Image segmentation: U-Net

UNet is a convolutional neural network (CNN) architecture designed primarily for biomedical image segmentation, introduced in 2015. It follows a **U-shaped** structure with an encoder-decoder design.

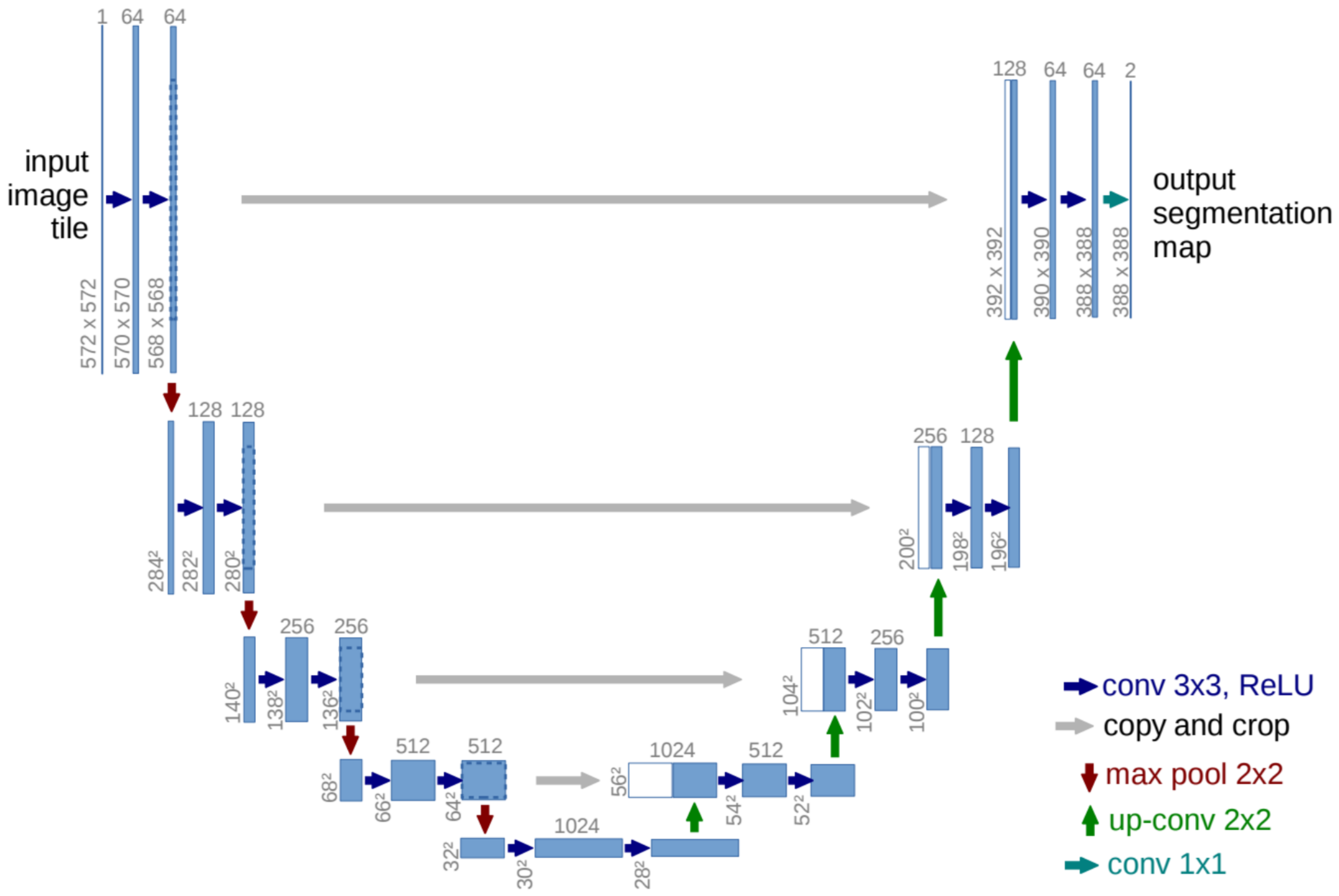
The encoder (**contracting path**) consists of convolutional and pooling layers that progressively extract spatial features and reduce the image size, similar to a traditional CNN.

