CNN, Detection, Segmentation

구름

도시공학과 일반대학원

한양대학교

1. **CNN SOTA**
2. Detection Algorithm
3. Segmentation

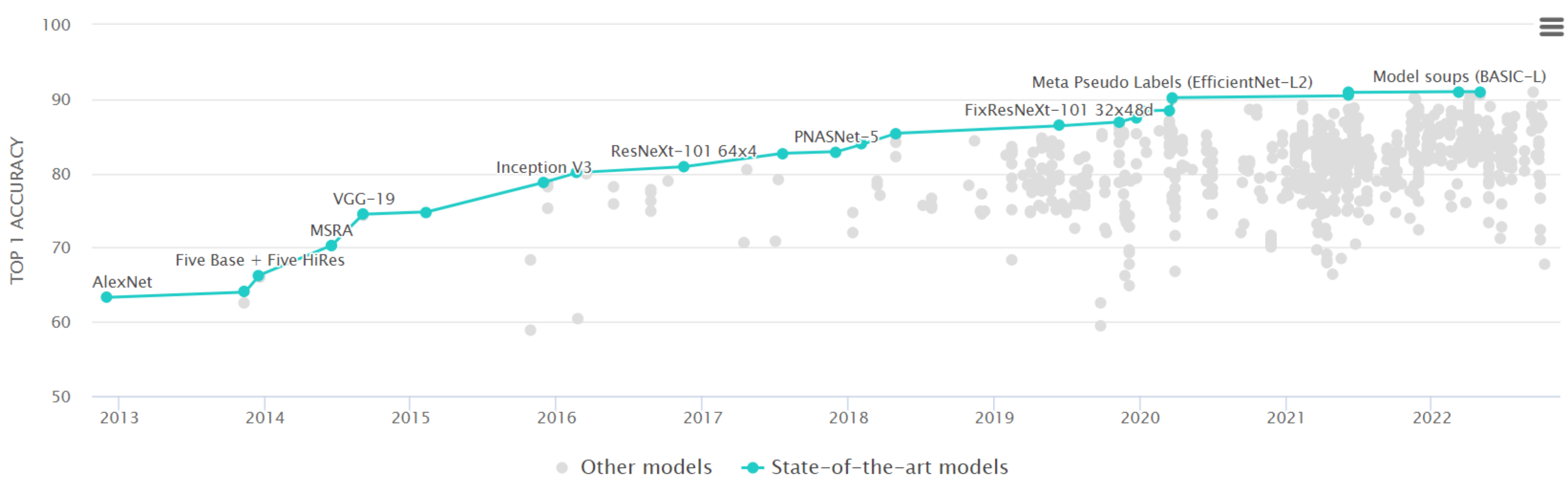
CNN State of the Art (SOTA)

https://paperswithcode.com/sota/image-classification-on-imagenet

Image Classification on ImageNet

Leaderboard Dataset

View Top 1 Accuracy by Date for All models

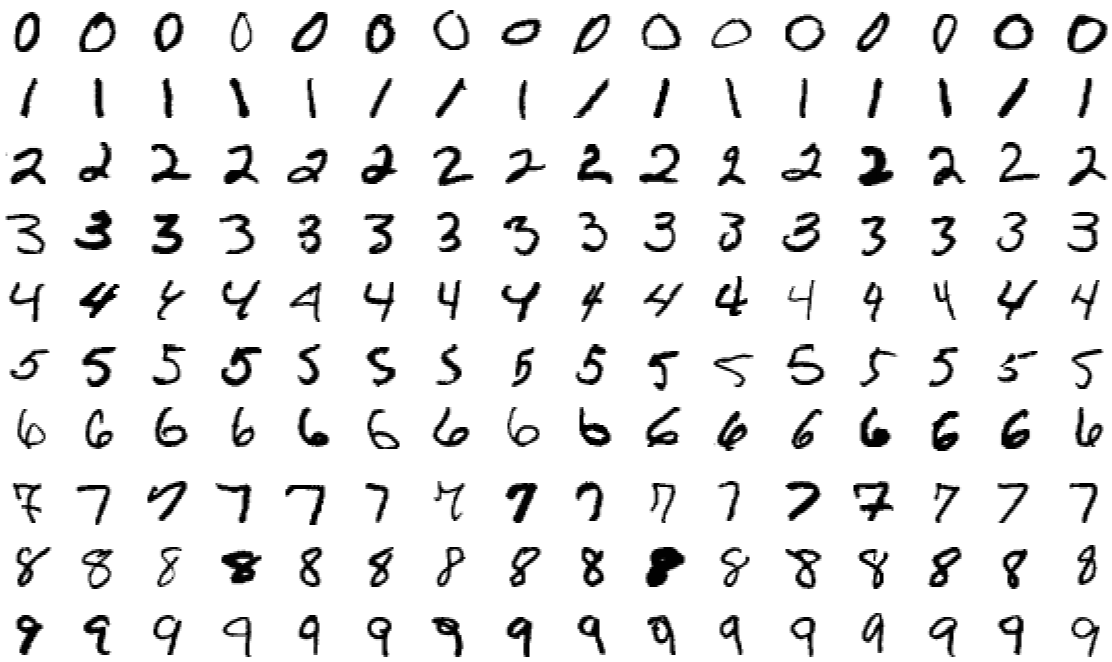


Filter: ImageNet-1k only, Transformer, ResNet, CNN, EfficientNet, ImageNet-22k, JFT-300M, MLP, ResNeXt, JFT-3B, Neighborhood Attention, NAT Transformer, PatchConvnet, Reversible, FPN, CLIP data, ALIGN, Conv+Transformer, IG-1B, Swin-Transformer, CrossCovarianceAttention, FLD-900M, Contrastive, Self-Supervised Learning, RegNet, Mixer, untagged, Hardware Burden, Operations per network pass, Robustness reports

Edit Leaderboard

MNIST

손 글씨 학습데이터 6만 개, 검증데이터 1만 개



LeNet (Gradient-based Learning Applied to Document Recognition)

1998

http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf

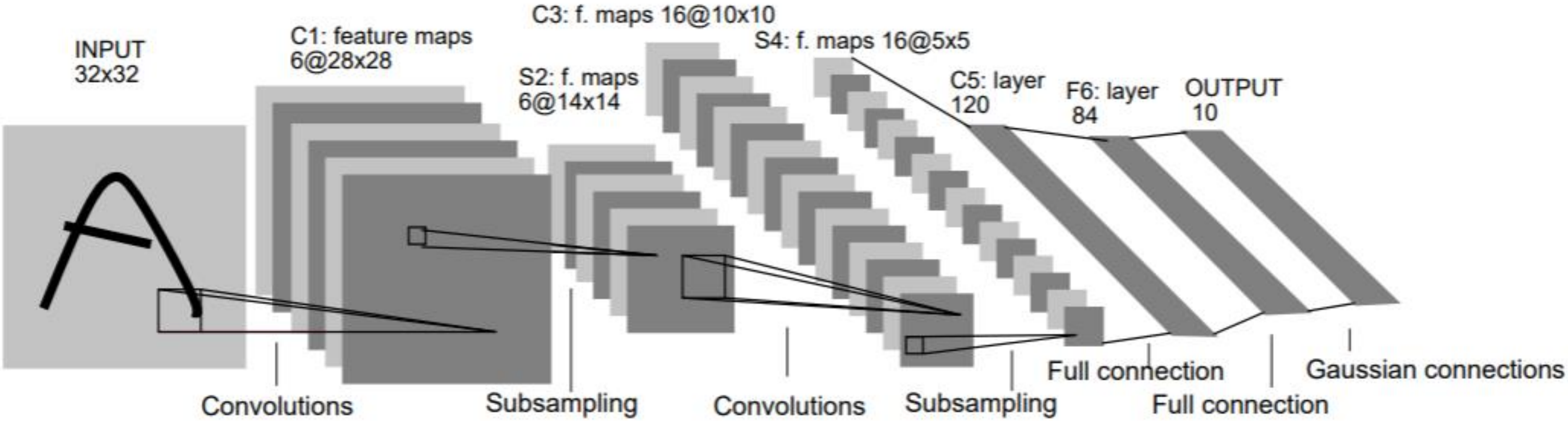


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)

고해상도 이미지셋, 1000개 클래스, 120만개 학습이미지, 5만개 검증 이미지, Labeling 데이터 등

https://image-net.org/challenges/LSVRC/2012/2012-downloads.php

Images

Training images (Task 1 & 2). 138GB. *MD5: 1d675b47d978889d74fa0da5fadfb00e*

Training images (Task 3). 728MB. *MD5: ccaf1013018ac1037801578038d370da*

Validation images (all tasks). 6.3GB. *MD5: 29b22e2961454d5413ddabcf34fc5622*

Test images (all tasks). 13GB. *MD5: e1b8681fff3d63731c599df9b4b6fc02*
If you downloaded ILSVRC 2012 test images on or before 10/10/2019, please apply this patch to replace a subset of images (a total of 2 images are replaced). Note that training and validation images are not affected.

