을 검토한다. 여기서는 변환 Ftr을 전체 Inception 모듈(그림 2 참조)로 간주하고 아키텍처의 각 모듈에 대해 이 변경을 가함으로써 SE-Inception 네트워크를 얻을 수 있습니다. SE 블록은 잔류 네트워크에서도 직접 사용할 수 있습니다(그림 3은 SE-ResNet 모듈의 스키마를 나타내고 있습니다). 여기서 SE 블록 변환 Ftr은 잔류 모듈의 비등식 분기로 간주된다. squeeze와 Excitation은 모두 아이덴티티 브랜치와의 합계 전에 동작합니다. SE 블록을 ResNeXt, Inception-ResNet, MobileNet 및 ShuffleNet과 통합하는 추가 변형은 유사한 방식을 따라 구성할 수 있습니다. SENet 아키텍처의 구체적인 예는 표 1에 SE-ResNet-50 및 SE-ResNeXt-50에 대한 자세한 설명이 나와 있습니다.

One consequence of the flexible nature of the SE block is that there are several viable ways in which it could be integrated into these architectures. Therefore, to assess sensitivity to the integration strategy used to incorporate SE blocks into a network architecture, we also provide ablation experiments exploring different designs for block inclusion in Section 6.5.

SE 블록의 유연성이 가져오는 결과 중 하나는 이러한 아키텍처에 통합할 수 있는 몇 가지 실행 가능한 방법이 있다는 것입니다. 따라서 SE 블록을 네트워크 아키텍처에 통합하는 데 사용되는 통합 전략에 대한 민감도를 평가하기 위해 섹션 6.5에서 블록 포함을 위한 다양한 설계를 탐색하는 절제 실험도 제공한다.

**4. Model and Computational Complexity**

For the proposed SE block design to be of practical use, it must offer a good trade-off between improved performance and increased model complexity. To illustrate the computational burden associated with the module, we consider a comparison between ResNet-50 and SE-ResNet-50 as an example. ResNet-50 requires ∼3.86 GFLOPs in a single forward pass for a 224 × 224 pixel input image. Each SE block makes use of a global average pooling operation in the squeeze phase and two small FC layers in the excitation phase, followed by an inexpensive channel-wise scaling operation. In the aggregate, when setting the reduction ratio r (introduced in Section 3.2) to 16, SE-ResNet-50 requires ∼3.87 GFLOPs, corresponding to a 0.26% relative increase over the original ResNet-50. In exchange for this slight additional computational burden, the accuracy of SE-ResNet-50 surpasses that of ResNet-50 and indeed, approaches that of a deeper ResNet-101 network requiring ∼7.58 GFLOPs (Table 2).

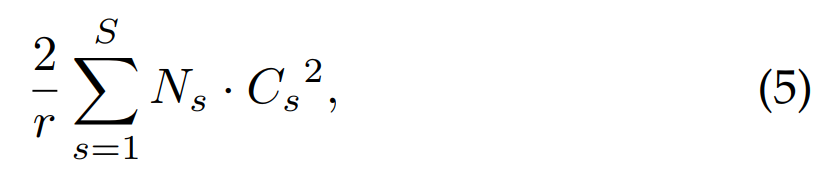
제안된 SE 블록 설계가 실용적이 되려면 성능 향상과 모델 복잡성 증가 사이의 균형을 잘 맞춰야 합니다. 모듈과 관련된 계산 부담을 설명하기 위해 ResNet-50과 SE-ResNet-50의 비교를 예로 들겠습니다. ResNet-50은 224 × 224 픽셀 입력 이미지에 대해 단일 포워드 패스로 최대 3.86 GFLOP가 필요합니다. 각 SE 블록은 스퀴즈 단계에서의 글로벌 평균 풀링 동작과 여기 단계에서의 2개의 작은 FC 레이어를 이용하고, 그 후 저렴한 채널 단위의 스케일링 동작을 이용합니다. 집약적으로 환원비 r(섹션 3.2에서 소개)을 16으로 설정할 때 SE-ResNet-50은 원래 ResNet-50에 비해 0.26%의 상대적인 증가에 해당하는 약 3.87 GFLOP를 필요로 한다. 이와 같은 약간의 계산 부하에 대한 교환으로 SE-ResNet-50의 정확도는 ResNet-50의 정확도를 능가하며, 실제로는 최대 7.58 GFLOPS를 필요로 하는 보다 깊은 ResNet-101 네트워크의 정확도에 근접합니다(표 2).

In practical terms, a single pass forwards and backwards through ResNet-50 takes 190 ms, compared to 209 ms for SE-ResNet-50 with a training minibatch of 256 images (both timings are performed on a server with 8 NVIDIA Titan X GPUs). We suggest that this represents a reasonable runtime overhead, which may be further reduced as global pooling and small inner-product operations receive further optimisation in popular GPU libraries. Due to its importance for embedded device applications, we further benchmark CPU inference time for each model: for a 224 × 224 pixel input image, ResNet-50 takes 164 ms in comparison to 167 ms for SE-ResNet-50. We believe that the small additional computational cost incurred by the SE block is justified by its contribution to model performance.

실제로 ResNet-50을 통과하는 단일 패스는 190밀리초 걸립니다.SE-ResNet-50은 256개의 이미지의 트레이닝 미니패치를 사용하여 209밀리초입니다(두 타이밍 모두 8개의 NVIDIA Titan X GPU를 탑재한 서버에서 실행). 이는 합리적인 런타임 오버헤드를 나타내며, 이는 글로벌 풀링 및 소규모 내부 제품 운영이 널리 사용되는 GPU 라이브러리에서 더욱 최적화되기 때문에 더욱 줄어들 수 있습니다. 임베디드 기기 애플리케이션의 중요성으로 인해 각 모델의 CPU 추론 시간을 더욱 벤치마킹합니다. 즉, 224 × 224 픽셀 입력 이미지의 경우 ResNet-50은 SE-ResNet-50의 167 ms에 비해 164 ms가 소요됩니다. SE 블록에 의해 발생하는 작은 추가 계산 비용은 모델 성능에 대한 기여로 정당화될 수 있다고 믿습니다.

We next consider the additional parameters introduced by the proposed SE block. These additional parameters result solely from the two FC layers of the gating mechanism and therefore constitute a small fraction of the total network capacity. Concretely, the total number introduced by the weight parameters of these FC layers is given by:

다음으로 제안된 SE 블록에 의해 도입된 추가 매개변수를 검토한다. 이러한 추가 파라미터는 게이트메커니즘의 2개의 FC 레이어에서만 발생하므로 전체 네트워크 캐퍼시티의 극히 일부를 구성합니다. 구체적으로는 이들 FC 레이어의 중량 파라미터에 의해 도입된 총수는 다음과 같습니다.



where r denotes the reduction ratio, S refers to the number of stages (a stage refers to the collection of blocks operating on feature maps of a common spatial dimension), Cs denotes the dimension of the output channels and Ns denotes the number of repeated blocks for stage s (when bias terms are used in FC layers, the introduced parameters and computational cost are typically negligible). SE-ResNet-50 introduces ∼2.5 million additional parameters beyond the ∼25 million parameters required by ResNet-50, corresponding to a ∼10% increase. In practice, the majority of these parameters come from the final stage of the network, where the excitation operation is performed across the greatest number of channels. However, we found that this comparatively costly final stage of SE blocks could be removed at only a small cost in performance (<0.1% top-5 error on ImageNet) reducing the relative parameter increase to ∼4%, which may prove useful in cases where parameter usage is a key consideration (see Section 6.4 and 7.2 for further discussion).

여기서 r은 감소율을 나타내고, S는 스테이지의 수(단위는 공통 공간 차원의 피쳐 맵에서 동작하는 블록의 집합), Cs는 출력 채널의 치수를 나타내고, Ns는 스테이지 s에 대해 반복되는 블록의 수(FC 레이어에서 바이어스 용어를 사용하는 경우, 도입된 파라미터 및 com)를 나타낸다.추정 비용은 일반적으로 무시할 수 있습니다.) SE-ResNet-50은 ResNet-50에 필요한 약 2,500만 개의 매개 변수보다 최대 250만 개의 추가 매개 변수를 도입하여 최대 10%의 증가에 해당합니다. 실제로 이들 파라미터의 대부분은 네트워크의 최종단계에서 발생하며, 여기서 들뜸 동작은 가장 많은 채널에서 실행됩니다. 그러나 비교적 비용이 많이 드는 SE 블록의 최종 단계는 성능 면에서 적은 비용(ImageNet의 경우 0.1%의 상위 5의 오류)으로 제거할 수 있다는 것을 알게 되었습니다. 이는 상대적인 파라미터 증가를 ~ 4%로 감소시키는 것으로, 파라미터 사용이 중요한 고려 사항인 경우에 유용할 수 있습니다(자세한 설명은 섹션 6.4 및 7.2 참조).

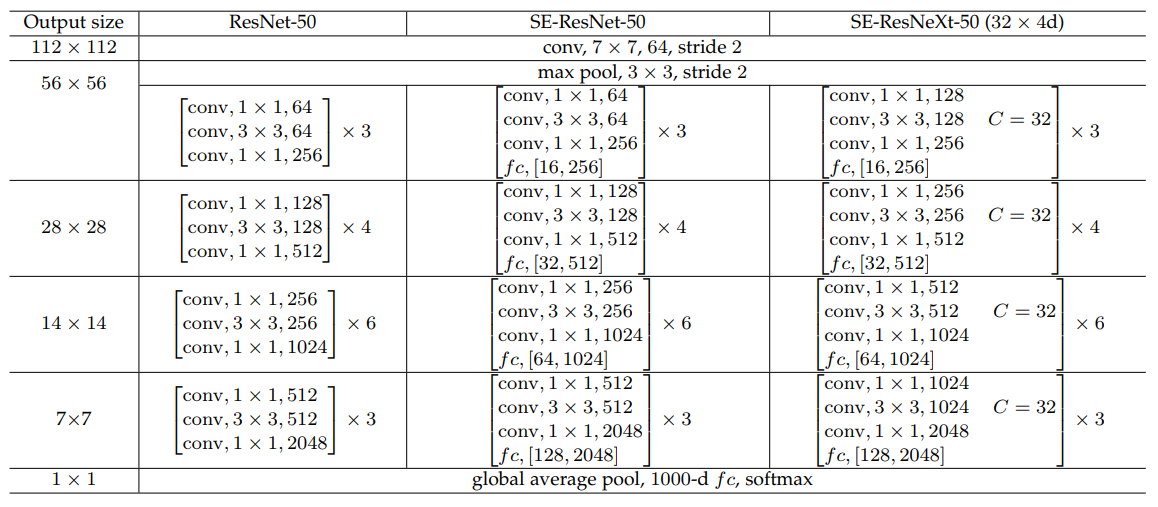


TABLE 1 (Left) ResNet-50. (Middle) SE-ResNet-50. (Right) SE-ResNeXt-50 with a 32×4d template. The shapes and operations with specific parameter settings of a residual building block are listed inside the brackets and the number of stacked blocks in a stage is presented outside. The inner brackets following by fc indicates the output dimension of the two fully connected layers in an SE module.

표 1 (왼쪽) ResNet-50. (중간) SE-ResNet-50. (오른쪽) SE-ResNeXt-50 (32×4d 템플릿 포함) 나머지 구성 블록의 특정 매개변수 설정을 포함한 모양과 연산이 브래킷 안에 나열되고 한 스테이지의 쌓인 블록 수가 외부에 표시됩니다. 안쪽 괄호 뒤에 fc가 붙어 있는 것은 SE 모듈에서 완전히 연결된2개의 레이어의 출력 치수를 나타냅니다.