The decoder (**expanding path**) uses upsampling and skip connections to restore spatial resolution while maintaining fine details, enabling precise segmentation.

Skip connections link corresponding layers from the encoder to the decoder, preserving spatial information and allowing the network to learn both **global and local** features.

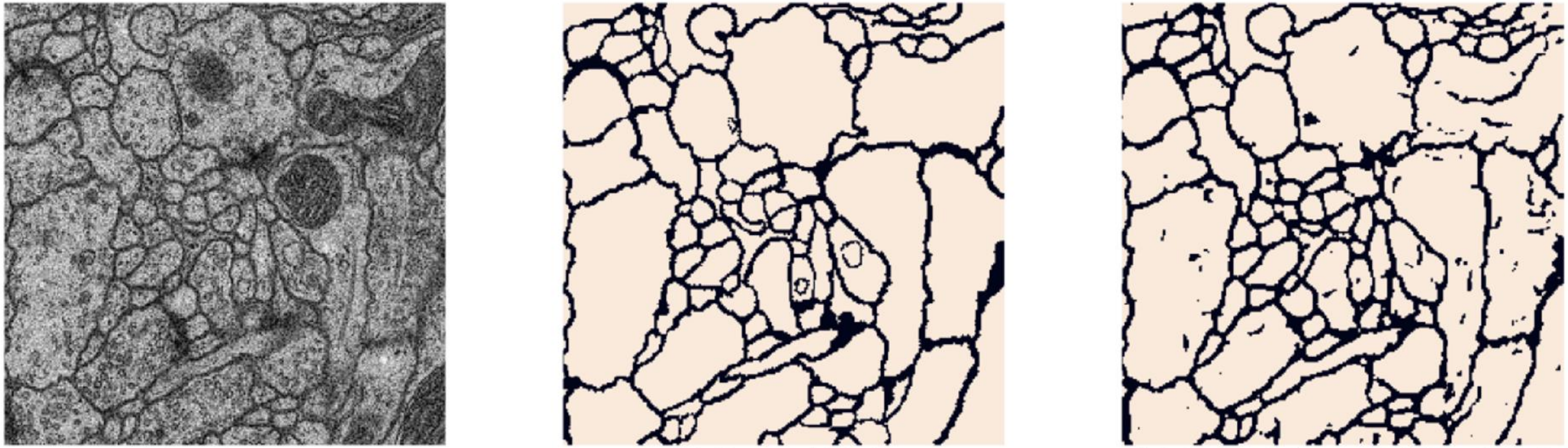
Olaf Ronneberger, Philipp Fischer, Thomas Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015

Image segmentation: U-Net



Olaf Ronneberger, Philipp Fischer, Thomas Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015