Terms of use: by downloading the image data from the above URLs, you agree to the following terms:

1. *You will use the data only for non-commercial research and educational purposes.*
2. *You will NOT distribute the above URL(s).p*
3. *Stanford University and Princeton University make no representations or warranties regarding the data, including but not limited to warranties of non-infringement or fitness for a particular purpose.*
4. *You accept full responsibility for your use of the data and shall defend and indemnify Stanford University and Princeton University, including their employees, officers and agents, against any and all claims arising from your use of the data, including but not limited to your use of any copies of copyrighted images that you may create from the data.*

Bounding Boxes

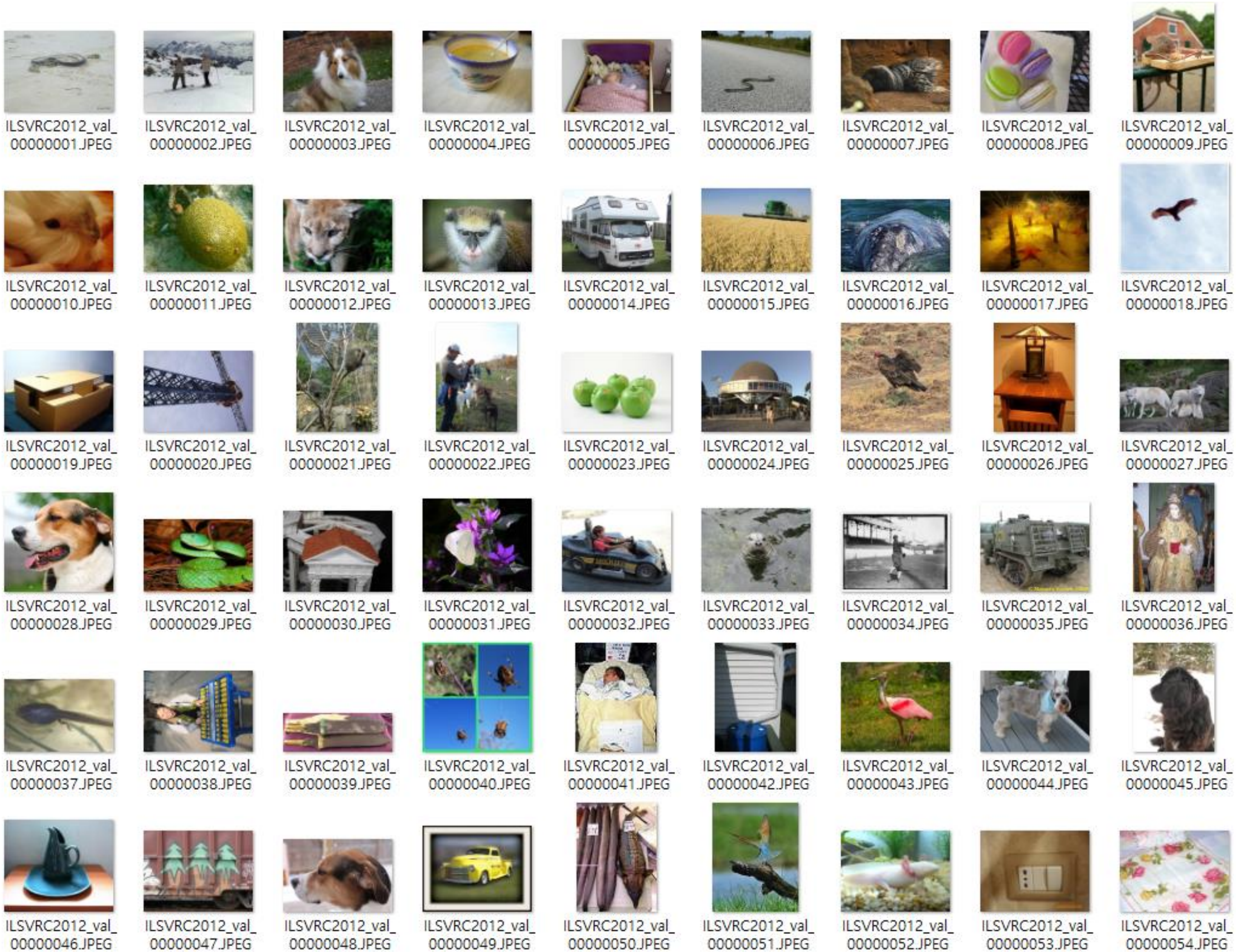
Training bounding box annotations (Task 1 & 2 only) . 20MB. *MD5: 9271167e2176350e65cfe4e546f14b17*

Training bounding box annotations (Task 3 only) . 1MB. *MD5: 61ebd3cc0e4793899a841b6b27f3d6d8*

Validation bounding box annotations (all tasks) . 2.2MB. *MD5: f4cd18b5ea29fe6bbea62ec9c20d80f0*

Test bounding box annotations (Task 3 only). 33MB. *MD5: 2dfdb2677fd9661585d17d5a5d027624*

ILSVRC2012


ILSVRC2012_val_00000001.JPEG
ILSVRC2012_val_00000002.JPEG
ILSVRC2012_val_00000003.JPEG
ILSVRC2012_val_00000004.JPEG
ILSVRC2012_val_00000005.JPEG
ILSVRC2012_val_00000006.JPEG
ILSVRC2012_val_00000007.JPEG
ILSVRC2012_val_00000008.JPEG
ILSVRC2012_val_00000009.JPEG
ILSVRC2012_val_00000010.JPEG
ILSVRC2012_val_00000011.JPEG
ILSVRC2012_val_00000012.JPEG
ILSVRC2012_val_00000013.JPEG
ILSVRC2012_val_00000014.JPEG
ILSVRC2012_val_00000015.JPEG
ILSVRC2012_val_00000016.JPEG
ILSVRC2012_val_00000017.JPEG
ILSVRC2012_val_00000018.JPEG
ILSVRC2012_val_00000019.JPEG
ILSVRC2012_val_00000020.JPEG
ILSVRC2012_val_00000021.JPEG
ILSVRC2012_val_00000022.JPEG
ILSVRC2012_val_00000023.JPEG
ILSVRC2012_val_00000024.JPEG
ILSVRC2012_val_00000025.JPEG
ILSVRC2012_val_00000026.JPEG
ILSVRC2012_val_00000027.JPEG
ILSVRC2012_val_00000028.JPEG
ILSVRC2012_val_00000029.JPEG
ILSVRC2012_val_00000030.JPEG
ILSVRC2012_val_00000031.JPEG
ILSVRC2012_val_00000032.JPEG
ILSVRC2012_val_00000033.JPEG
ILSVRC2012_val_00000034.JPEG
ILSVRC2012_val_00000035.JPEG
ILSVRC2012_val_00000036.JPEG
ILSVRC2012_val_00000037.JPEG
ILSVRC2012_val_00000038.JPEG
ILSVRC2012_val_00000039.JPEG
ILSVRC2012_val_00000040.JPEG
ILSVRC2012_val_00000041.JPEG
ILSVRC2012_val_00000042.JPEG
ILSVRC2012_val_00000043.JPEG
ILSVRC2012_val_00000044.JPEG
ILSVRC2012_val_00000045.JPEG
ILSVRC2012_val_00000046.JPEG
ILSVRC2012_val_00000047.JPEG
ILSVRC2012_val_00000048.JPEG
ILSVRC2012_val_00000049.JPEG
ILSVRC2012_val_00000050.JPEG
ILSVRC2012_val_00000051.JPEG
ILSVRC2012_val_00000052.JPEG
ILSVRC2012_val_00000053.JPEG
ILSVRC2012_val_00000054.JPEG

Revolution of Depth

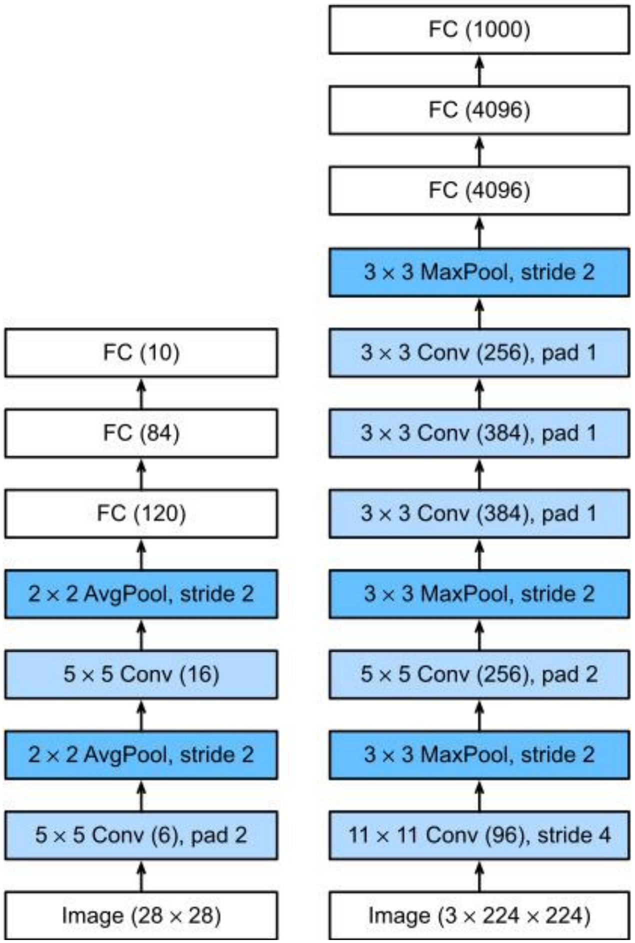
ImageNet Classification top-5 error (%)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". arXiv 2015.