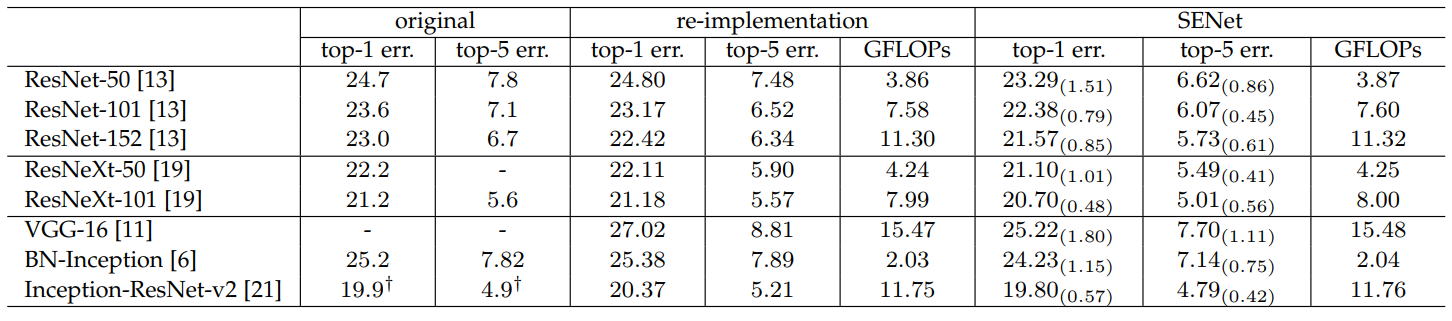


TABLE 2 Single-crop error rates (%) on the ImageNet validation set and complexity comparisons. The original column refers to the results reported in the original papers (the results of ResNets are obtained from the website: https://github.com/Kaiminghe/deep-residual-networks). To enable a fair comparison, we re-train the baseline models and report the scores in the re-implementation column. The SENet column refers to the corresponding architectures in which SE blocks have been added. The numbers in brackets denote the performance improvement over the re-implemented baselines. † indicates that the model has been evaluated on the non-blacklisted subset of the validation set (this is discussed in more detail in [21]), which may slightly improve results. VGG-16 and SE-VGG-16 are trained with batch normalization.

표 2 ImageNet 검증 세트의 싱글 크롭 에러율(%)과 복잡도 비교. 원래 컬럼은 원본 논문에 보고된 결과를 참조합니다(ResNets 결과는 웹사이트 https://github.com/Kaiminghe/deep-residual-networks)에서 얻을 수 있습니다). 공정한 비교를 위해 기준 모델을 재교육하고 재실행 열에 점수를 보고합니다. SENet 열은 SE 블록이 추가된 해당 아키텍처를 나타냅니다. 괄호 안의 숫자는 다시 구현된 기준선에 비해 성능이 향상되었음을 나타냅니다. † 검증 세트의 블랙리스트에 없는 서브셋에 대해 모델이 평가되었음을 나타냅니다(자세한 내용은 [21]에서 설명합니다).이것에 의해, 결과가 약간 개선될 가능성이 있습니다. VGG-16 및 SE-VGG-16은 배치 정규화를 통해 훈련된다.

**5. Experiments**

In this section, we conduct experiments to investigate the effectiveness of SE blocks across a range of tasks, datasets and model architectures.

이 섹션에서는 다양한 작업, 데이터 세트 및 모델 아키텍처에 걸쳐 SE 블록의 효과를 조사하기 위한 실험을 실시합니다.

**5.1. Image Classification**

To evaluate the influence of SE blocks, we first perform experiments on the ImageNet 2012 dataset [10] which comprises 1.28 million training images and 50K validation images from 1000 different classes. We train networks on the training set and report the top-1 and top-5 error on the validation set.

SE 블록의 영향을 평가하기 위해 먼저 128만 개의 교육 이미지와 1000개의 다른 클래스의 50K 검증 이미지로 구성된 ImageNet 2012 데이터 세트 [10]에 대한 실험을 수행합니다. 트레이닝 세트의 네트워크를 트레이닝 해, 검증 세트의 상위 1 에러와 상위 5 에러를 보고합니다.

Each baseline network architecture and its corresponding SE counterpart are trained with identical optimisation schemes. We follow standard practices and perform data augmentation with random cropping using scale and aspect ratio [5] to a size of 224 × 224 pixels (or 299 × 299 for Inception-ResNet-v2 [21] and SE-Inception-ResNet-v2) and perform random horizontal flipping. Each input image is normalised through mean RGB-channel subtraction. All models are trained on our distributed learning system ROCS which is designed to handle efficient parallel training of large networks. Optimisation is performed using synchronous SGD with momentum 0.9 and a minibatch size of 1024. The initial learning rate is set to 0.6 and decreased by a factor of 10 every 30 epochs. Models are trained for 100 epochs from scratch, using the weight initialisation strategy described in [66]. The reduction ratio r (in Section 3.2) is set to 16 by default (except where stated otherwise).

각 베이스라인 네트워크 아키텍처와 대응하는 SE 아키텍처는 동일한 최적화 스킴으로 훈련됩니다. 표준 관행을 따르고 224 × 224 픽셀 크기(Inception-ResNet-v2 [21] 및 SE-Inception-ResNet-v2의 경우 299 × 299)에 대한 스케일 및 애스펙트 비 [5]를 사용하여 랜덤 크롭을 사용하여 데이터 증가를 수행하고 임의 수평 플립을 수행합니다. 각 입력 화상은 평균 RGB 채널 감산을 통해 정규화된다. 모든 모델은 대규모 네트워크의 효율적인 병렬 훈련을 처리하기 위해 설계된 분산 학습 시스템 ROCS에 기반하여 교육됩니다. 최적화는 모멘텀 0.9와 미니배치 크기 1024의 동기 SGD를 사용하여 수행됩니다. 초기 학습 속도는 0.6으로 설정되며 30에폭마다 10배씩 감소합니다. 모델은 [66]에 기술된 체중 초기화 전략을 사용하여 처음부터 100에폭 동안 훈련된다. (섹션 3.2의) 감소비 r은 기본적으로 16으로 설정됩니다(특별히 명시되지 않은 경우 제외).

When evaluating the models we apply centre-cropping so that 224 × 224 pixels are cropped from each image, after its shorter edge is first resized to 256 (299 × 299 from each image whose shorter edge is first resized to 352 for Inception-ResNet-v2 and SE-Inception-ResNet-v2).

모델을 평가할 때 각 이미지에서 224 × 224 픽셀이 잘리도록 중앙 자르기 기능을 적용하여 짧은 가장자리가 먼저 256으로 조정된 후(Inception-ResNet-v2 및 SE-Inception-ResNet-v2의 경우 352로 크기가 조정된 각 이미지에서 299 × 299).

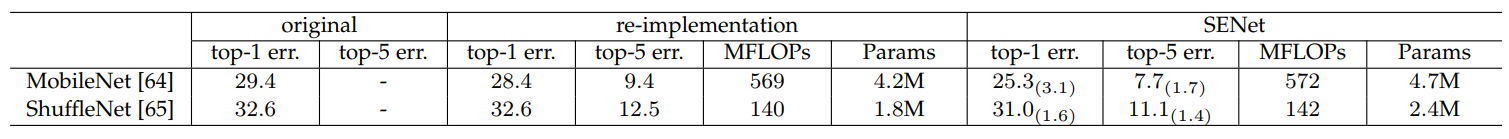


TABLE 3 Single-crop error rates (%) on the ImageNet validation set and complexity comparisons. MobileNet refers to “1.0 MobileNet-224” in [64] and ShuffleNet refers to “ShuffleNet 1 × (g = 3)” in [65]. The numbers in brackets denote the performance improvement over the re-implementation.

표 3 ImageNet 검증 세트의 싱글 크롭 에러율(%)과 복잡도 비교. MobileNet은 [64]에서 "1.0 MobileNet-224"를 참조하고 ShuffleNet은 [65]에서 "ShuffleNet 1 × (g = 3)"을 참조합니다. 괄호 안의 숫자는 재실장 시 성능 향상을 나타냅니다.

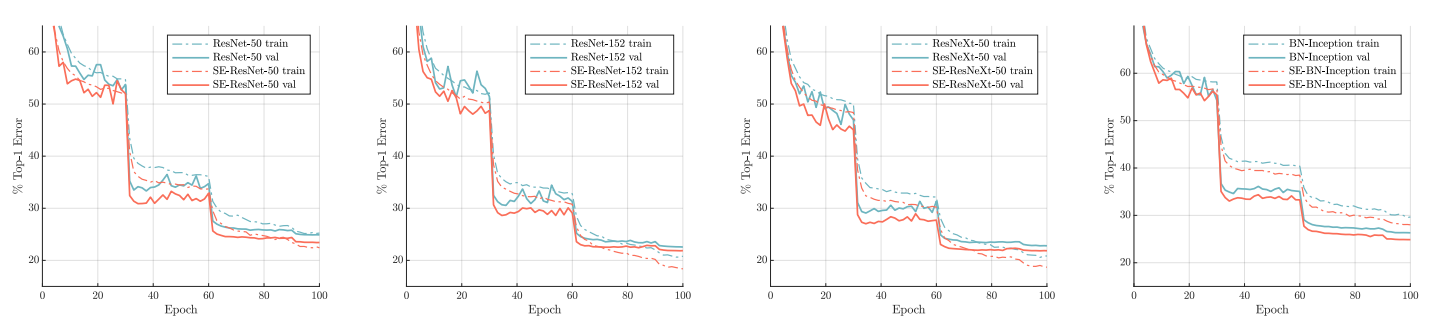


Fig. 4. Training baseline architectures and their SENet counterparts on ImageNet. SENets exhibit improved optimisation characteristics and produce consistent gains in performance which are sustained throughout the training process.

그림 4. ImageNet에서의 베이스라인 아키텍처와 그 SENet 아키텍처 트레이닝 SENets는 개선된 최적화 특성을 보이며 교육 과정 전체에 걸쳐 일관된 성능 향상을 실현합니다.

**Network depth.** We begin by comparing SE-ResNet against ResNet architectures with different depths and report the results in Table 2. We observe that SE blocks consistently improve performance across different depths with an extremely small increase in computational complexity. Remarkably, SE-ResNet-50 achieves a single-crop top-5 validation error of 6.62%, exceeding ResNet-50 (7.48%) by 0.86% and approaching the performance achieved by the much deeper ResNet-101 network (6.52% top-5 error) with only half of the total computational burden (3.87 GFLOPs vs. 7.58 GFLOPs). This pattern is repeated at greater depth, where SE-ResNet-101 (6.07% top-5 error) not only matches, but outperforms the deeper ResNet-152 network (6.34% top-5 error) by 0.27%. While it should be noted that the SE blocks themselves add depth, they do so in an extremely computationally efficient manner and yield good returns even at the point at which extending the depth of the base architecture achieves diminishing returns. Moreover, we see that the gains are consistent across a range of different network depths, suggesting that the improvements induced by SE blocks may be complementary to those obtained by simply increasing the depth of the base architecture.

**네트워크 깊이.** 우선 SE-ResNet과 ResNet 아키텍처의 깊이가 다른 ResNet을 비교하여 그 결과를 표 2에 보고합니다. SE 블록은 계산 복잡도가 극히 적은 수준이지만 서로 다른 깊이에서 일관되게 성능을 향상시킨다는 것을 알 수 있습니다. 놀랍게도 SE-ResNet-50은 6.62%의 싱글 크롭 TOP-5 검증 오류를 달성하여 ResNet-50(7.48%)을 0.86% 웃돌고 훨씬 더 깊은 ResNet-101 네트워크(상위 6.52%의 오류)에 의해 달성된 퍼포먼스에 근접하여 계산 부하가 절반에 불과합니다(787 VS FLOP). 이 패턴은 더 깊은 곳에서 반복되며 SE-ResNet-101(6.07% 상위5% 오류)은 일치할 뿐만 아니라 더 깊은 ResNet-152 네트워크(6.34% 상위5 오류)를 0.27% 능가합니다. SE 블록 자체가 깊이를 더한다는 점에 유의해야 하지만, 이러한 블록은 매우 효율적인 계산 방식으로 깊이를 더하고 기본 아키텍처의 깊이를 확장하면 수익률이 감소하는 시점에서도 좋은 수익을 창출합니다. 더욱이, 우리는 이득이 다양한 네트워크 깊이 범위에 걸쳐 일관된다는 것을 알 수 있으며, 이는 SE 블록에 의해 유도되는 개선이 단순히 기본 아키텍처의 깊이를 증가시킴으로써 얻어지는 개선사항과 보완될 수 있음을 시사한다.