Image segmentation: U-Net

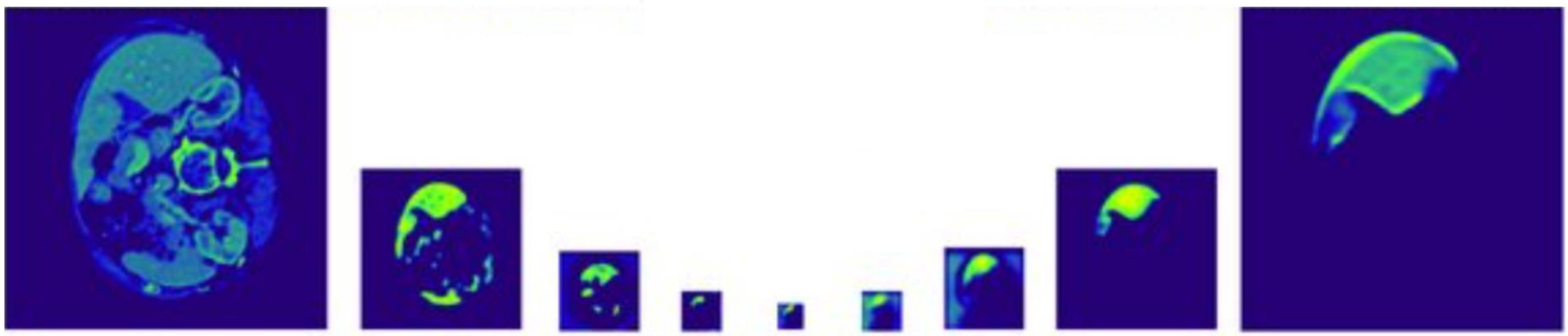


(a) Input Image (b) Ground Truth (c) Output from MultiResUNet

Olaf Ronneberger, Philipp Fischer, Thomas Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015

Image segmentation: U-Net

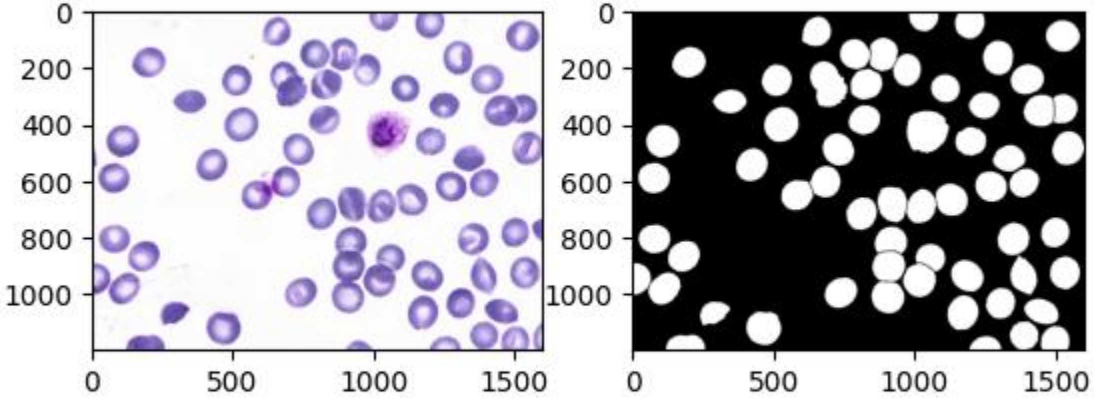
Apply U-Net to segment relevant regions in an image first, then use the segmented output to classify the image based on the presence, shape, or characteristics of the detected regions.



Olaf Ronneberger, Philipp Fischer, Thomas Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015