AlexNet (ImageNet Classification with Deep Convolutional Neural Networks)

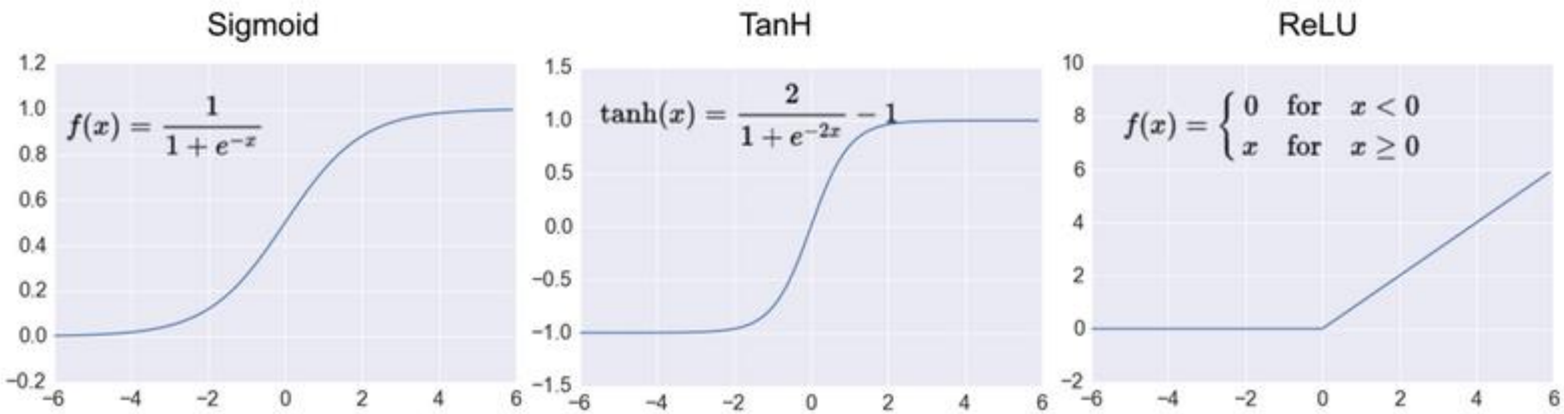
2012

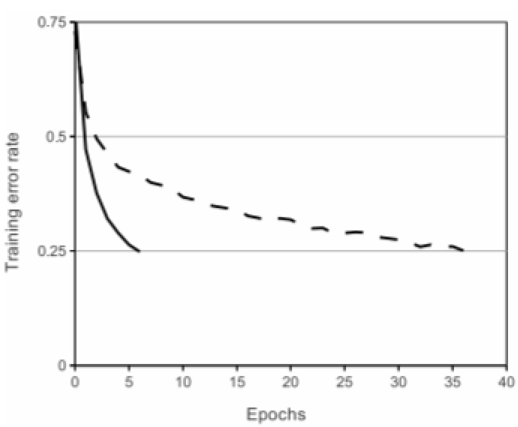
https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf



AlexNet Architecture

3.1 ReLU Nonlinearity





AlexNet Architecture

3.2 Training on Multiple GPUs

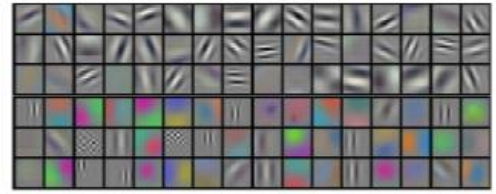
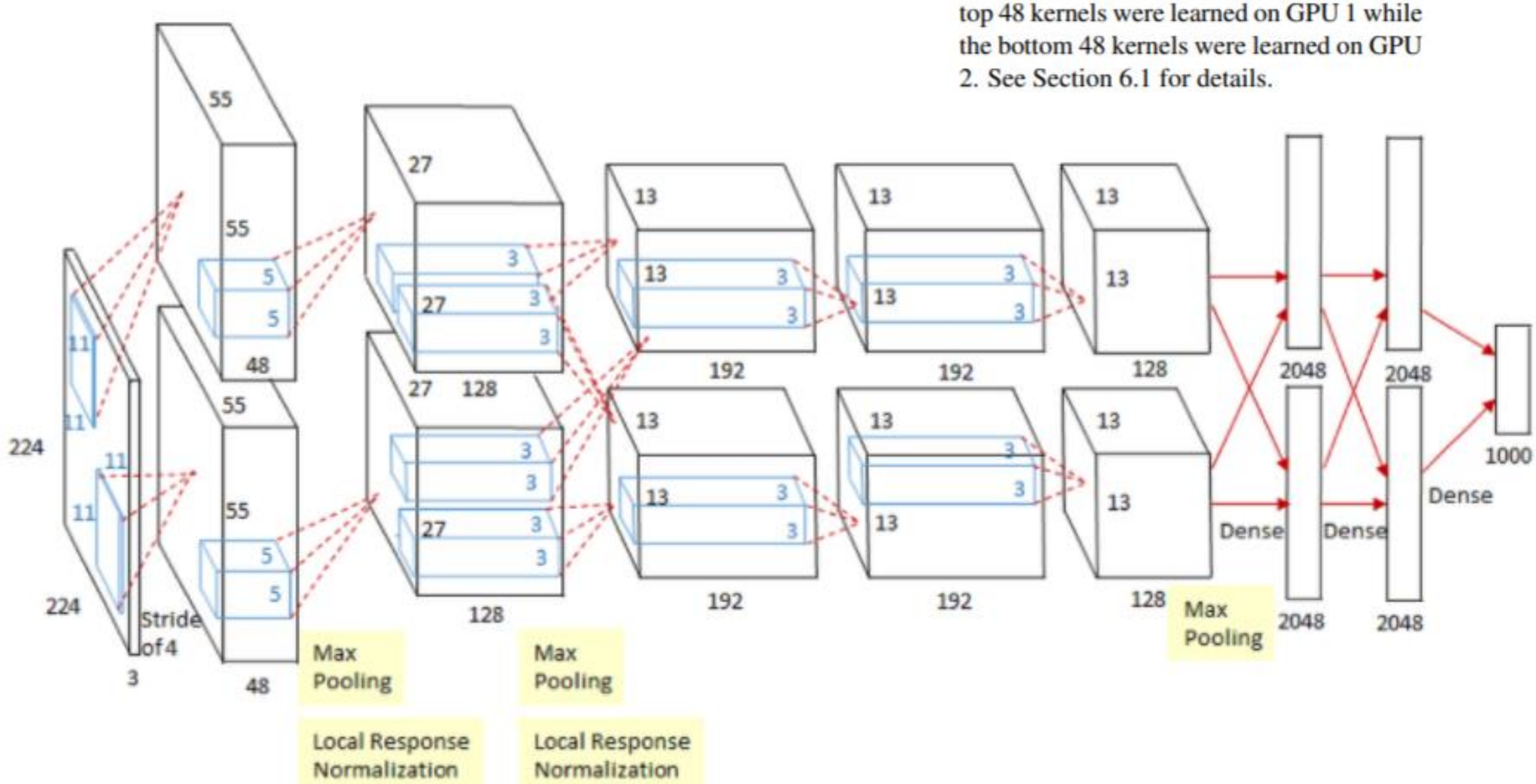


Figure 3: 96 convolutional kernels of size 11×11×3 learned by the first convolutional layer on the 224×224×3 input images. The top 48 kernels were learned on GPU 1 while the bottom 48 kernels were learned on GPU 2. See Section 6.1 for details.