**Integration with modern architectures**. We next study the effect of integrating SE blocks with two further state-of-the-art architectures, Inception-ResNet-v2 and ResNeXt (using the setting of 32 × 4d), both of which introduce additional computational building blocks into the base network. We construct SENet equivalents of these networks, SE-Inception-ResNet-v2 and SE-ResNeXt (the configuration of SE-ResNeXt-50 is given in Table 1) and report results in Table 2. As with the previous experiments, we observe significant performance improvements induced by the introduction of SE blocks into both architectures. In particular, SE-ResNeXt-50 has a top-5 error of 5.49% which is superior to both its direct counterpart ResNeXt-50 (5.90% top-5 error) as well as the deeper ResNeXt-101 (5.57% top-5 error), a model which has almost twice the total number of parameters and computational overhead. We note a slight difference in performance between our re-implementation of Inception-ResNet-v2 and the result reported in [21]. However, we observe a similar trend with regard to the effect of SE blocks, finding that SE counterpart (4.79% top-5 error) outperforms our reimplemented Inception-ResNet-v2 baseline (5.21% top-5 error) by 0.42% as well as the reported result in [21].

**최신 아키텍처와의 통합.** 다음으로 SE 블록을 2개의 최신 아키텍처인 Inception-ResNet-v2 및 ResNeXt(32 × 4d 설정 사용)와 통합하는 효과를 연구한다. 이 두 아키텍처는 모두 기본 네트워크에 추가적인 계산 구성 요소를 도입한다. 이러한 네트워크에 대응하는 SE-Inception-ResNet-v2 및 SE-ResNeXt(SE-ResNeXt-50 의 설정은 표 1 에 나타나 있습니다)를 구축하고, 그 결과를 표 2 에 나타냅니다. 이전 실험과 마찬가지로, 우리는 두 아키텍처에 SE 블록을 도입함으로써 유발된 상당한 성능 향상을 관찰했습니다. 특히 SE-ResNeXt-50의 상위 5개 오류는 5.49%로 직접적인 ResNeXt-50(5.90% 상위 5개 오류)뿐만 아니라 전체 매개 변수 수와 계산 오버헤드의 두 배에 가까운 더 깊은 ResNeXt-101(5.57% 상위 5개 오류)보다 우수합니다. 우리는 Inception-ResNet-v2의 재이행과 [21]에 보고된 결과 사이에 약간의 성능 차이가 있음을 주목한다. 그러나 SE 블록의 영향과 관련하여 유사한 경향을 관찰했는데, SE 대응 요소(4.79% 상위 5개 오류)가 [21]에서 보고된 결과와 함께 재실장된 Inception-ResNet-v2 기준선(5.21% 상위 5개 오류)을 0.42% 능가하는 것으로 나타났다.

We also assess the effect of SE blocks when operating on non-residual networks by conducting experiments with the VGG-16 and BN-Inception architecture. To facilitate the training of VGG-16 from scratch, we add Batch Normalization layers after each convolution. We use identical training schemes for both VGG-16 and SE-VGG-16. The results of the comparison are shown in Table 2. Similarly to the results reported for the residual baseline architectures, we observe that SE blocks bring improvements in performance on the non-residual settings.

또한 VGG-16 및 BN-Inception 아키텍처를 사용하여 실험을 실시함으로써 비잔존 네트워크 상에서 동작할 때 SE 블록의 영향을 평가합니다. VGG-16의 트레이닝을 처음부터 용이하게 하기 위해 각 컨볼루션 후에 배치 정규화 레이어를 추가합니다. VGG-16과 SE-VGG-16에 대해 동일한 훈련 체계를 사용합니다. 비교 결과는 표 2와 같다. 잔류 기준선 아키텍처에 대해 보고된 결과와 마찬가지로, SE 블록이 잔류하지 않은 설정의 성능을 향상시킨다는 것을 알 수 있습니다.

To provide some insight into influence of SE blocks on the optimisation of these models, example training curves for runs of the baseline architectures and their respective SE counterparts are depicted in Fig. 4. We observe that SE blocks yield a steady improvement throughout the optimisation procedure. Moreover, this trend is fairly consistent across a range of network architectures considered as baselines.

SE 블록이 이러한 모델의 최적화에 미치는 영향에 대한 통찰력을 제공하기 위해, 기준선 아키텍처와 각 SE 블록의 실행에 대한 훈련 곡선의 예가 그림 4에 나와 있다. SE 블록은 최적화 절차 전반에 걸쳐 꾸준히 개선되는 것을 관찰한다. 게다가 이러한 경향은 베이스라인으로 간주되는 네트워크 아키텍처의 범위 전체에 걸쳐 상당히 일관성이 있습니다.

**Mobile setting.** Finally, we consider two representative architectures from the class of mobile-optimised networks, MobileNet [64] and ShuffleNet [65]. For these experiments, we used a minibatch size of 256 and slightly less aggressive data augmentation and regularisation as in [65]. We trained the models across 8 GPUs using SGD with momentum (set to 0.9) and an initial learning rate of 0.1 which was reduced by a factor of 10 each time the validation loss plateaued. The total training process required ∼ 400 epochs (enabling us to reproduce the baseline performance of [65]). The results reported in Table 3 show that SE blocks consistently improve the accuracy by a large margin at a minimal increase in computational cost.

**모바일 설정**. 마지막으로, 우리는 모바일 최적화 네트워크 등급의 두 가지 대표적인 아키텍처, MobileNet [64]와 ShuffleNet [65]를 고려합니다. 이러한 실험을 위해, 우리는 [65]에서와 같이 256개의 미니 패치 크기를 사용하고 약간 덜 공격적인 데이터 확대 및 정규화를 사용했습니다. 우리는 SGD를 사용하여 8개의 GPU에 걸쳐 모델을 훈련시켰는데, 이때 모멘텀(0.9로 설정)과 0.1의 초기 학습률은 검증 손실이 안정될 때마다 10배 감소했습니다. 총 교육 프로세스는 최대 400에폭([65]의 기본 성능을 재현할 수 있도록 지원)을 필요로 했습니다. 표 3에 보고된 결과에 따르면 SE 블록은 최소한의 계산 비용 증가로 일관되게 정확도를 크게 향상시킵니다.

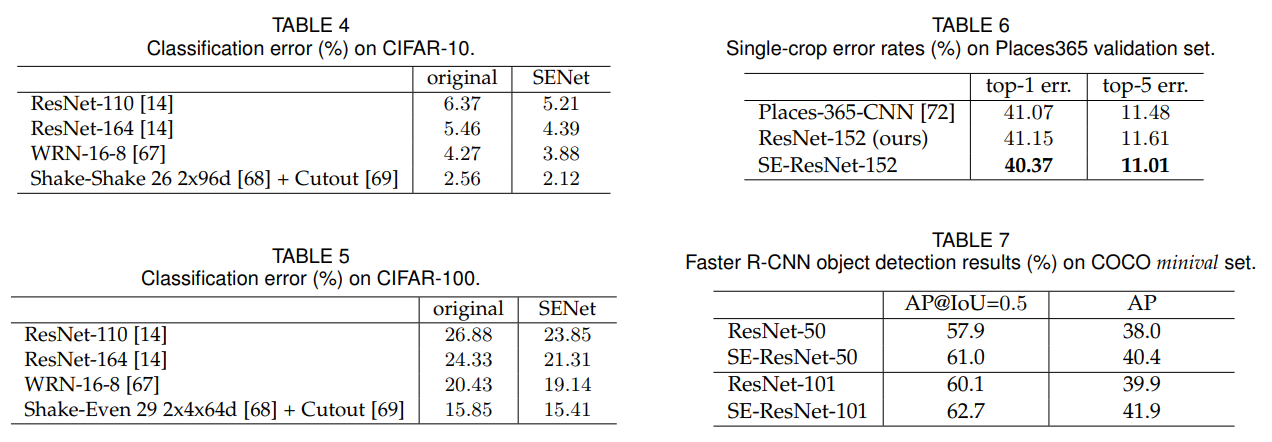


TABLE 4 Classification error (%) on CIFAR-10.

표 4 CIFAR-10의 분류 오류(%)

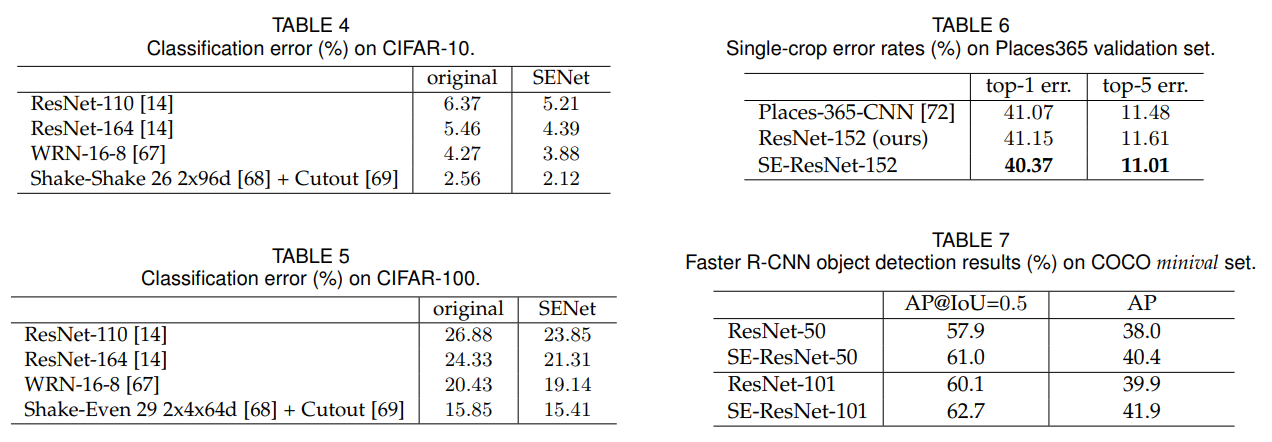


TABLE 5 Classification error (%) on CIFAR-100.

표 5 CIFAR-100의 분류 오류(%)

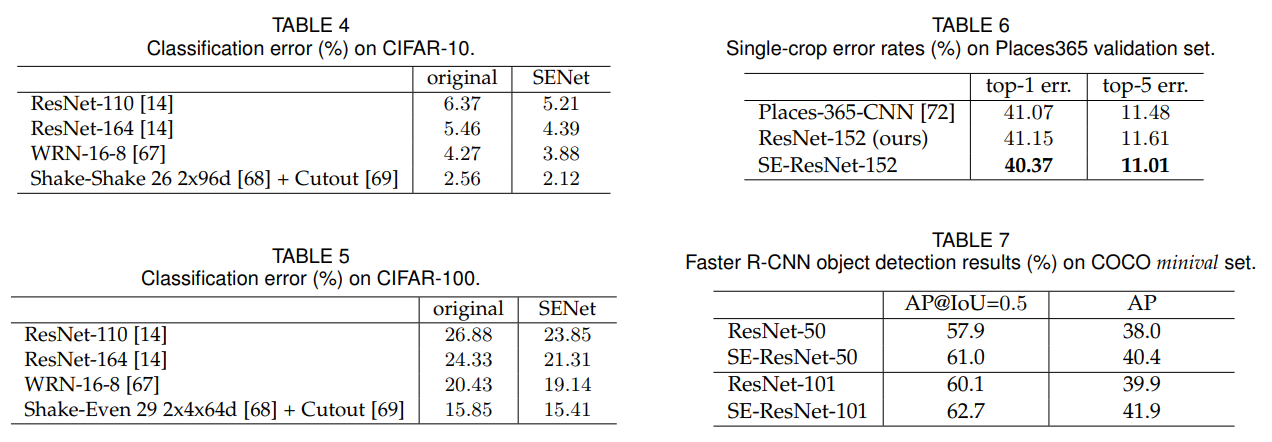


TABLE 6 Single-crop error rates (%) on Places365 validation set.