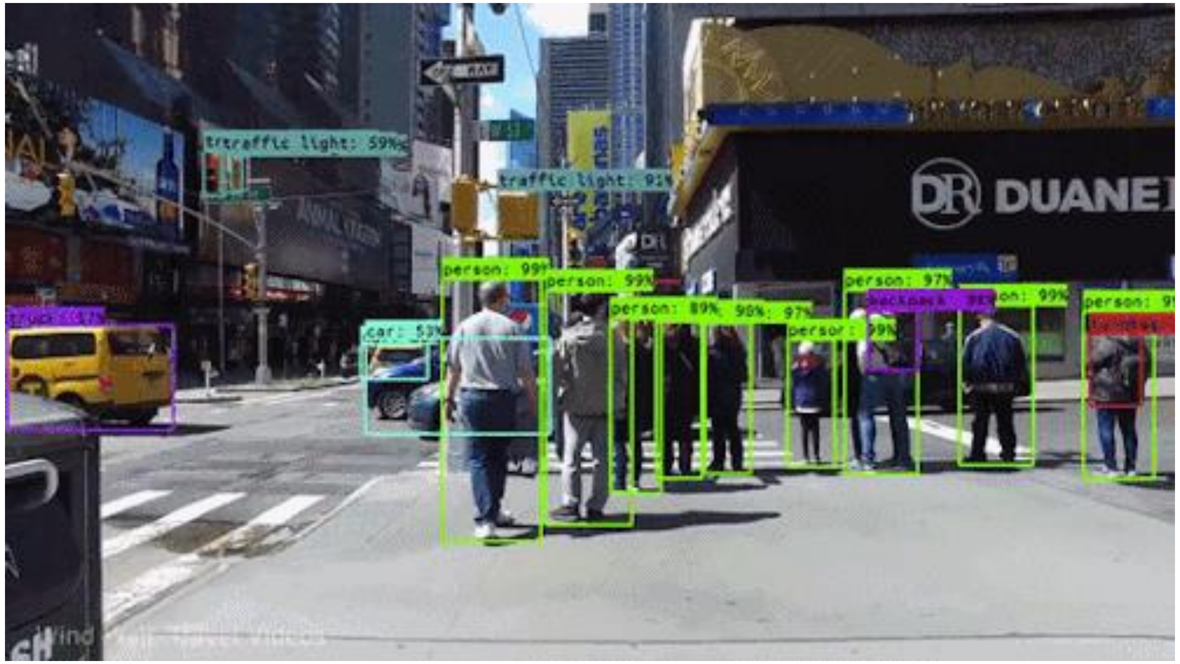
U-Net

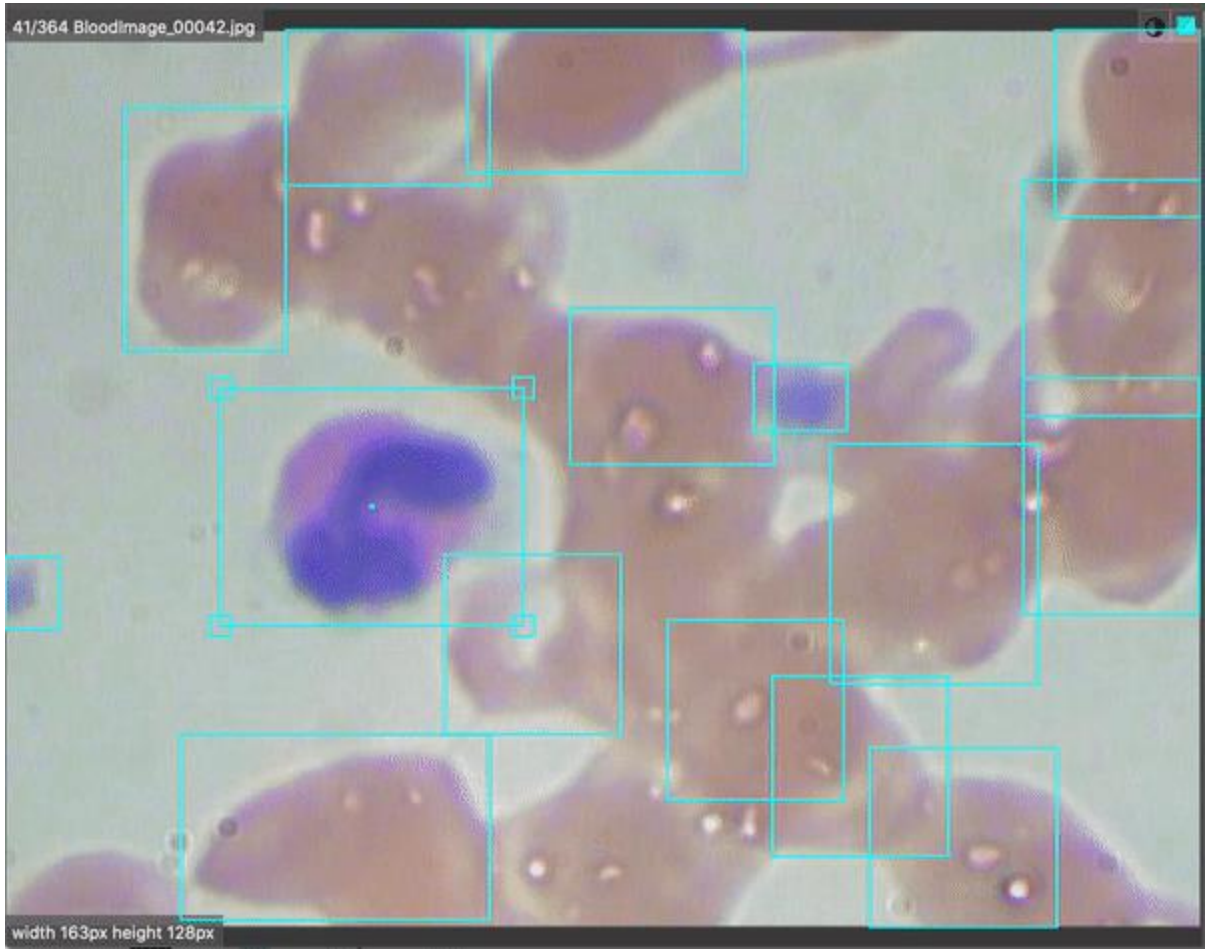
GitHub



unet-blood-cell-segmentation.ipynb

Object recognition: YOLO (You Only Look Once)





Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. CVPR

Object recognition: YOLO (You Only Look Once)

Grid-based detection: The image is divided into an **S×S grid**, where each grid cell is responsible for predicting objects whose centers fall within it.

Bounding box and class predictions: Each grid cell predicts **B bounding boxes** (with coordinates, width, height, and confidence scores) and C class probabilities, all in a single tensor.