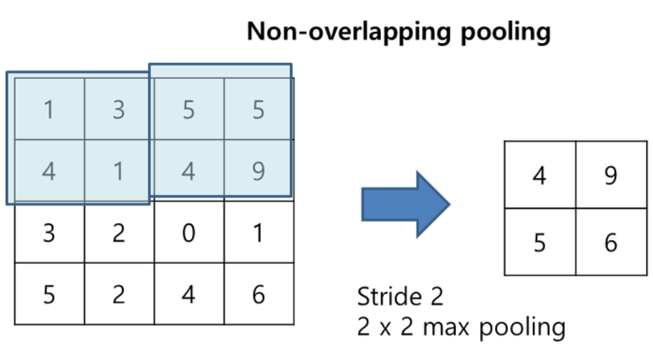


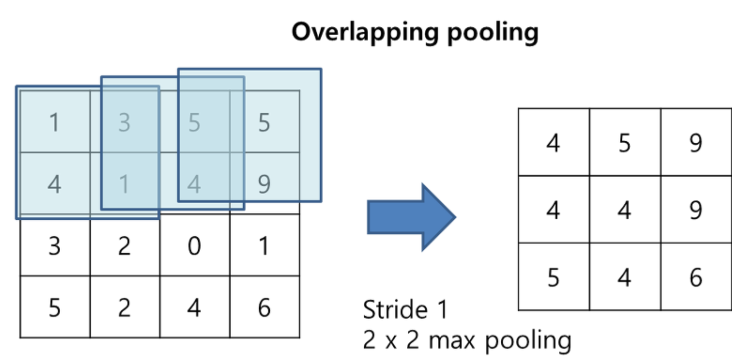
AlexNet Architecture

3.3 Local Response Normalization

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

3.4 Overlapping Pooling



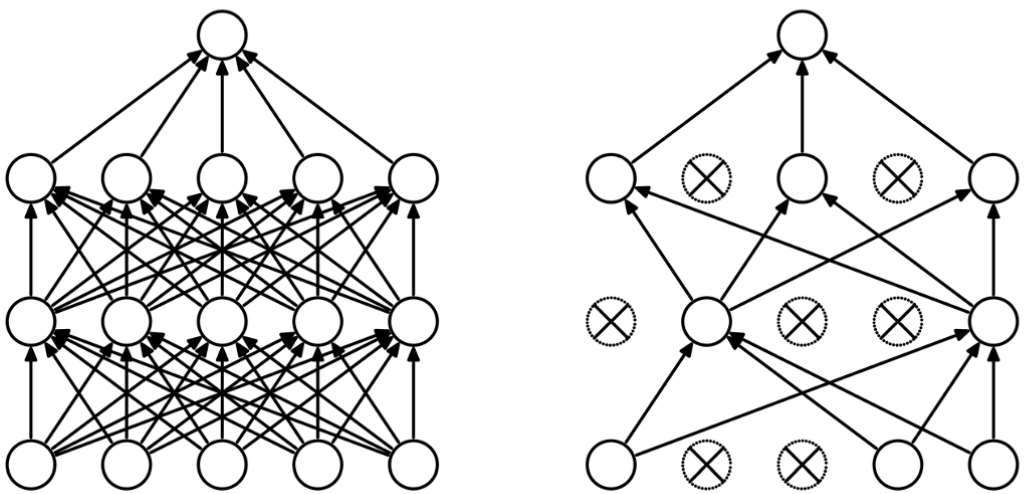


AlexNet Reducing Overfitting

4.1 Data Augmentation

이미지 반전, 이미지 자르기 등을 통해 학습 이미지 양 증가

4.2 Dropout

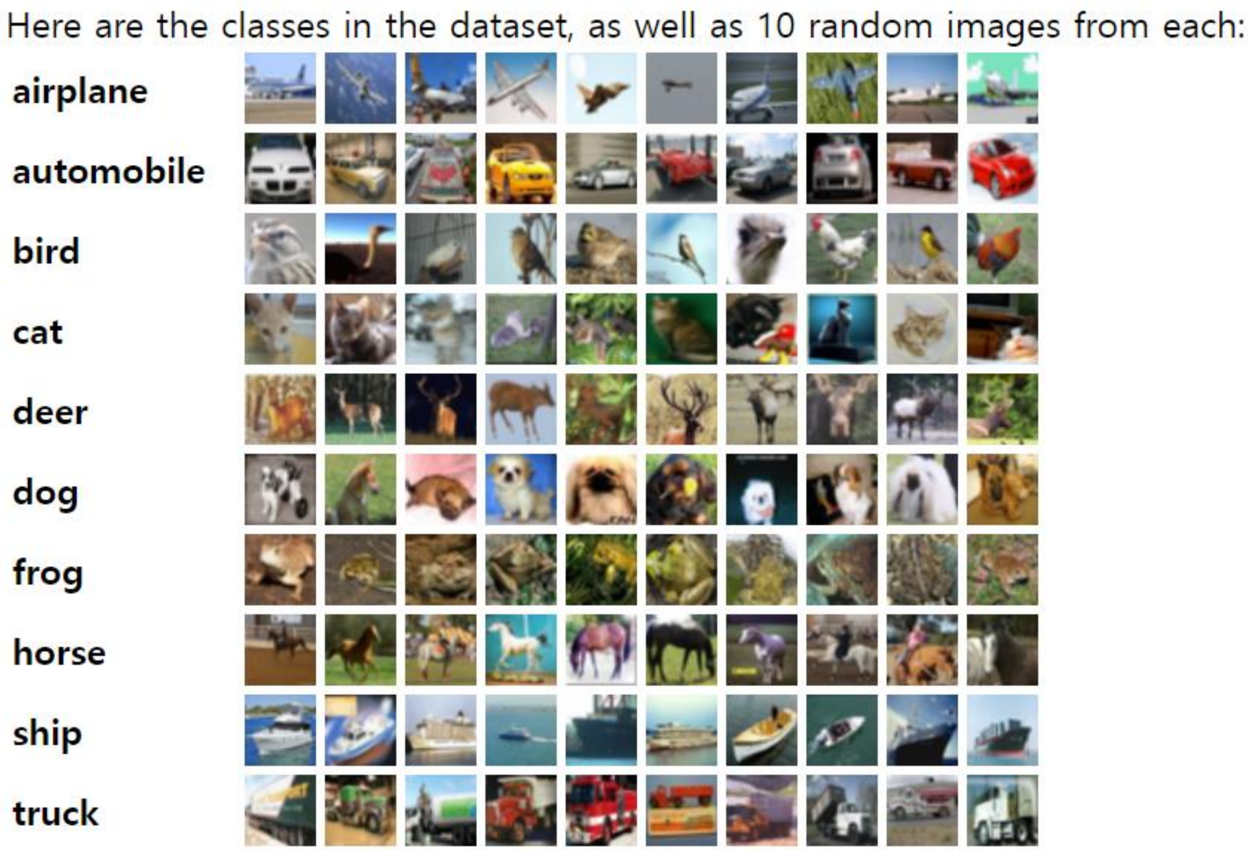


(a) Standard Neural Net (b) After applying dropout.

CIFAR (Canadian Institute For Advanced Research)

32x32 크기의 5만개 학습데이터와 1만개 검증데이터셋

https://www.cs.toronto.edu/~kriz/cifar.html



CIFAR 이미지로 Alexnet : https://daeun-computer-uneasy.tistory.com/35

Keras 지원 데이터 셋

https://keras.io/api/datasets/

Available datasets

MNIST digits classification dataset

- load_data function

CIFAR10 small images classification dataset

- load_data function

CIFAR100 small images classification dataset

- load_data function

IMDB movie review sentiment classification dataset

- load_data function
- get_word_index function

Reuters newswire classification dataset

- load_data function
- get_word_index function

Fashion MNIST dataset, an alternative to MNIST

- load_data function

Boston Housing price regression dataset

- load_data function

Keras 지원 학습 모델

https://keras.io/api/applications/

1. Very Deep CNN

2. Residual Learning

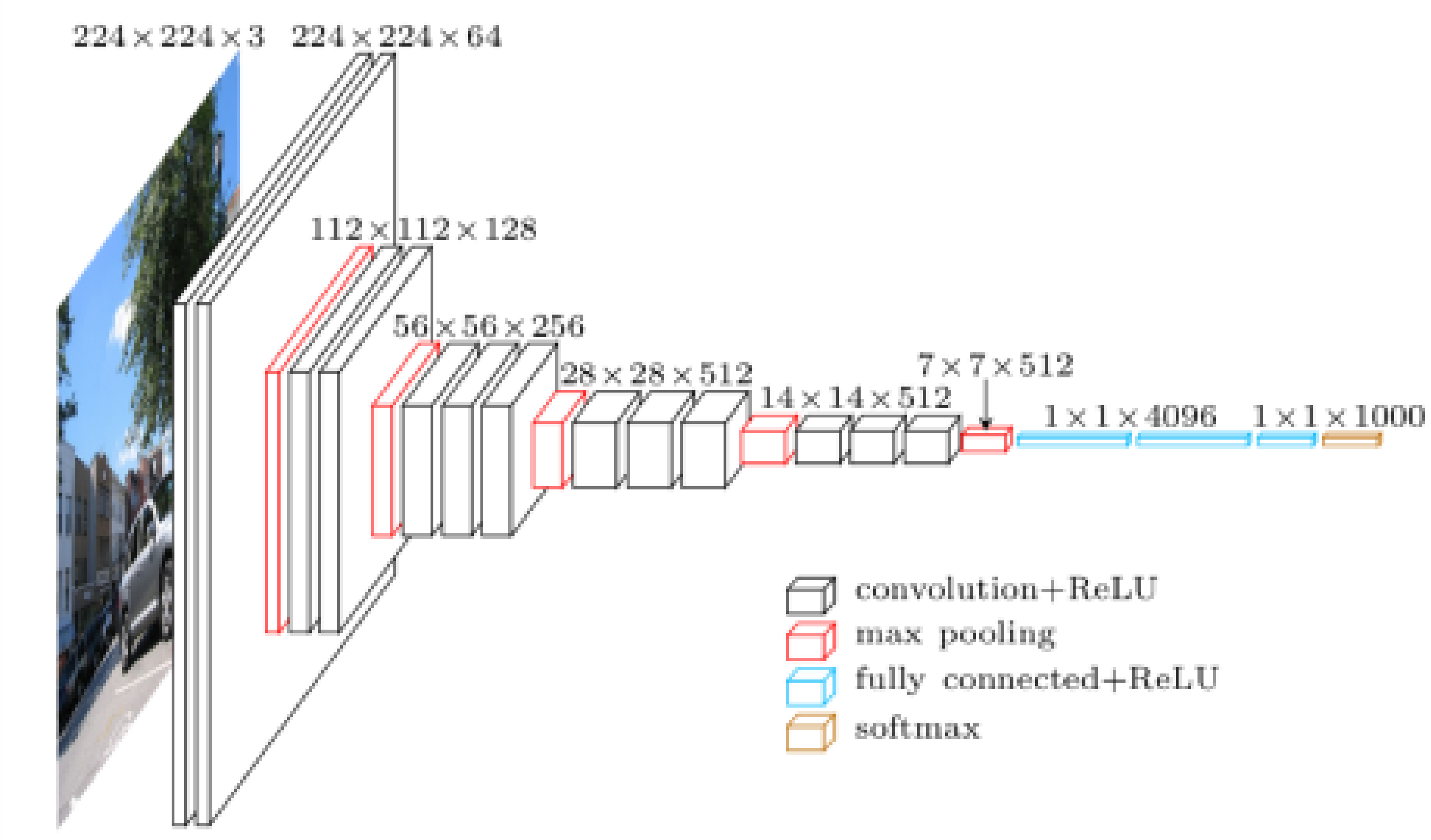
3. DenseNet

4. EfficientNet

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	0.790	0.945	22,910,480	126	109.42	8.06
VGG16	528	0.713	0.901	138,357,544	23	69.50	4.16
VGG19	549	0.713	0.900	143,667,240	26	84.75	4.38
ResNet50	98	0.749	0.921	25,636,712	-	58.20	4.55
ResNet101	171	0.764	0.928	44,707,176	-	89.59	5.19
ResNet152	232	0.766	0.931	60,419,944	-	127.43	6.54
ResNet50V2	98	0.760	0.930	25,613,800	-	45.63	4.42
ResNet101V2	171	0.772	0.938	44,675,560	-	72.73	5.43
ResNet152V2	232	0.780	0.942	60,380,648	-	107.50	6.64
InceptionV3	92	0.779	0.937	23,851,784	159	42.25	6.86
InceptionResNetV2	215	0.803	0.953	55,873,736	572	130.19	10.02
MobileNet	16	0.704	0.895	4,253,864	88	22.60	3.44
MobileNetV2	14	0.713	0.901	3,538,984	88	25.90	3.83
DenseNet121	33	0.750	0.923	8,062,504	121	77.14	5.38
DenseNet169	57	0.762	0.932	14,307,880	169	96.40	6.28
DenseNet201	80	0.773	0.936	20,242,984	201	127.24	6.67
NASNetMobile	23	0.744	0.919	5,326,716	-	27.04	6.70
NASNetLarge	343	0.825	0.960	88,949,818	-	344.51	19.96
EfficientNetB0	29	-	-	5,330,571	-	46.00	4.91
EfficientNetB1	31	-	-	7,856,239	-	60.20	5.55
EfficientNetB2	36	-	-	9,177,569	-	80.79	6.50
EfficientNetB3	48	-	-	12,320,535	-	139.97	8.77
EfficientNetB4	75	-	-	19,466,823	-	308.33	15.12
EfficientNetB5	118	-	-	30,562,527	-	579.18	25.29
EfficientNetB6	166	-	-	43,265,143	-	958.12	40.45
EfficientNetB7	256	-	-	66,658,687	-	1578.90	61.62

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION (VGGnet, Visual Geometry Group)

https://arxiv.org/pdf/1409.1556.pdf



Deep Residual Learning for Image Recognition (ResNet)

https://arxiv.org/pdf/1512.03385.pdf

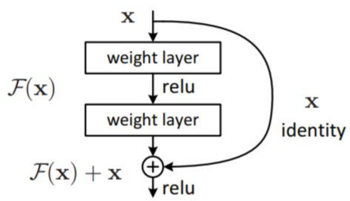


Figure 2. Residual learning: a building block.