표 6 Places365 검증 세트의 싱글 크롭 오류율(%)

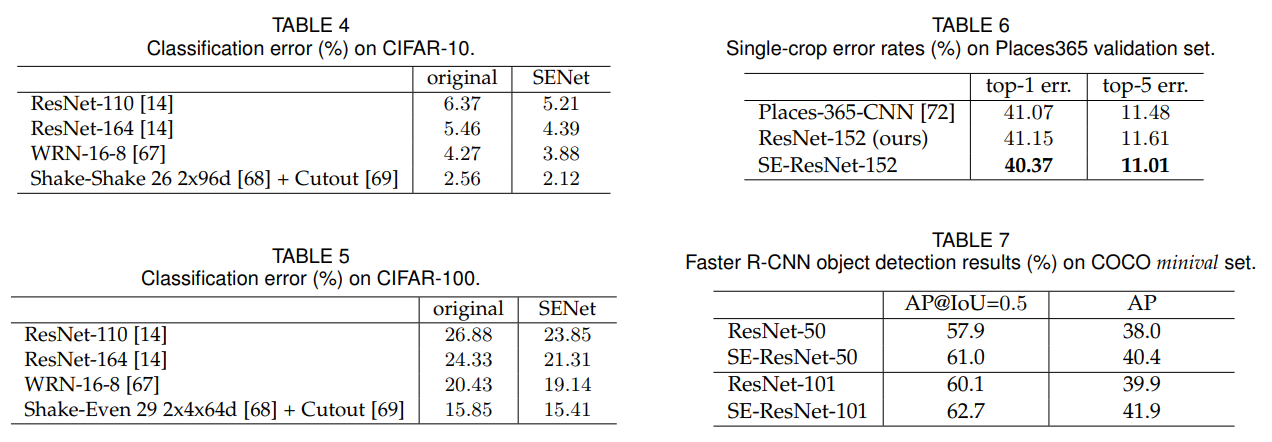


TABLE 7 Faster R-CNN object detection results (%) on COCO minival set.

표 7 COCO 미니벌 세트의 보다 빠른 R-CNN 물체 감지 결과(%)

**Additional datasets.** We next investigate whether the benefits of SE blocks generalise to datasets beyond ImageNet. We perform experiments with several popular baseline architectures and techniques (ResNet-110 [14], ResNet-164 [14], WideResNet-16-8 [67], Shake-Shake [68] and Cutout [69]) on the CIFAR-10 and CIFAR-100 datasets [70]. These comprise a collection of 50k training and 10k test 32 × 32 pixel RGB images, labelled with 10 and 100 classes respectively. The integration of SE blocks into these networks follows the same approach that was described in Section 3.3. Each baseline and its SENet counterpart are trained with standard data augmentation strategies [24], [71]. During training, images are randomly horizontally flipped and zero-padded on each side with four pixels before taking a random 32 × 32 crop. Mean and standard deviation normalisation is also applied. The setting of the training hyperparameters (e.g. minibatch size, initial learning rate, weight decay) match those suggested by the original papers. We report the performance of each baseline and its SENet counterpart on CIFAR-10 in Table 4 and performance on CIFAR-100 in Table 5. We observe that in every comparison SENets outperform the baseline architectures, suggesting that the benefits of SE blocks are not confined to the ImageNet dataset.

**추가 데이터 세트**. 다음으로 SE 블록의 이점이 ImageNet 이외의 데이터셋으로 일반화되는지 여부를 조사합니다. CIFAR-10 및 CIFAR-100 데이터셋[70]에서 몇 가지 인기 있는 기준 아키텍처 및 기법(ResNet-110 [14], ResNet-164 [14], WideResNet-16-8 [67], Shake-Shake [68] 및 Cutout [69])을 사용하여 실험을 수행합니다. 50k 훈련 및 10k 테스트 32 x 32 픽셀 RGB 이미지 모음으로 구성되며, 각각 10과 100 클래스로 라벨이 지정됩니다. SE 블록을 이러한 네트워크에 통합하는 것은 섹션 3.3에서 설명한 것과 동일한 접근방식을 따릅니다. 각 기준선과 SENet 대응물은 표준 데이터 증강 전략으로 훈련된다[24], [71]. 트레이닝중에, 화상을 랜덤으로 수평으로 플립 해, 4개의 픽셀로 양쪽에 제로 패드를 붙인 후, 랜덤 32 × 32 크롭을 촬영합니다. 평균 및 표준 편차 정규화도 적용됩니다. 훈련 하이퍼 파라미터의 설정(예: 미니배치 크기, 초기 학습 속도, 체중 감소)은 원본 논문이 제안한 것과 일치한다. 우리는 표 4의 CIFAR-10에 대한 각 기준선과 그 SENet 대응책의 성능과 표 5의 CIFAR-100에 대한 성능을 보고한다. 모든 비교에서 SEets는 기본 아키텍처를 능가하는 성능을 발휘하며, 이는 SE 블록의 이점이 ImageNet 데이터 세트에 국한되지 않음을 알 수 있습니다.

**5.2. Scene Classification**

We also conduct experiments on the Places365-Challenge dataset for scene classification. This dataset comprises 8 million training images and 36, 500 validation images across 365 categories. Relative to classification, the task of scene understanding offers an alternative assessment of a model’s ability to generalise well and handle abstraction. This is because it often requires the model to handle more complex data associations and to be robust to a greater level of appearance variation.

또한 장면 분류를 위해 Places365-Challenge 데이터 세트에 대한 실험을 실시한다. 이 데이터 세트는 365개 카테고리에 걸쳐 800만 개의 교육 이미지와 36,500개의 검증 이미지로 구성됩니다. 분류와 관련하여, 장면 이해 작업은 모델의 잘 일반화하고 추상화를 처리하는 능력에 대한 대안적 평가를 제공한다. 이는 종종 모형이 더 복잡한 데이터 연결을 처리하고 더 큰 수준의 모양 변동에 대해 견고해야 하기 때문입니다.

We opted to use ResNet-152 as a strong baseline to assess the effectiveness of SE blocks and follow the training and evaluation protocols described in [72], [74]. In these experiments, models are trained from scratch. We report the results in Table 6, comparing also with prior work. We observe that SE-ResNet-152 (11.01% top-5 error) achieves a lower validation error than ResNet-152 (11.61% top-5 error), providing evidence that SE blocks can also yield improvements for scene classification. This SENet surpasses the previous state-of-the-art model Places-365-CNN [72] which has a top-5 error of 11.48% on this task.

우리는 ResNet-152를 SE 블록의 효과를 평가하고 [72], [74]에 설명된 훈련 및 평가 프로토콜을 따르기 위한 강력한 기준선으로 사용하기로 선택했습니다. 이러한 실험에서 모델은 처음부터 훈련됩니다. 이전 작업과 비교한 결과를 표 6에 보고합니다. SE-ResNet-152(11.01% 상위 5개 오류)가 ResNet-152(11.61% 상위 5개 오류)보다 낮은 검증 오류를 달성하여 SE 블록이 장면 분류에도 개선을 가져올 수 있다는 증거를 제공하는 것을 관찰했다. 이 SENet은 이 작업에서 상위 5개 오류가 11.48%인 이전 최신 모델 Places-365-CNN[72]을 능가한다.

**5.3. Object Detection on COCO**

We further assess the generalisation of SE blocks on the task of object detection using the COCO dataset. As in previous work, we use the minival protocol, i.e., training the models on the union of the 80k training set and a 35k val subset and evaluating on the remaining 5k val subset. Weights are initialised by the parameters of the model trained on the ImageNet dataset. We use the Faster R-CNN detection framework as the basis for evaluating our models and follow the hyperparameter setting described in [76] (i.e., end-to-end training with the ’2x’ learning schedule). Our goal is to evaluate the effect of replacing the trunk architecture (ResNet) in the object detector with SE-ResNet, so that any changes in performance can be attributed to better representations. Table 7 reports the validation set performance of the object detector using ResNet-50, ResNet-101 and their SE counterparts as trunk architectures. SE-ResNet-50 outperforms ResNet-50 by 2.4% (a relative 6.3% improvement) on COCO’s standard AP metric and by 3.1% on AP@IoU=0.5. SE blocks also benefit the deeper ResNet-101 architecture achieving a 2.0% improvement (5.0% relative improvement) on the AP metric. In summary, this set of experiments demonstrate the generalisability of SE blocks. The induced improvements can be realised across a broad range of architectures, tasks and datasets.

COCO 데이터 세트를 사용하여 물체 감지 작업에서 SE 블록의 일반화를 추가로 평가한다. 이전 작업과 마찬가지로 미니벌 프로토콜을 사용한다. 즉, 80k 훈련 세트와 35k val subset의 결합에 대해 모델을 교육하고 나머지 5k val subset에 대해 평가한다. 가중치는 ImageNet 데이터 세트에 대해 훈련받은 모델의 파라미터에 의해 초기화됩니다. 모델을 평가하기 위한 기준으로 고속 R-CNN 감지 프레임워크를 사용하고 [76]에 설명된 하이퍼 매개 변수 설정(즉, '2x' 학습 일정의 엔드 투 엔드 교육)을 따른다. 우리의 목표는 객체 디텍터의 트렁크아키텍처(ResNet)를 SE-ResNet으로 대체하는 효과를 평가하여 성능의 변화가 더 나은 표현에 기인하도록 하는 것입니다. 표 7은 ResNet-50, ResNet-101 및 그 SE를 트렁크아키텍처로서 사용하는 오브젝트 디텍터의 검증 세트의 퍼포먼스를 나타내고 있습니다. SE-ResNet-50은 COCO의 표준 AP 메트릭에서 ResNet-50보다 2.4%(상대적인 6.3% 향상), AP@IoU=0.5에서 3.1% 향상됩니다. 또한 SE 블록은 더 심층적인 ResNet-101 아키텍처가 AP 메트릭에서 2.0%(상대적인 5.0%) 향상 효과를 발휘합니다. 요약하면, 이 실험 집합은 SE 블럭의 일반화 가능성을 입증합니다. 이러한 개선은 광범위한 아키텍처, 작업 및 데이터셋에 걸쳐 실현될 수 있습니다.

**5.4. ILSVRC 2017 Classification Competition**

SENets formed the foundation of our submission to the ILSVRC competition where we achieved first place. Our winning entry comprised a small ensemble of SENets that employed a standard multi-scale and multi-crop fusion strategy to obtain a top-5 error of 2.251% on the test set. As part of this submission, we constructed an additional model, SENet-154, by integrating SE blocks with a modified ResNeXt [19] (the details of the architecture are provided in Appendix). We compare this model with prior work on the ImageNet validation set in Table 8 using standard crop sizes (224 × 224 and 320 × 320). We observe that SENet-154 achieves a top-1 error of 18.68% and a top-5 error of 4.47% using a 224 × 224 centre crop evaluation, which represents the strongest reported result.

SENets는 ILSVRC 경쟁에의 참가의 기초를 형성해, 1위를 달성했습니다. 수상작은 표준 멀티스케일 및 멀티크롭 퓨전 전략을 채택하여 테스트 세트에서 상위 5개 오류 2.251%를 달성한 SENets의 소규모 앙상블이었습니다. 이 제출의 일환으로 SE 블록을 수정된 ResNeXt [19]와 통합하여 추가 모델인 SEet-154를 구축했습니다(아키텍처의 자세한 내용은 부록에 기재되어 있습니다). 이 모델을 표준 크롭 크기(224 × 224 및 320 × 320)를 사용하여 표 8의 ImageNet 검증 세트에 대한 이전 작업과 비교합니다. SENet-154는 보고된 가장 강력한 결과를 나타내는 224 × 224 중심 작물 평가를 사용하여 상위 1의 오차 18.68%와 상위 5의 오차 4.47%를 달성한 것을 관찰했다.

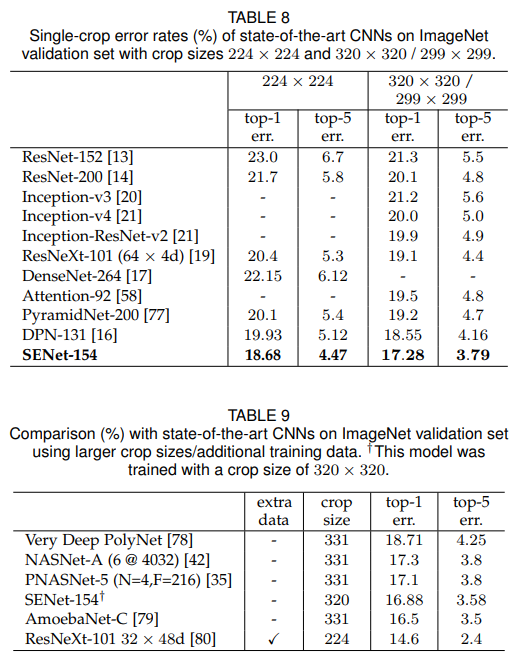


TABLE 8 Single-crop error rates (%) of state-of-the-art CNNs on ImageNet validation set with crop sizes 224 × 224 and 320 × 320 / 299 × 299.

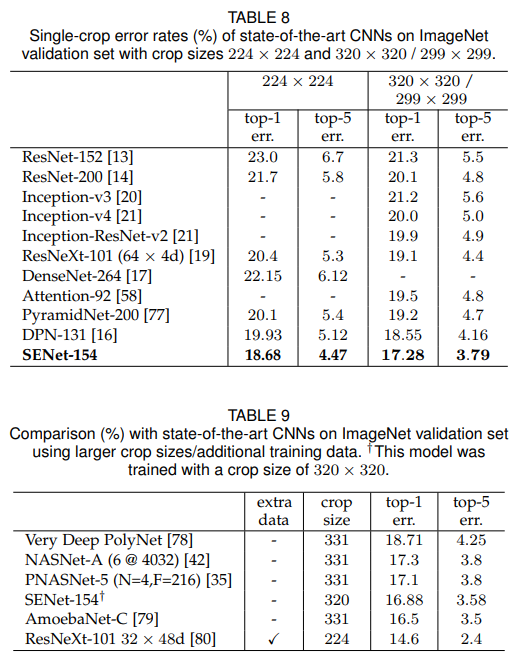
표 8 크롭 크기 224 × 224 및 320 × 320 / 299 × 299의 ImageNet 검증 세트에 대한 최첨단 CNN의 단일 크롭 오류율(%)

TABLE 9 Comparison (%) with state-of-the-art CNNs on ImageNet validation set using larger crop sizes/additional training data. †This model was trained with a crop size of 320 × 320.

표 9 더 큰 크롭 크기/추가 훈련 데이터를 사용한 ImageNet 검증 세트에 대한 최신 CNN과의 비교(%) ②이 모델은 320×320의 크롭 사이즈로 트레이닝 되었습니다.

Following the challenge there has been a great deal of further progress on the ImageNet benchmark. For comparison, we include the strongest results that we are currently aware of in Table 9. The best performance using only ImageNet data was recently reported by [79]. This method uses reinforcement learning to develop new policies for data augmentation during training to improve the performance of the architecture searched by [31]. The best overall performance was reported by [80] using a ResNeXt-101 32×48d architecture. This was achieved by pretraining their model on approximately one billion weakly labelled images and finetuning on ImageNet. The improvements yielded by more sophisticated data augmentation and extensive pretraining may be complementary to our proposed changes to the network architecture.

이 과제에 수반해, ImageNet 벤치마크에 큰 진전이 있었습니다. 비교를 위해 현재 알고 있는 가장 강력한 결과를 표 9에 포함시켰습니다. 최근 [79]에 의해 ImageNet 데이터만을 사용한 최고의 성능이 보고되었습니다. 이 방법은 [31]에 의해 검색된 아키텍처의 성능을 개선하기 위해 훈련 중에 데이터 증강을 위한 새로운 정책을 개발하기 위해 강화 학습을 사용합니다. ResNeXt-101 32×48d 아키텍처를 사용한 [80]은(는) 최고의 전체 성능을 보고했습니다. 이는 약 10억 개의 약하게 라벨링된 이미지에 대한 모델을 사전 교육하고 ImageNet에서 미세 조정함으로써 달성되었습니다. 보다 정교한 데이터 증강과 광범위한 사전 훈련에 의해 도출된 개선 사항은 네트워크 아키텍처에 제안된 변경 사항을 보완할 수 있습니다.

**6. Ablation Study**

In this section we conduct ablation experiments to gain a better understanding of the effect of using different configurations on components of the SE blocks. All ablation experiments are performed on the ImageNet dataset on a single machine (with 8 GPUs). ResNet-50 is used as the backbone architecture. We found empirically that on ResNet architectures, removing the biases of the FC layers in the excitation operation facilitates the modelling of channel dependencies, and use this configuration in the following experiments. The data augmentation strategy follows the approach described in Section 5.1. To allow us to study the upper limit of performance for each variant, the learning rate is initialised to 0.1 and training continues until the validation loss plateaus2 (∼300 epochs in total). The learning rate is then reduced by a factor of 10 and then this process is repeated (three times in total). Label-smoothing regularisation is used during training

이 단원에서는 SE 블록의 구성요소에 대한 다양한 구성 사용의 영향을 더 잘 이해하기 위해 절제 실험을 수행합니다. 모든 절제 실험은 단일 머신(GPU 8개)의 ImageNet 데이터 세트에 대해 수행됩니다. ResNet-50은 백본 아키텍처로 사용됩니다. 우리는 ResNet 아키텍처에서, 여기 동작에서 FC 계층의 편견을 제거하는 것이 채널 의존성의 모델링을 용이하게 한다는 것을 경험적으로 발견했고, 다음 실험에서 이 구성을 사용합니다. 데이터 증강 전략은 섹션 5.1에 설명된 접근방식을 따른다. 각 변형에 대한 성능 상한을 연구할 수 있도록 학습 속도를 0.1로 초기화하고 검증 손실 평판2(총 300에폭)까지 훈련을 계속한다. 그런 다음 학습률이 10배 감소되고 이 과정이 반복됩니다(총 3회). 라벨 스무딩 정규화는 교육 중에 사용됩니다.

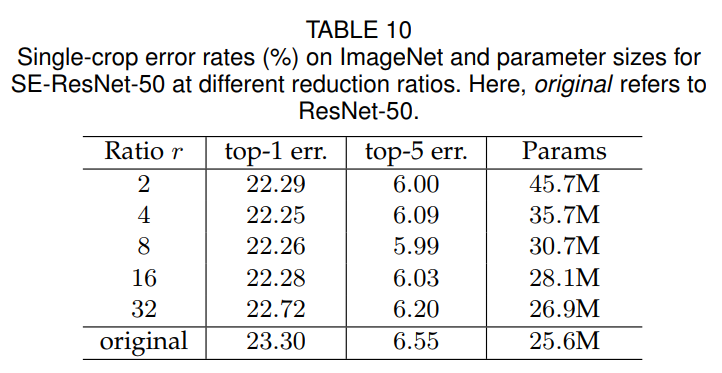


TABLE 10 Single-crop error rates (%) on ImageNet and parameter sizes for SE-ResNet-50 at different reduction ratios. Here, original refers to ResNet-50.

표 10 ImageNet의 싱글 크롭 에러율(%)과 SE-ResNet-50의 파라미터 크기(감소율이 다름) 여기서 original은 ResNet-50을 나타냅니다.

**6.1. Reduction ratio**

The reduction ratio r introduced in Eqn. 5 is a hyperparameter which allows us to vary the capacity and computational cost of the SE blocks in the network. To investigate the trade-off between performance and computational cost mediated by this hyperparameter, we conduct experiments with SE-ResNet-50 for a range of different r values. The comparison in Table 10 shows that performance is robust to a range of reduction ratios. Increased complexity does not improve performance monotonically while a smaller ratio dramatically increases the parameter size of the model. Setting r = 16 achieves a good balance between accuracy and complexity. In practice, using an identical ratio throughout a network may not be optimal (due to the distinct roles performed by different layers), so further improvements may be achievable by tuning the ratios to meet the needs of a given base architecture.

Eqn. 5에서 도입된 감소율 r은 네트워크 내 SE 블록의 용량과 계산 비용을 변화시킬 수 있는 하이퍼 파라미터입니다. 이 하이퍼 매개 변수에 의해 매개되는 성능과 계산 비용 사이의 균형을 조사하기 위해, 우리는 다양한 r 값의 범위에 대해 SE-ResNet-50으로 실험을 수행한다. 표 10의 비교는 다양한 감소율에 대해 성능이 강하다는 것을 보여준다. 복잡성이 증가한다고 해서 성능이 단조롭게 향상되지는 않지만 비율이 작을수록 모델의 매개변수 크기가 크게 증가합니다. r = 16으로 설정하면 정확도와 복잡도 사이의 균형이 잘 잡힙니다. 실제로는 네트워크 전체에서 동일한 비율을 사용하는 것이 최적이라고는 할 수 없습니다(각 계층이 수행하는 역할이 다르기 때문에).따라서 비율을 조정하여 특정 기본 아키텍처의 요구를 충족시킴으로써 더욱 개선할 수 있습니다.

**6.2. Squeeze Operator**

We examine the significance of using global average pooling as opposed to global max pooling as our choice of squeeze operator (since this worked well, we did not consider more sophisticated alternatives). The results are reported in Table 11. While both max and average pooling are effective, average pooling achieves slightly better performance, justifying its selection as the basis of the squeeze operation. However, we note that the performance of SE blocks is fairly robust to the choice of specific aggregation operator.

우리는 스퀴즈 연산자의 선택으로 글로벌 최대 풀링과 달리 글로벌 평균 풀링을 사용하는 것의 중요성을 검토한다(이것이 잘 작동했기 때문에 우리는 더 정교한 대안을 고려하지 않았다). 결과는 표 11에 보고된다. 최대 풀링과 평균 풀링 모두 효과적이지만 평균 풀링은 약간 더 나은 성능을 달성하므로 스퀴즈 작업의 기준으로 선택한 것이 정당합니다. 그러나 SE 블록의 성능은 특정 집계 운영자의 선택에 상당히 강력합니다.

**6.3. Excitation Operator**

We next assess the choice of non-linearity for the excitation mechanism. We consider two further options: ReLU and tanh, and experiment with replacing the sigmoid with these alternative non-linearities. The results are reported in Table 12. We see that exchanging the sigmoid for tanh slightly worsens performance, while using ReLU is dramatically worse and in fact causes the performance of SE-ResNet-50 to drop below that of the ResNet-50 baseline. This suggests that for the SE block to be effective, careful construction of the excitation operator is important.