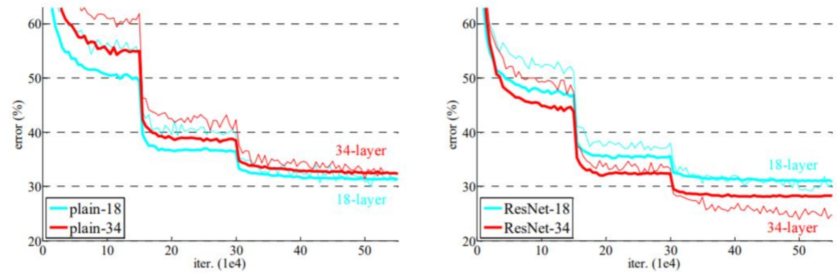
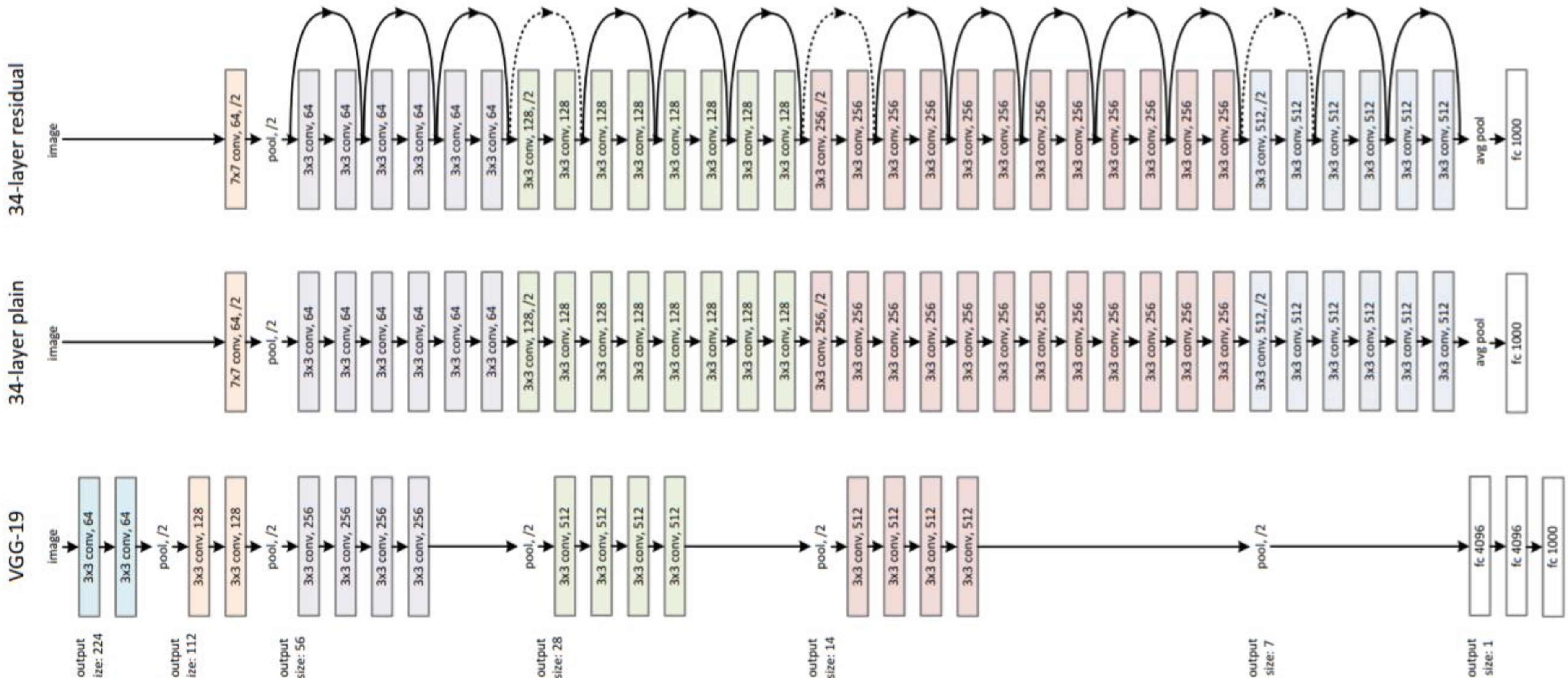


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.



Densely Connected Convolutional Networks (DenseNet)

https://arxiv.org/pdf/1608.06993.pdf

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

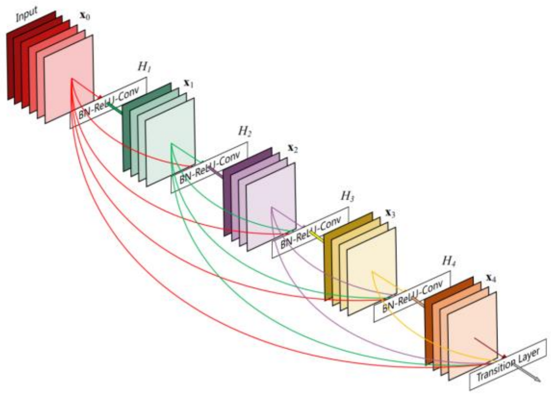


Figure 1: A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.

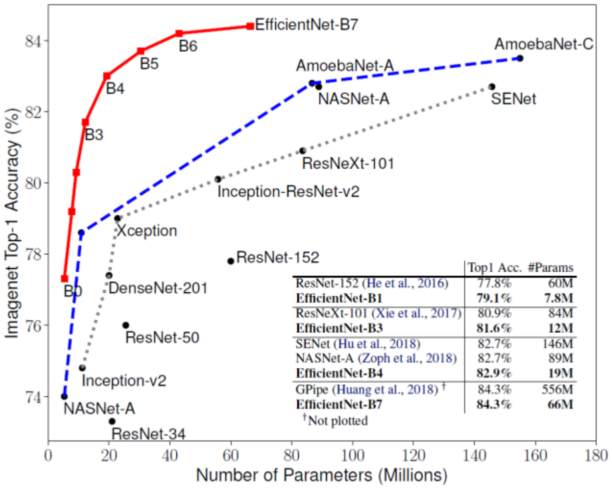
SVHN(The Street View House Numbers)

Method	Depth	Params	C10	C10+	C100	C100+	SVHN
Network in Network [22]	-	-	10.41	8.81	35.68	-	2.35
All-CNN [32]	-	-	9.08	7.25	-	33.71	-
Deeply Supervised Net [20]	-	-	9.69	7.97	-	34.57	1.92
Highway Network [34]	-	-	-	7.72	-	32.39	-
FractalNet [17]	21	38.6M	10.18	5.22	35.34	23.30	2.01
with Dropout/Drop-path	21	38.6M	7.33	4.60	28.20	23.73	1.87
ResNet [11]	110	1.7M	-	6.61	-	-	-
ResNet (reported by [13])	110	1.7M	13.63	6.41	44.74	27.22	2.01
ResNet with Stochastic Depth [13]	110	1.7M	11.66	5.23	37.80	24.58	1.75
	1202	10.2M	-	4.91	-	-	-
Wide ResNet [42]	16	11.0M	-	4.81	-	22.07	-
	28	36.5M	-	4.17	-	20.50	-
with Dropout	16	2.7M	-	-	-	-	1.64
ResNet (pre-activation) [12]	164	1.7M	11.26*	5.46	35.58*	24.33	-
	1001	10.2M	10.56*	4.62	33.47*	22.71	-
DenseNet ($k = 12$)	40	1.0M	**7.00**	5.24	**27.55**	24.42	1.79
DenseNet ($k = 12$)	100	7.0M	**5.77**	**4.10**	**23.79**	**20.20**	1.67
DenseNet ($k = 24$)	100	27.2M	**5.83**	**3.74**	**23.42**	**19.25**	**1.59**
DenseNet-BC ($k = 12$)	100	0.8M	**5.92**	4.51	**24.15**	22.27	1.76
DenseNet-BC ($k = 24$)	250	15.3M	**5.19**	**3.62**	**19.64**	**17.60**	1.74
DenseNet-BC ($k = 40$)	190	25.6M	-	**3.46**	-	**17.18**	-

Table 2: Error rates (%) on CIFAR and SVHN datasets. k denotes network's growth rate. Results that surpass all competing methods are **bold** and the overall best results are **blue**. "+" indicates standard data augmentation (translation and/or mirroring). $*$ indicates results run by ourselves. All the results of DenseNets without data augmentation (C10, C100, SVHN) are obtained using Dropout. DenseNets achieve lower error rates while using fewer parameters than ResNet. Without data augmentation, DenseNet performs better by a large margin.

EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

https://arxiv.org/abs/1905.11946



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.1%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.6%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.9%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.3%**	**66M**

† Not plotted

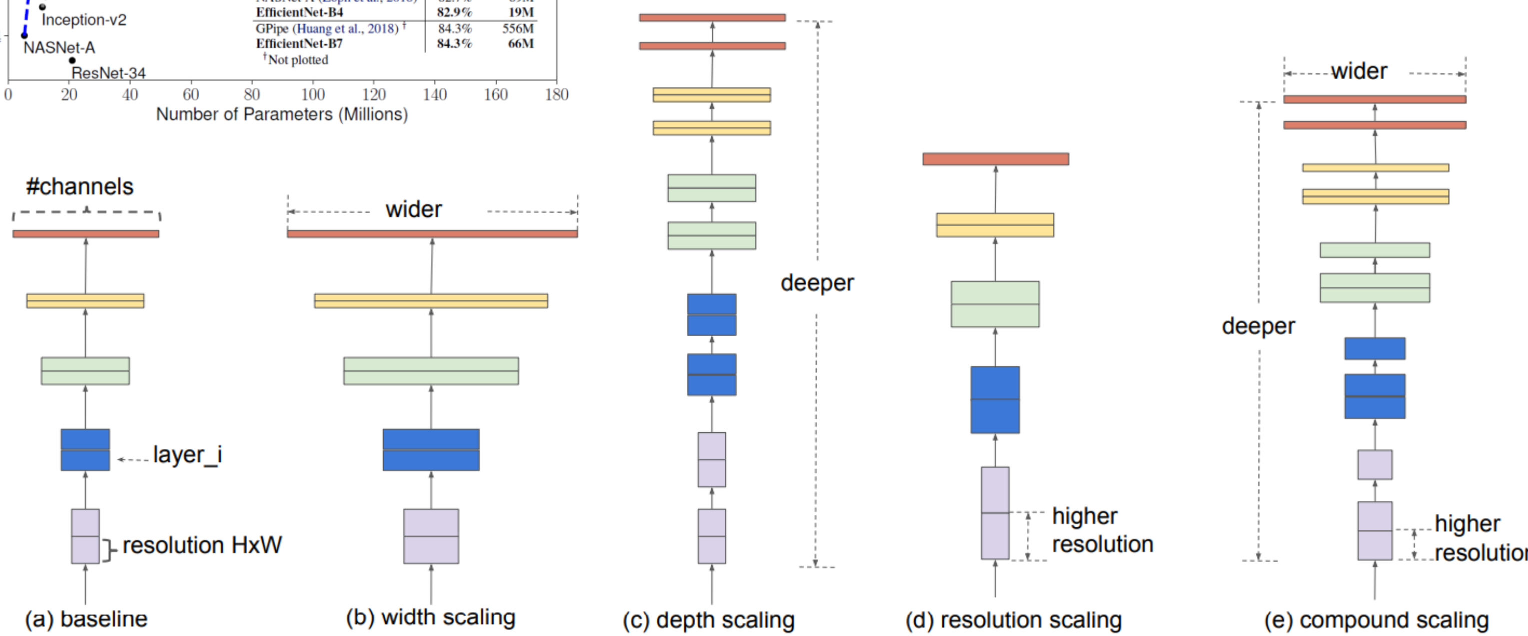


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

1. CNN SOTA
2. **Detection Algorithm**
3. Segmentation

Object Detection in 20 Years: A Survey

https://arxiv.org/pdf/1905.05055.pdf

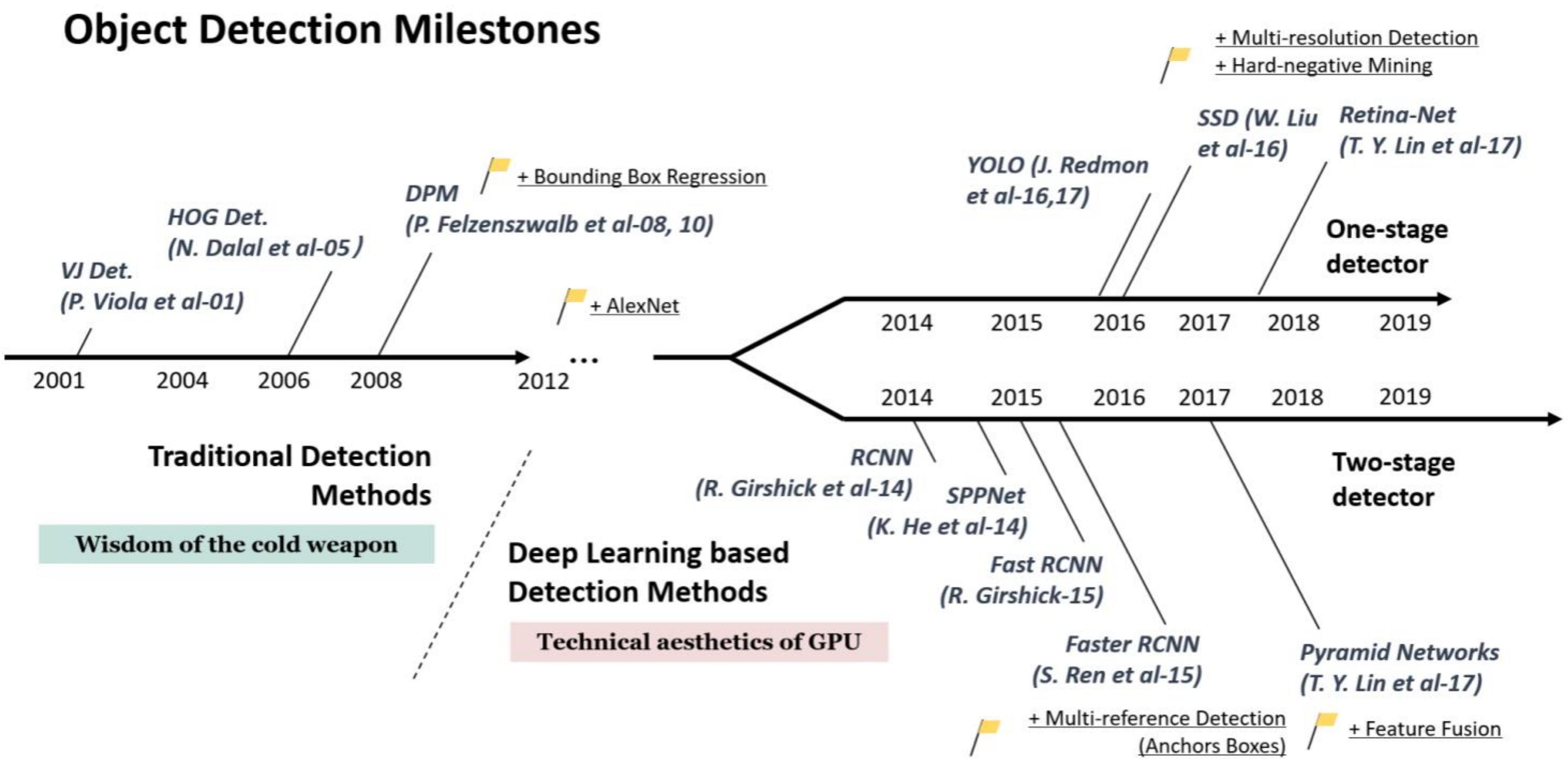
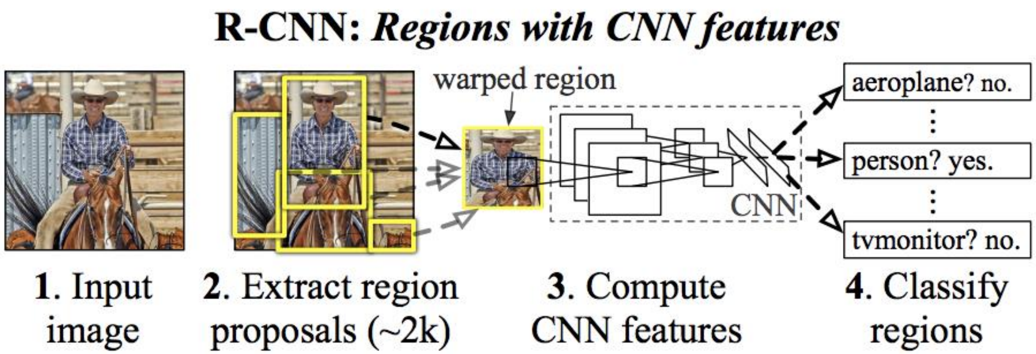


Fig. 2. A road map of object detection. Milestone detectors in this figure: VJ Det. [10, 11], HOG Det. [12], DPM [13–15], RCNN [16], SPPNet [17], Fast RCNN [18], Faster RCNN [19], YOLO [20], SSD [21], Pyramid Networks [22], Retina-Net [23].

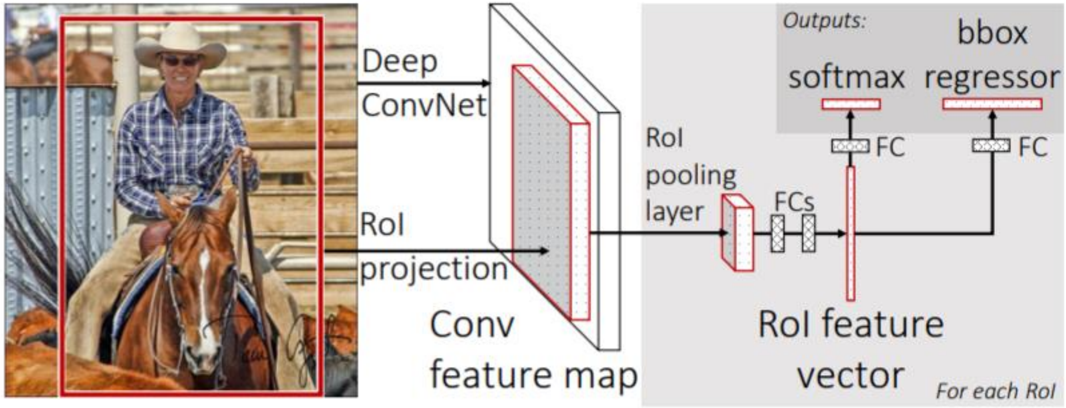
Rich feature hierarchies for accurate object detection and semantic segmentation

https://arxiv.org/pdf/1311.2524.pdf



Fast R-CNN

https://arxiv.org/pdf/1504.08083.pdf



Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

https://arxiv.org/pdf/1506.01497.pdf

(YOLO) You Only Look Once: Unified, Real-Time Object Detection

https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf

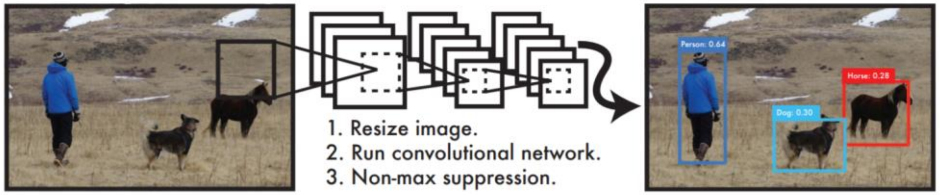


Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448 × 448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