다음으로 들뜸 메커니즘에 대한 비선형성 선택을 평가한다. 다음 두 가지 옵션을 고려해야 합니다. ReLU와 tanh를 사용하여 Sigmoid를 이러한 대체 비선형성으로 대체하는 실험을 합니다. 결과는 표 12에 기재되어 있다. Sigmoid를 tanh로 교환하면 퍼포먼스가 약간 저하되지만 ReLU를 사용하면 퍼포먼스가 현저하게 저하되어 실제로는 SE-ResNet-50의 퍼포먼스가 ResNet-50 베이스라인보다 저하되는 것을 알 수 있습니다. 이는 SE 블럭이 효과적이기 위해서는 여기 연산자를 신중하게 구성하는 것이 중요하다는 것을 의미합니다.

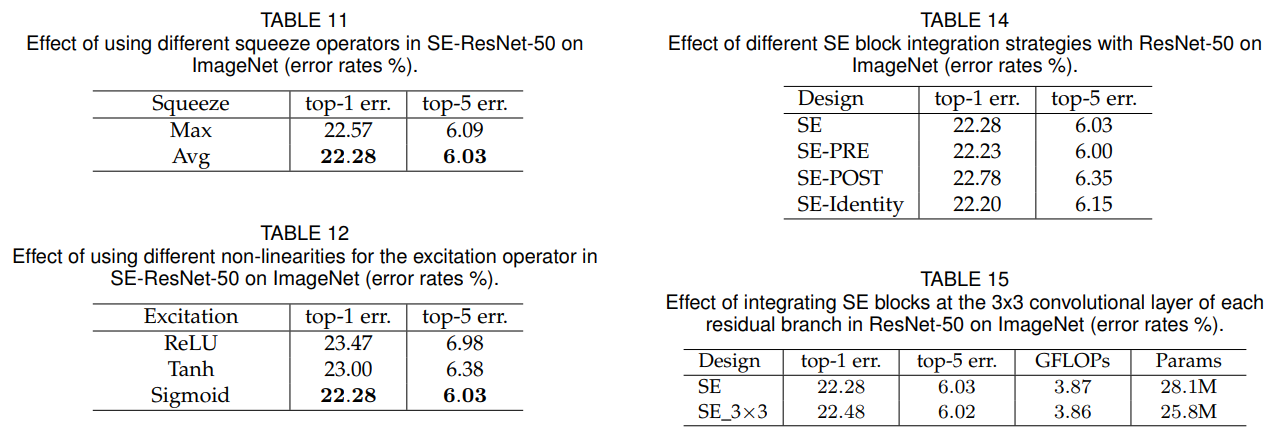


TABLE 11 Effect of using different squeeze operators in SE-ResNet-50 on ImageNet (error rates %).

표 11 SE-ResNet-50에서 다른 스퀴즈 연산자를 사용한 ImageNet의 효과(오류율 %)

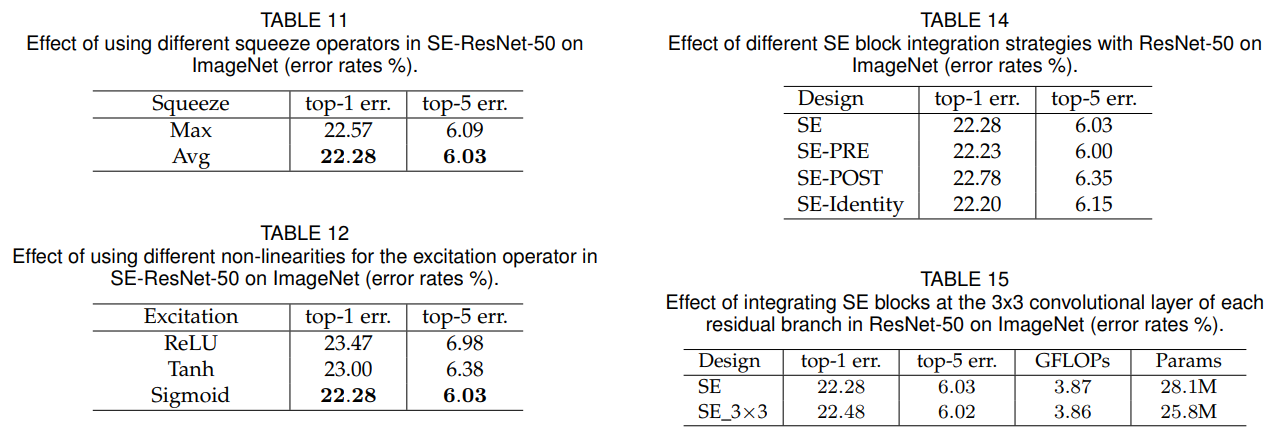


TABLE 12 Effect of using different non-linearities for the excitation operator in SE-ResNet-50 on ImageNet (error rates %).

표 12 SE-ResNet-50의 들뜸 연산자에 대해 ImageNet에서 다른 비선형성 사용의 효과(오류율 %)

**6.4. Different Stages**

We explore the influence of SE blocks at different stages by integrating SE blocks into ResNet-50, one stage at a time. Specifically, we add SE blocks to the intermediate stages: stage 2, stage 3 and stage 4, and report the results in Table 13. We observe that SE blocks bring performance benefits when introduced at each of these stages of the architecture. Moreover, the gains induced by SE blocks at different stages are complementary, in the sense that they can be combined effectively to further bolster network performance.

SE 블록을 ResNet-50에 한 번에 한 단계씩 통합하여 다양한 단계에서 SE 블록의 영향을 살펴봅니다. 구체적으로는 중간 단계인 2단계, 3단계, 4단계에 SE 블록을 추가하고 그 결과를 표 13에 보고한다. SE 블록은 아키텍처의 각 단계에서 도입될 때 성능상의 이점을 가져온다는 것을 알 수 있습니다. 더욱이, 다양한 단계에서 SE 블록에 의해 유도되는 이득은 네트워크 성능을 더욱 강화하기 위해 효과적으로 결합될 수 있다는 점에서 상호 보완적입니다.

**6.5. Integration strategy**

Finally, we perform an ablation study to assess the influence of the location of the SE block when integrating it into existing architectures. In addition to the proposed SE design, we consider three variants: (1) SE-PRE block, in which the SE block is moved before the residual unit; (2) SE-POST block, in which the SE unit is moved after the summation with the identity branch (after ReLU) and (3) SE-Identity block, in which the SE unit is placed on the identity connection in parallel to the residual unit. These variants are illustrated in Figure 5 and the performance of each variant is reported in Table 14. We observe that the SE-PRE, SE-Identity and proposed SE block each perform similarly well, while usage of the SE-POST block leads to a drop in performance. This experiment suggests that the performance improvements produced by SE units are fairly robust to their location, provided that they are applied prior to branch aggregation.

마지막으로 기존 아키텍처에 통합할 때 SE 블록의 위치의 영향을 평가하기 위해 절제 연구를 수행합니다. 제안된 SE 설계 외에, 우리는 세 가지 변형을 고려한다. (1) SE 블록이 잔존 유닛보다 먼저 이동되는 SE-PRE 블록, (2) SE 유닛이 (ReLU 이후) ID 브랜치와의 합산 후에 이동되는 SE-POST 블록, (3) SE 유닛이 ID에 평행하게 연결되는 SE-Identity 블록. 잔여 단위 이러한 변형은 그림 5에 나타나 있으며, 각 변형들의 성능은 표 14에 보고되어 있다. SE-PRE, SE-Identity 및 제안된 SE 블록은 각각 비슷한 성능을 발휘하는 반면 SE-POST 블록을 사용하면 성능이 저하되는 것으로 나타났습니다. 이 실험은 SE 유닛에 의해 생성된 성능 개선이 지점 집계에 앞서 적용된다는 전제 하에 해당 위치에 상당히 견고하다는 것을 시사한다.

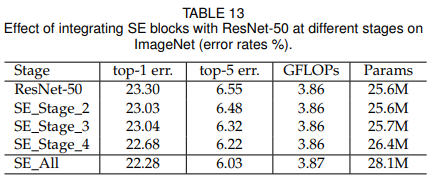


TABLE 13 Effect of integrating SE blocks with ResNet-50 at different stages on ImageNet (error rates %).

표 13 SE 블록과 ResNet-50을 ImageNet 상에서의 다양한 단계에서 통합하는 효과(오류율 %)

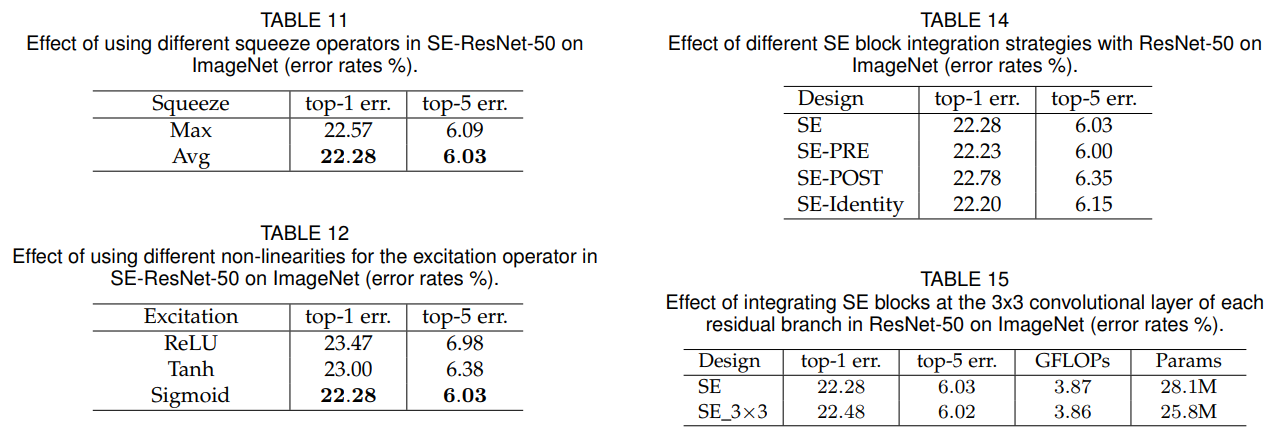


TABLE 14 Effect of different SE block integration strategies with ResNet-50 on ImageNet (error rates %).

표 14 ResNet-50을 사용한 다양한 SE 블록 통합 전략이 ImageNet에 미치는 영향(오류율 %)

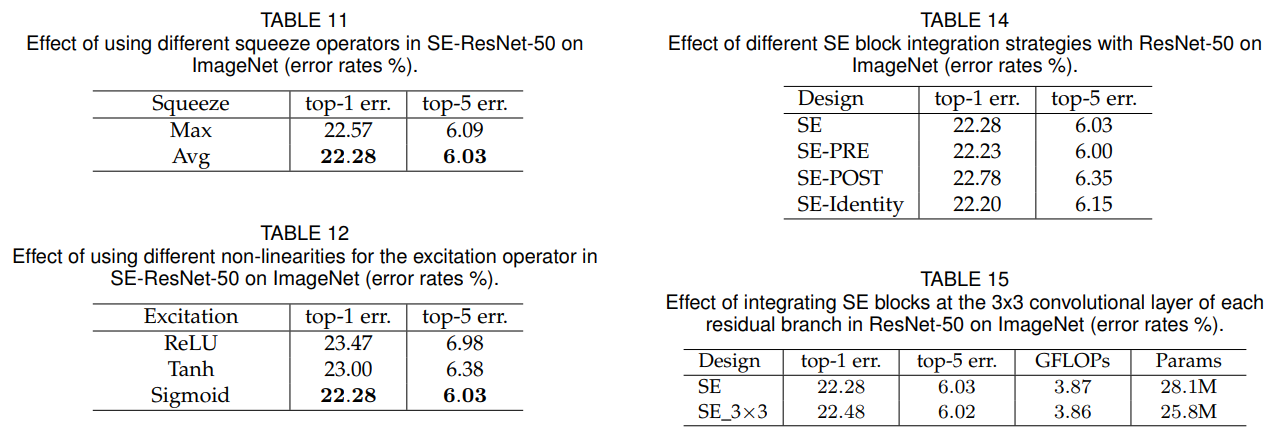


TABLE 15 Effect of integrating SE blocks at the 3x3 convolutional layer of each residual branch in ResNet-50 on ImageNet (error rates %).

표 15 ResNet-50의 각 잔여 분기의 3x3 컨볼루션 층에서 SE 블록을 통합하는 효과(오류율 %)

In the experiments above, each SE block was placed outside the structure of a residual unit. We also construct a variant of the design which moves the SE block inside the residual unit, placing it directly after the 3 × 3 convolutional layer. Since the 3 × 3 convolutional layer possesses fewer channels, the number of parameters introduced by the corresponding SE block is also reduced. The comparison in Table 15 shows that the SE 3×3 variant achieves comparable classification accuracy with fewer parameters than the standard SE block. Although it is beyond the scope of this work, we anticipate that further efficiency gains will be achievable by tailoring SE block usage for specific architectures.

위의 실험에서 각 SE 블록은 잔류 유닛의 구조 외부에 배치되었습니다. 우리는 또한 SE 블록을 3 × 3 컨볼루션 레이어 바로 뒤에 배치하여 잔류 유닛 내부로 이동하는 설계의 변형을 구성한다. 3 × 3 컨볼루션 층은 더 적은 채널을 소유하기 때문에 대응하는 SE 블록에 의해 도입되는 파라미터의 수도 감소합니다. 표 15의 비교 결과 SE 3×3 변종은 표준 SE 블록보다 적은 매개변수로 유사한 분류 정확도를 달성했습니다. 이 작업의 범위를 벗어나지만 특정 아키텍처에 맞게 SE 블록 사용을 맞춤화함으로써 추가적인 효율성 향상을 달성할 수 있을 것으로 예상됩니다.

**7. Role of SE-Blocks**

Although the proposed SE block has been shown to improve network performance on multiple visual tasks, we would also like to understand the relative importance of the squeeze operation and how the excitation mechanism operates in practice. A rigorous theoretical analysis of the representations learned by deep neural networks remains challenging, we therefore take an empirical approach to examining the role played by the SE block with the goal of attaining at least a primitive understanding of its practical function.

제안된 SE 블록은 여러 시각적 작업에서 네트워크 성능을 향상시키는 것으로 나타났지만, 스퀴즈 작동의 상대적 중요성과 들뜸 메커니즘이 실제로 어떻게 작동하는지도 이해하고 싶습니다. 심층 신경 네트워크에서 학습한 표현에 대한 엄밀한 이론적 분석은 여전히 어려우며, 따라서 우리는 적어도 그 실제 기능에 대한 원시적인 이해를 얻는 것을 목표로 SE 블록이 수행하는 역할을 검토하는 경험적 접근법을 취한다.

**7.1. Effect of Squeeze**

To assess whether the global embedding produced by the squeeze operation plays an important role in performance, we experiment with a variant of the SE block that adds an equal number of parameters, but does not perform global average pooling. Specifically, we remove the pooling operation and replace the two FC layers with corresponding 1 × 1 convolutions with identical channel dimensions in the excitation operator, namely NoSqueeze, where the excitation output maintains the spatial dimensions as input. In contrast to the SE block, these point-wise convolutions can only remap the channels as a function of the output of a local operator. While in practice, the later layers of a deep network will typically possess a (theoretical) global receptive field, global embeddings are no longer directly accessible throughout the network in the NoSqueeze variant. The accuracy and computational complexity of both models are compared to a standard ResNet-50 model in Table 16. We observe that the use of global information has a significant influence on the model performance, underlining the importance of the squeeze operation. Moreover, in comparison to the NoSqueeze design, the SE block allows this global information to be used in a computationally parsimonious manner.

스퀴즈 연산에 의해 생성된 글로벌 임베딩이 성능에 중요한 역할을 하는지 평가하기 위해 동일한 수의 매개변수를 추가하지만 글로벌 평균 풀링을 수행하지 않는 변형 SE 블록으로 실험한다. 구체적으로 풀링 작업을 제거하고 두 FC 레이어를 여자 연산자의 채널 치수가 동일한 1 × 1 컨볼루션, 즉 여자 출력이 입력으로 공간 치수를 유지하는 NoSqueeze로 교체한다. SE 블록과 달리 이러한 포인트별 컨볼루션은 로컬 운영자의 출력 함수로서만 채널을 재매핑할 수 있습니다. 실제로 딥 네트워크의 후층에는 일반적으로 (이론적인) 글로벌 수용 필드가 포함되어 있지만 NoSqueze 변종에서는 글로벌 임베딩이 네트워크 전체에서 직접 액세스 할 수 없게 됩니다. 두 모델의 정확성과 계산 복잡성은 표 16의 표준 ResNet-50 모델과 비교됩니다. 글로벌 정보의 사용이 모델 성능에 큰 영향을 미쳐 스퀴즈 작업의 중요성을 강조합니다. 또한 NoSqueise 설계와 비교하여 SE 블록은 이러한 글로벌 정보를 계산적으로 절약하여 사용할 수 있습니다.

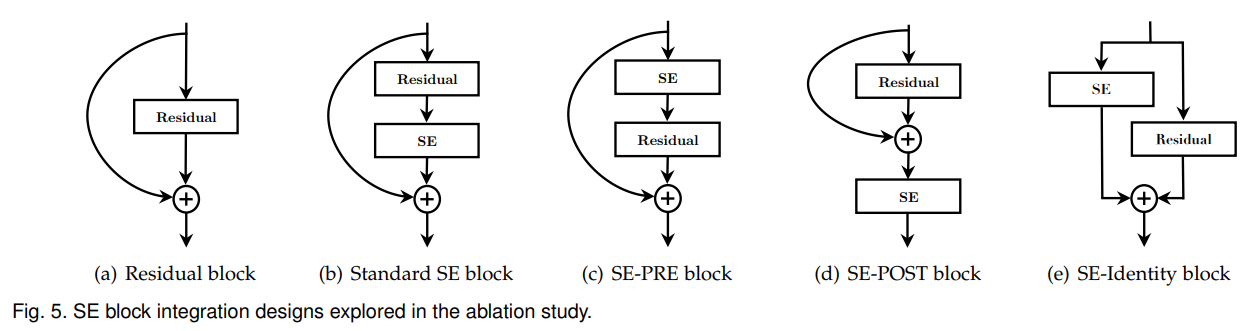


Fig. 5. SE block integration designs explored in the ablation study.

그림 5. 절제 연구에서 탐색한 SE 블록 통합 설계.

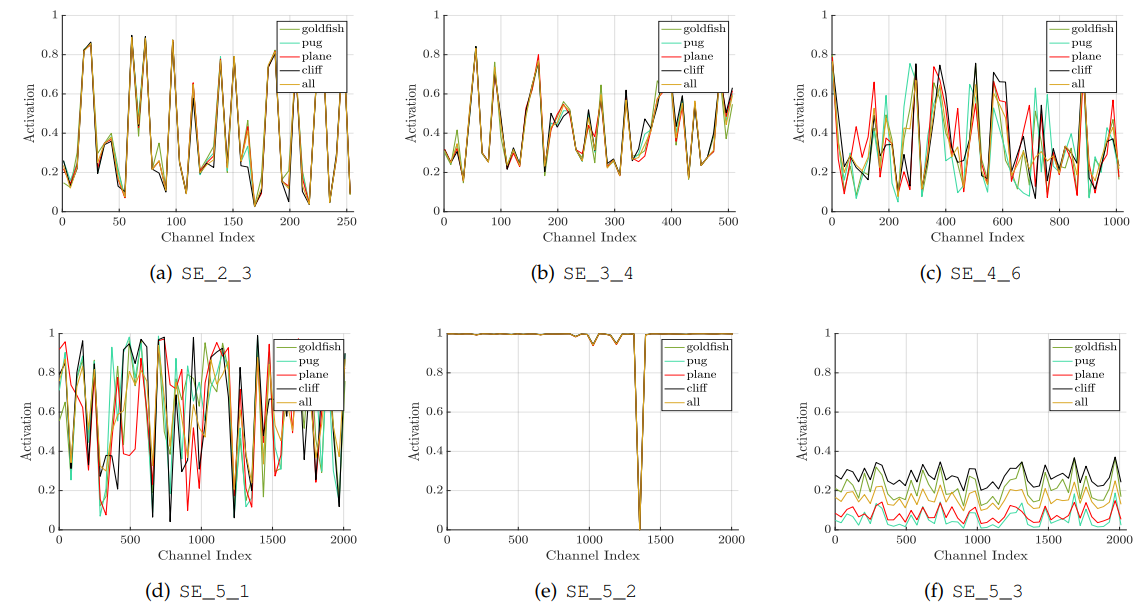


Fig. 6. Activations induced by the Excitation operator at different depths in the SE-ResNet-50 on ImageNet. Each set of activations is named according to the following scheme: SE\_stageID\_blockID. With the exception of the unusual behaviour at SE\_5\_2, the activations become increasingly class-specific with increasing depth.

그림 6. ImageNet의 SE-ResNet-50의 다른 깊이에서 Excitation 오퍼레이터에 의해 유도되는 활성화. 각 액티베이션세트는 다음 스킴에 따라 이름이 지정됩니다.SE\_stageID\_블록ID. SE\_5\_2에서의 비정상적인 동작을 제외하고, 활성화는 깊이가 증가함에 따라 점점 더 세분화된다.

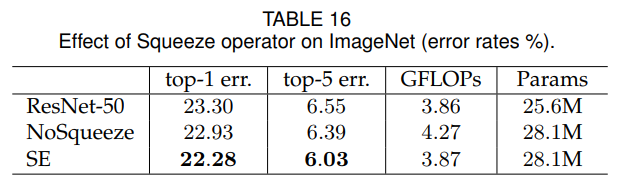


TABLE 16 Effect of Squeeze operator on ImageNet (error rates %).

표 16 ImageNet에서의 Squeeze 연산자의 효과(오류율 %)

**7.2. Role of Excitation**

To provide a clearer picture of the function of the excitation operator in SE blocks, in this section we study example activations from the SE-ResNet-50 model and examine their distribution with respect to different classes and different input images at various depths in the network. In particular, we would like to understand how excitations vary across images of different classes, and across images within a class.

SE 블록에서의 여기 연산자의 기능을 보다 명확하게 파악하기 위해, 이 섹션에서는 SE-ResNet-50 모델의 활성화 예를 연구하여 네트워크의 다양한 깊이에서 다양한 클래스 및 다른 입력 이미지에 대한 분포를 조사한다. 특히, 들뜸이 다른 클래스의 이미지와 클래스 내의 이미지 간에 어떻게 다른지 알고 싶습니다.

We first consider the distribution of excitations for different classes. Specifically, we sample four classes from the ImageNet dataset that exhibit semantic and appearance diversity, namely goldfish, pug, plane and cliff (example images from these classes are shown in Appendix). We then draw fifty samples for each class from the validation set and compute the average activations for fifty uniformly sampled channels in the last SE block of each stage (immediately prior to downsampling) and plot their distribution in Fig. 6. For reference, we also plot the distribution of the mean activations across all of the 1000 classes.

우리는 먼저 다른 클래스에 대한 들뜸의 분포를 고려합니다. 구체적으로는 의미와 외관의 다양성을 나타내는 4개의 클래스, 즉 금붕어, 퍼그, 평면 및 절벽(이러한 클래스의 예시는 부록에 나타나 있다)을 샘플로 한다. 그런 다음 검증 세트에서 각 등급에 대해 50개의 표본을 추출하고 각 단계의 마지막 SE 블록(다운샘플링 직전)에서 균일하게 샘플링된 50개의 채널에 대한 평균 활성화를 계산하고, 그 분포를 그림 6에 그린다. 참고로 1000개 등급 전체에 걸친 평균 활성화의 분포를 그림으로 표시하기도 한다.