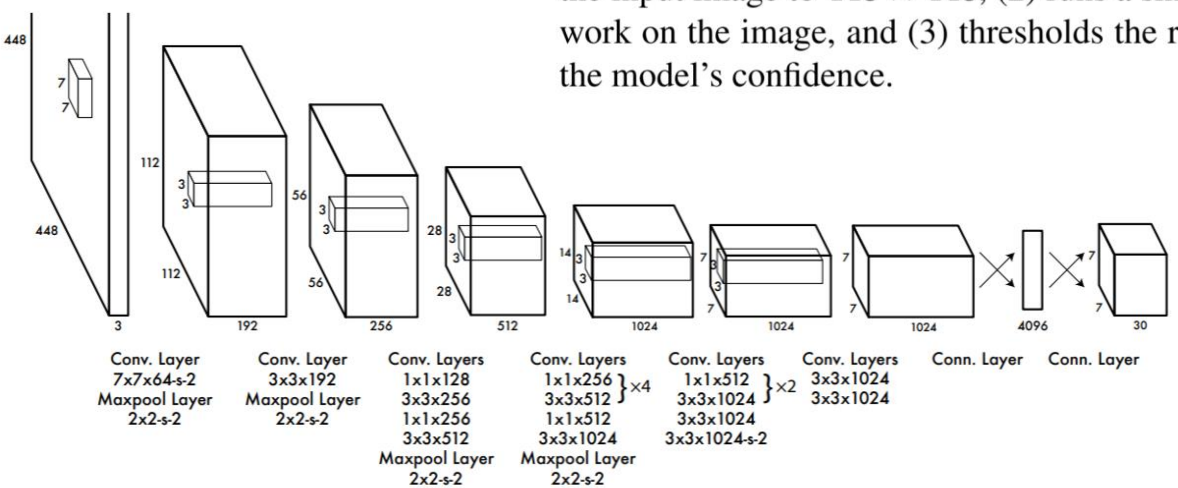
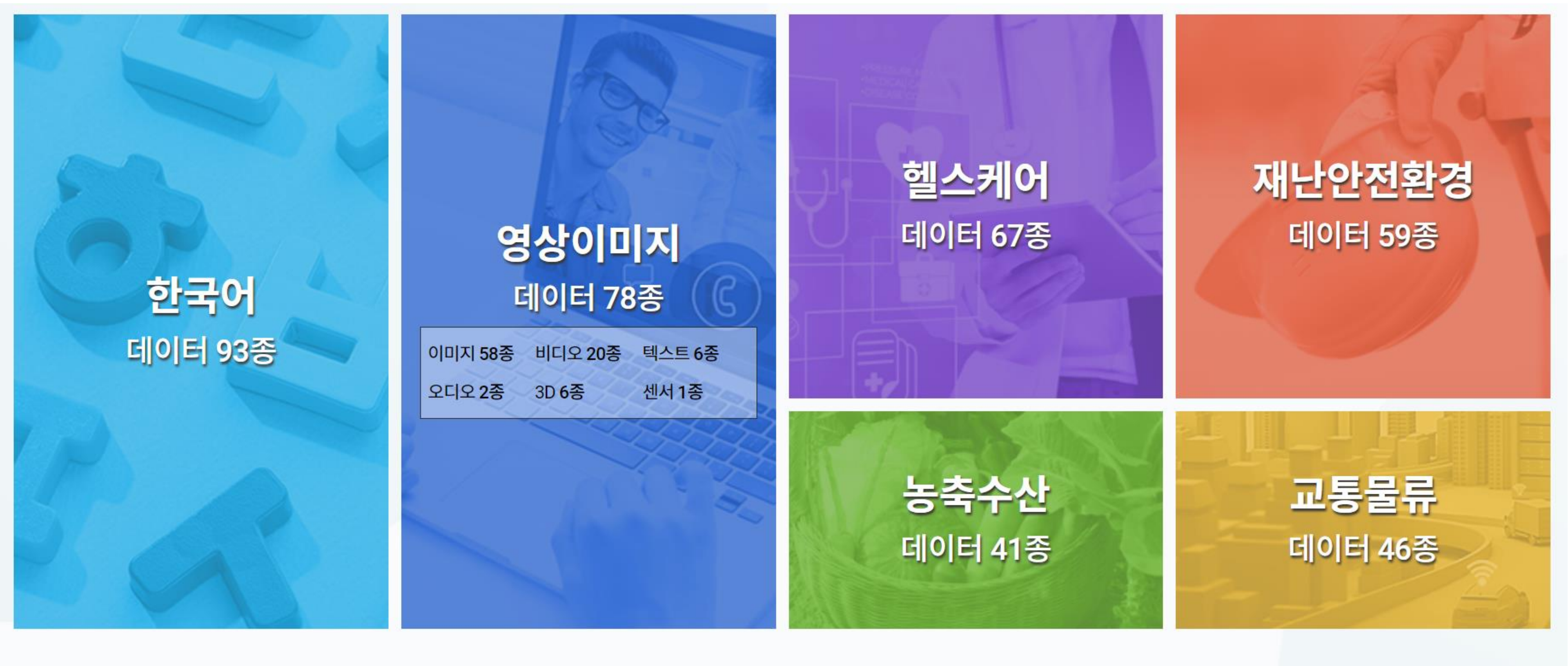


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1 × 1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224 × 224 input image) and then double the resolution for detection.

AI Hub

https://www.aihub.or.kr/



https://github.com/ultralytics/yolov5

https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=165

1. CNN SOTA
2. Detection Algorithm
3. **Segmentation**

#토지피복 # 환경 변화 # 주제도 # 항공사진 # 위성영상

토지 피복지도 항공위성 이미지(강원 및 충청)