We make the following three observations about the role of the excitation operation. First, the distribution across different classes is very similar at the earlier layers of the network, e.g. SE 2 3. This suggests that the importance of feature channels is likely to be shared by different classes in the early stages. The second observation is that at greater depth, the value of each channel becomes much more class-specific as different classes exhibit different preferences to the discriminative value of features, e.g. SE 4 6 and SE 5 1. These observations are consistent with findings in previous work, namely that earlier layer features are typically more general (e.g. class agnostic in the context of the classification task) while later layer features exhibit greater levels of specificity.

여기 조작의 역할에 대해서, 다음의 3개의 관찰을 실시합니다. 첫째, 서로 다른 클래스에 걸친 분포는 네트워크의 초기 계층들, 예를 들어 SE 2 3에서 매우 유사합니다. 이는 피처 채널의 중요성이 초기 단계에서 여러 클래스에서 공유될 가능성이 높다는 것을 나타냅니다. 두 번째 관찰에서는 SE 4 6 및 SE 5 1과 같이 서로 다른 클래스가 기능의 차별적 값에 대해 서로 다른 선호도를 나타냄에 따라 각 채널의 값이 훨씬 더 클래스 고유해집니다. 이러한 관찰 결과는 이전 연구에서 발견한 것과 일치한다. 즉, 초기 계층 특성은 일반적으로 더 일반적이며(예: 분류 작업의 맥락에서 등급 불가지론적) 후 계층 특성은 더 높은 수준의 특수성을 나타낸다.

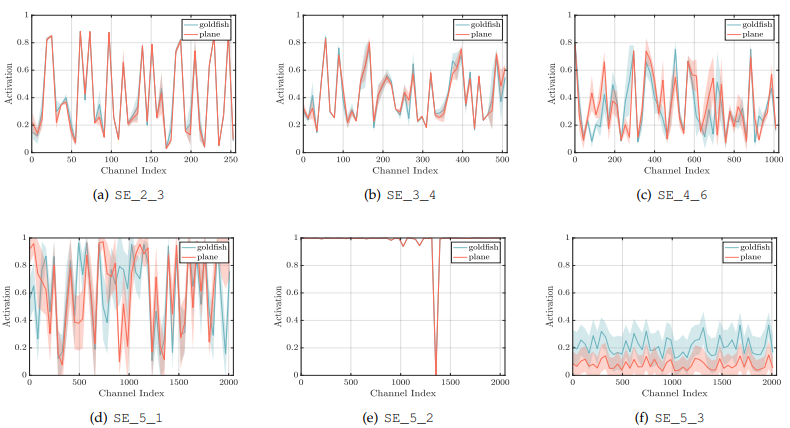


Fig. 7. Activations induced by Excitation in the different modules of SE-ResNet-50 on image samples from the goldfish and plane classes of ImageNet. The module is named “SE\_stageID\_blockID".

그림 7. SE-ResNet-50의 다른 모듈에서 ImageNet의 금붕어 및 플레인 클래스의 이미지 샘플에 대한 들뜸에 의해 유발되는 활성화. 모듈의 이름은 "SE\_stage"입니다.ID\_블록아이디」.

Next, we observe a somewhat different phenomena in the last stage of the network. SE 5 2 exhibits an interesting tendency towards a saturated state in which most of the activations are close to one. At the point at which all activations take the value one, an SE block reduces to the identity operator. At the end of the network in the SE 5 3 (which is immediately followed by global pooling prior before classifiers), a similar pattern emerges over different classes, up to a modest change in scale (which could be tuned by the classifiers). This suggests that SE 5 2 and SE 5 3 are less important than previous blocks in providing recalibration to the network. This finding is consistent with the result of the empirical investigation in Section 4 which demonstrated that the additional parameter count could be significantly reduced by removing the SE blocks for the last stage with only a marginal loss of performance.

다음으로 네트워크의 마지막 단계에서 다소 다른 현상을 관찰합니다. SE 5 2는 대부분의 활성화가 1에 가까운 포화 상태에 대한 흥미로운 경향을 보입니다. 모든 액티베이션이 값 1을 갖는 시점에서 SE 블록은 ID 연산자로 감소합니다. SE 5 3(분급기 전에 글로벌 풀링 직후)의 네트워크 끝에서 유사한 패턴이 여러 클래스에 걸쳐 나타나며, 규모에서 약간의 변화(분급기에 의해 조정될 수 있음)까지 나타납니다. 이는 SE 5 2 및 SE 5 3이 네트워크에 재보정을 제공하는 데 있어 이전의 블록보다 덜 중요하다는 것을 나타냅니다. 이 결과는 성능의 한계 손실만 있는 마지막 단계의 SE 블록을 제거함으로써 추가 매개변수 카운트를 상당히 줄일 수 있다는 것을 입증한 섹션 4의 경험적 조사 결과와 일치한다.

Finally, we show the mean and standard deviations of the activations for image instances within the same class for two sample classes (goldfish and plane) in Fig. 7. We observe a trend consistent with the inter-class visualisation, indicating that the dynamic behaviour of SE blocks varies over both classes and instances within a class. Particularly in the later layers of the network where there is considerable diversity of representation within a single class, the network learns to take advantage of feature recalibration to improve its discriminative performance. In summary, SE blocks produce instance-specific responses which nevertheless function to support the increasingly class-specific needs of the model at different layers in the architecture.

마지막으로 그림 7의 두 가지 샘플 클래스(골드피쉬 및 평면)에 대해 동일한 클래스 내의 이미지 인스턴스에 대한 활성화의 평균 및 표준 편차를 보여준다. 클래스 간 시각화와 일치하는 추세를 관찰하여 SE 블록의 동적 동작이 클래스 내 및 인스턴스 모두에 걸쳐 변화함을 나타낸다. 특히 단일 클래스 내에서 상당히 다양한 표현이 존재하는 네트워크의 후기 계층에서 네트워크는 차별적인 성능을 개선하기 위해 기능 재보정을 이용하는 방법을 학습합니다. 요약하자면, SE 블록은 인스턴스별 응답을 생성하며, 그럼에도 불구하고 아키텍처의 다른 계층에서 점점 더 클래스별 요구 사항을 지원하는 기능을 합니다.

**8. Conclusion**

In this paper we proposed the SE block, an architectural unit designed to improve the representational power of a network by enabling it to perform dynamic channel-wise feature recalibration. A wide range of experiments show the effectiveness of SENets, which achieve state-of-the-art performance across multiple datasets and tasks. In addition, SE blocks shed some light on the inability of previous architectures to adequately model channel-wise feature dependencies. We hope this insight may prove useful for other tasks requiring strong discriminative features. Finally, the feature importance values produced by SE blocks may be of use for other tasks such as network pruning for model compression.

이 백서에서는 동적 채널별 기능 재보정을 가능하게 함으로써 네트워크의 표현력을 향상시키도록 설계된 아키텍처 유닛인 SE 블록을 제안했습니다. 다양한 실험을 통해 여러 데이터 세트와 작업에서 최첨단 성능을 달성하는 SENet의 효과를 확인할 수 있습니다. 또한 SE 블록은 이전 아키텍처가 채널별 기능 종속성을 적절하게 모델링할 수 없다는 점을 어느 정도 보여줍니다. 이 통찰력이 강력한 차별적 기능을 필요로 하는 다른 작업에 유용하게 쓰일 수 있기를 바랍니다. 마지막으로 SE 블록에 의해 생성되는 기능 중요도 값은 모델 압축을 위한 네트워크 프루닝 등의 다른 작업에 사용할 수 있습니다.

**Acknowledgements**

The authors would like to thank Chao Li and Guangyuan Wang from Momenta for their contributions in the training system optimisation and experiments on CIFAR dataset. We would also like to thank Andrew Zisserman, Aravindh Mahendran and Andrea Vedaldi for many helpful discussions. The work is supported in part by NSFC Grants (61632003, 61620106003, 61672502, 61571439), National Key R&D Program of China (2017YFB1002701), and Macao FDCT Grant (068/2015/A2). Samuel Albanie is supported by EPSRC AIMS CDT EP/L015897/1.

저자들은 교육 시스템 최적화 및 CIFAR 데이터 세트에 대한 실험에 기여한 모멘타의 Chao Li와 Gwangyuan Wang에게 감사를 표합니다. Andrew Zisserman, Aravindh Mahendran, Andrea Vedaldi도 많은 유익한 논의를 해 주셔서 감사합니다. 이 작업은 NSFC 보조금(61632003, 616206003, 61672502, 61571439), 중국 국가핵심연구개발프로그램(2017YFB1002701), 마카오 FDCT 보조금(068/2015/A2)에 의해 부분적으로 지원된다. Samuel Albanie는 EPSRC AIMS CDT EP/L015897/1에서 지원됩니다.

**Appendix: Details of SENet-154**

SENet-154 is constructed by incorporating SE blocks into a modified version of the 64×4d ResNeXt-152 which extends the original ResNeXt-101 by adopting the block stacking strategy of ResNet-152. Further differences to the design and training of this model (beyond the use of SE blocks) are as follows: (a) The number of the first 1 × 1 convolutional channels for each bottleneck building block was halved to reduce the computational cost of the model with a minimal decrease in performance. (b) The first 7 × 7 convolutional layer was replaced with three consecutive 3 × 3 convolutional layers. (c) The 1 × 1 down-sampling projection with stride-2 convolution was replaced with a 3 × 3 stride-2 convolution to preserve information. (d) A dropout layer (with a dropout ratio of 0.2) was inserted before the classification layer to reduce overfitting. (e) Labelsmoothing regularisation (as introduced in [20]) was used during training. (f) The parameters of all BN layers were frozen for the last few training epochs to ensure consistency between training and testing. (g) Training was performed with 8 servers (64 GPUs) in parallel to enable large batch sizes (2048). The initial learning rate was set to 1.0.

SENet-154는 ResNet-152의 블록 스태킹 전략을 채택하여 오리지널 ResNeXt-101을 확장한 64×4d ResNeXt-152의 수정 버전에 SE 블록을 삽입하여 구축되었다. (SE 블록의 사용을 넘어) 이 모델의 설계 및 훈련에 대한 추가적인 차이는 다음과 같다. (a) 성능의 최소 감소로 모델의 계산 비용을 줄이기 위해 각 병목 구성 블록에 대한 첫 번째 1 × 1 컨볼루션 채널의 수를 절반으로 줄였다. (b) 첫 번째 7 × 7 컨볼루션 계층이 교체되었다. (c) 스트라이드-2 컨볼루션의 1 × 1 다운샘플링 투영법은 정보를 보존하기 위해 3 × 3 스트라이드-2 컨볼루션으로 대체되었다. (d) 드롭아웃 레이어(드롭아웃 비율 0.2)는 과적합을 줄이기 위해 분류 레이어 앞에 삽입되었다. (e) 라벨 무팅 정규화([20]에서 소개한 바와 같이)는 훈련 중에 사용되었습니다. (f) 훈련과 테스트 간의 일관성을 보장하기 위해 지난 몇 번의 훈련 기간 동안 모든 BN 계층의 매개변수가 동결되었습니다. (g) 대규모 배치 크기(2048)를 가능하게 하기 위해 8대의 서버(64 GPU)를 병렬로 훈련했습니다. 초기 학습률은 1.0으로 설정되었습니다.

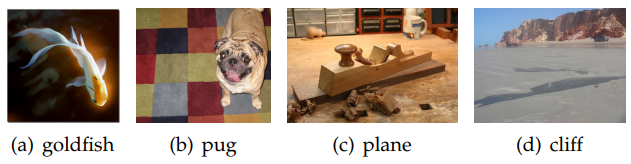


Fig. 8. Sample images from the four classes of ImageNet used in the experiments described in Sec. 7.2.

그림 8. 7.2항에 설명된 실험에 사용된 4개 등급의 ImageNet 샘플 이미지.