강원 및 충청

분야 재난안전환경 유형 이미지

갱신년월 : 2022-10 구축년도 : 2020 조회수 : 265 다운로드 : 163 용량 : 35.59 GB

https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=142

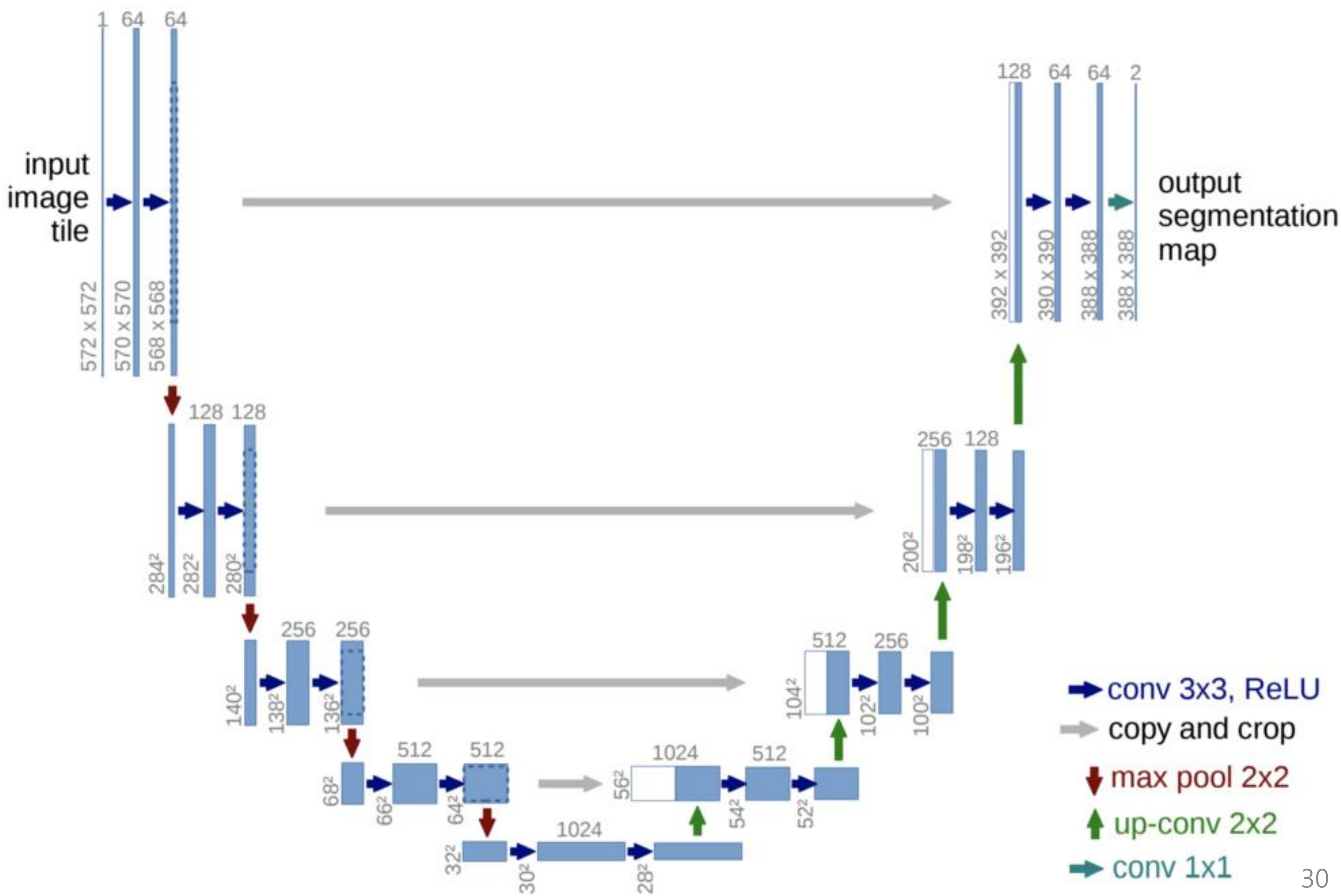
Raster File, tiff format

- 10 건물
- 20 주차장
- 30 도로
- 40 가로수
- 50 논
- 60 밭
- 70 산림
- 80 나지
- 100 비대상지

```
array([[100, 100, 100, 100, 100, 100,  10,  10,  10,  10,  10,  10,  10],
       [100, 100, 100, 100, 100, 100,  10,  10,  10,  10,  10,  10,  10],
       [100, 100, 100, 100, 100, 100,  10,  10,  10,  10,  10,  10,  10],
       [100, 100, 100, 100, 100, 100,  10,  10,  10,  10,  10,  10,  10],
       [100, 100, 100, 100, 100, 100, 100,  10,  10,  10,  10,  10,  10],
       [100, 100, 100, 100, 100, 100, 100,  10,  10,  10,  10,  10,  10],
       [100, 100, 100, 100, 100, 100, 100,  10,  10,  10,  10,  10,  10],
       [100, 100, 100, 100, 100, 100, 100,  10,  10,  10,  10,  10,  10],
       [100, 100, 100, 100, 100, 100, 100,  10,  10,  10,  10,  10,  10],
       [100, 100, 100, 100, 100, 100, 100, 100,  10,  10,  10,  10,  10],
       [100, 100, 100, 100, 100, 100, 100, 100,  10,  10,  10,  10,  10],
       [100, 100, 100, 100, 100, 100, 100, 100,  10,  10,  10,  10,  10],
       [100, 100, 100, 100, 100, 100, 100, 100,  10,  10,  10,  10,  10],
       [100, 100, 100, 100, 100, 100, 100, 100,  10,  10,  10,  10,  10],
       [100, 100, 100, 100, 100, 100, 100, 100, 100,  10,  10,  10,  10],
       [100, 100, 100, 100, 100, 100, 100, 100, 100,  10,  10,  10,  10],
       [100, 100, 100, 100, 100, 100, 100, 100, 100,  10,  10, 100, 100],
       [100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100],
       [100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100],
       [100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100],
       [100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100],
       [100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100],
       [100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100]],
      dtype=uint8)
```

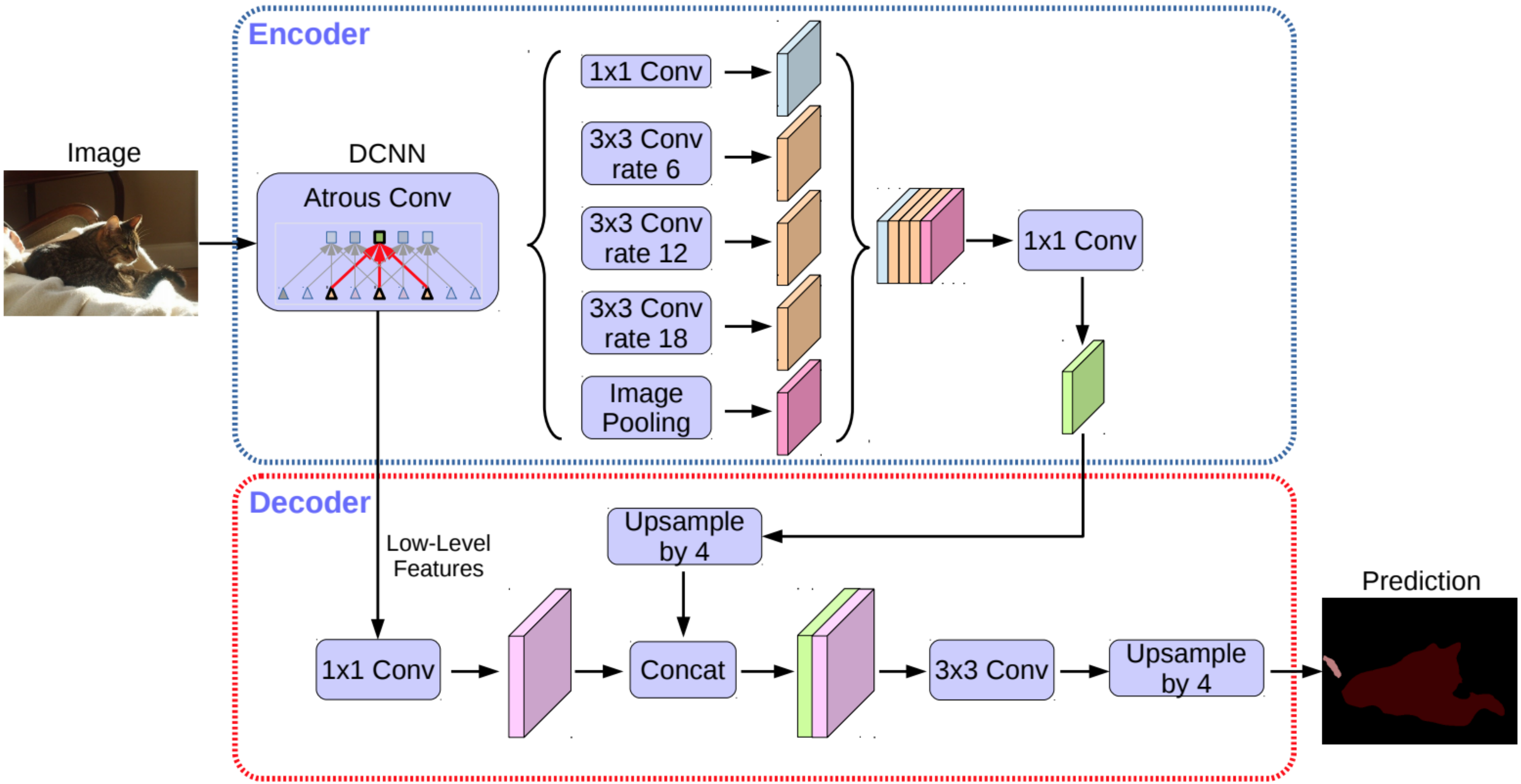
U-Net: Convolutional Networks for Biomedical Image Segmentation

https://arxiv.org/pdf/1505.04597.pdf



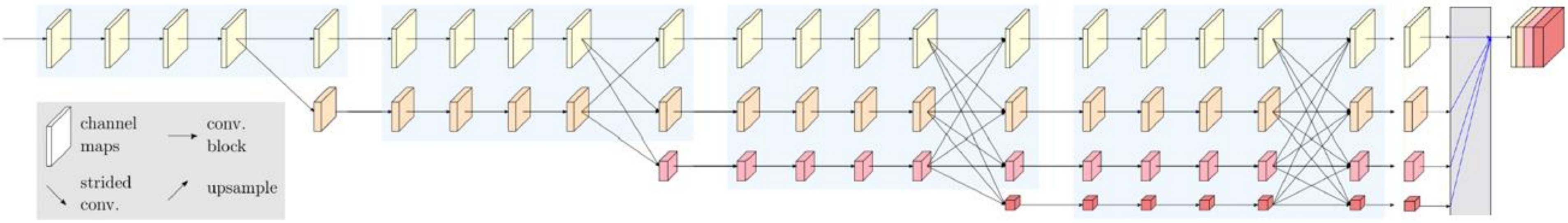
Multiclass semantic segmentation using DeepLabV3+

https://keras.io/examples/vision/deeplabv3_plus/



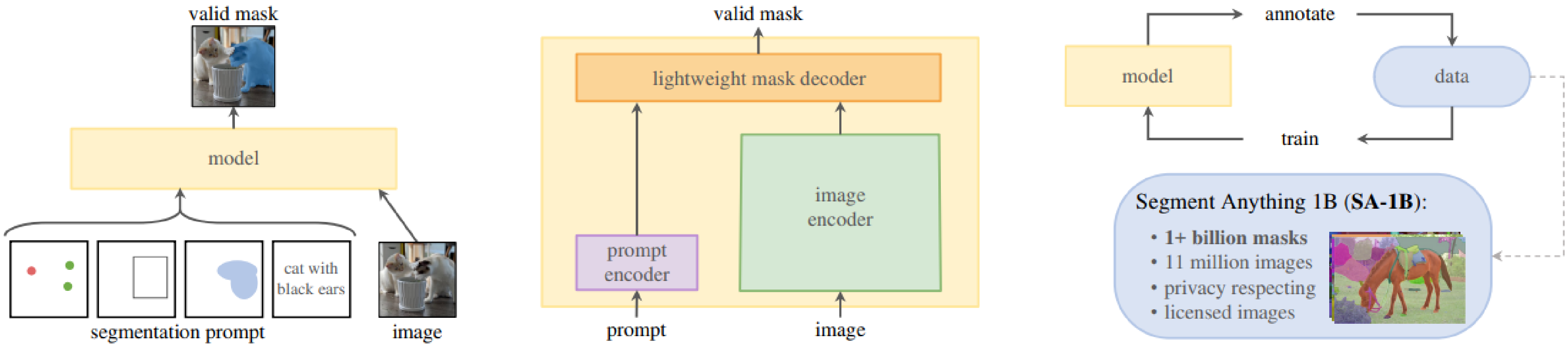
High-resolution networks and Segmentation Transformer for Semantic Segmentation

https://github.com/HRNet/HRNet-Semantic-Segmentation



Segment Anything

https://segment-anything.com/



(a) **Task**: promptable segmentation (b) **Model**: Segment Anything Model (**SAM**) (c) **Data**: data engine (top) & dataset (bottom)

Figure 1: We aim to build a foundation model for segmentation by introducing three interconnected components: a promptable segmentation *task*, a segmentation *model* (SAM) that powers data annotation and enables zero-shot transfer to a range of tasks via prompt engineering, and a *data* engine for collecting SA-1B, our dataset of over 1 billion masks.