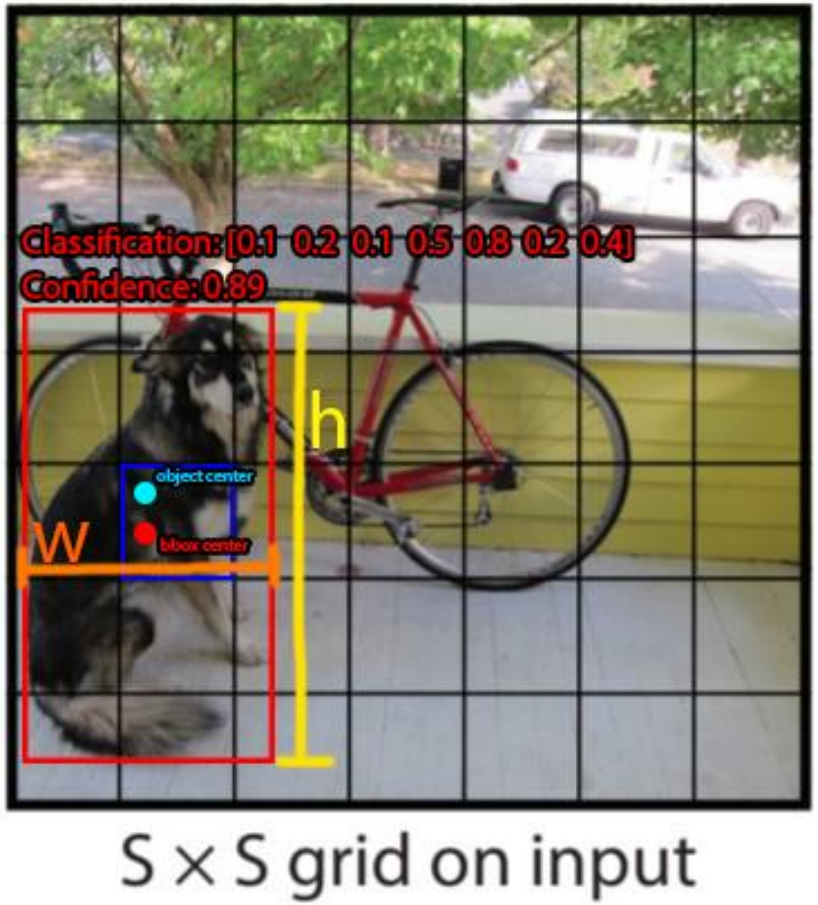
Non-maximum suppression (NMS): To eliminate redundant detections, YOLO applies NMS, keeping only the highest-confidence bounding boxes while discarding overlapping ones.

Single-pass object detection: YOLO processes the entire image in one forward pass through a deep neural network, making it extremely fast.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. CVPR

Object recognition: YOLO (You Only Look Once)

Grid-based detection: The image is divided into an S×S grid, where each grid cell is responsible for predicting objects whose centers fall within it.



Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. CVPR

Object recognition: YOLO (You Only Look Once)

Non-maximum suppression (NMS): To eliminate redundant detections, YOLO applies NMS, keeping only the highest-confidence bounding boxes while discarding overlapping ones.

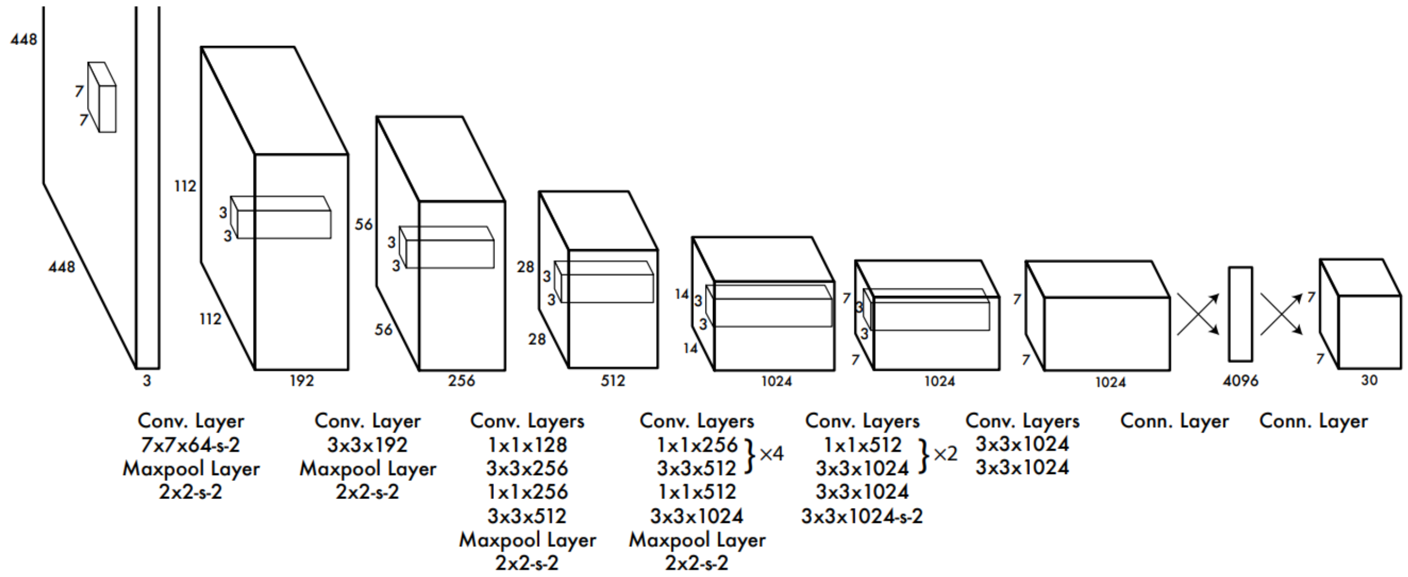
Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. CVPR

Object recognition: YOLO (You Only Look Once)

Single-pass object detection: YOLO processes the entire image in one forward pass through a deep neural network, making it extremely fast.

The final output of YOLO is a tensor of shape (S, S, B × (5 + C)), meaning:

- Each (i, j) position in this output tensor corresponds to the (i, j) grid cell in the original image.
- Each grid cell contains B bounding boxes, each predicting:
    - Object's center (x, y)
    - Size (w, h)
    - Confidence score
    - Class probabilities (C classes)



Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. CVPR

Common Objects in Context dataset (COCO)

COCO is a large-scale dataset for object detection, segmentation, and captioning that contains over 330,000 images with 1.5 million object instances across 80 categories.

It includes bounding boxes, object masks (segmentation), keypoints for human poses, and dense image captions, making it highly versatile for multiple computer vision tasks.

Serves as a benchmark dataset for state-of-the-art object detection and segmentation models, including YOLO and other deep learning architectures.

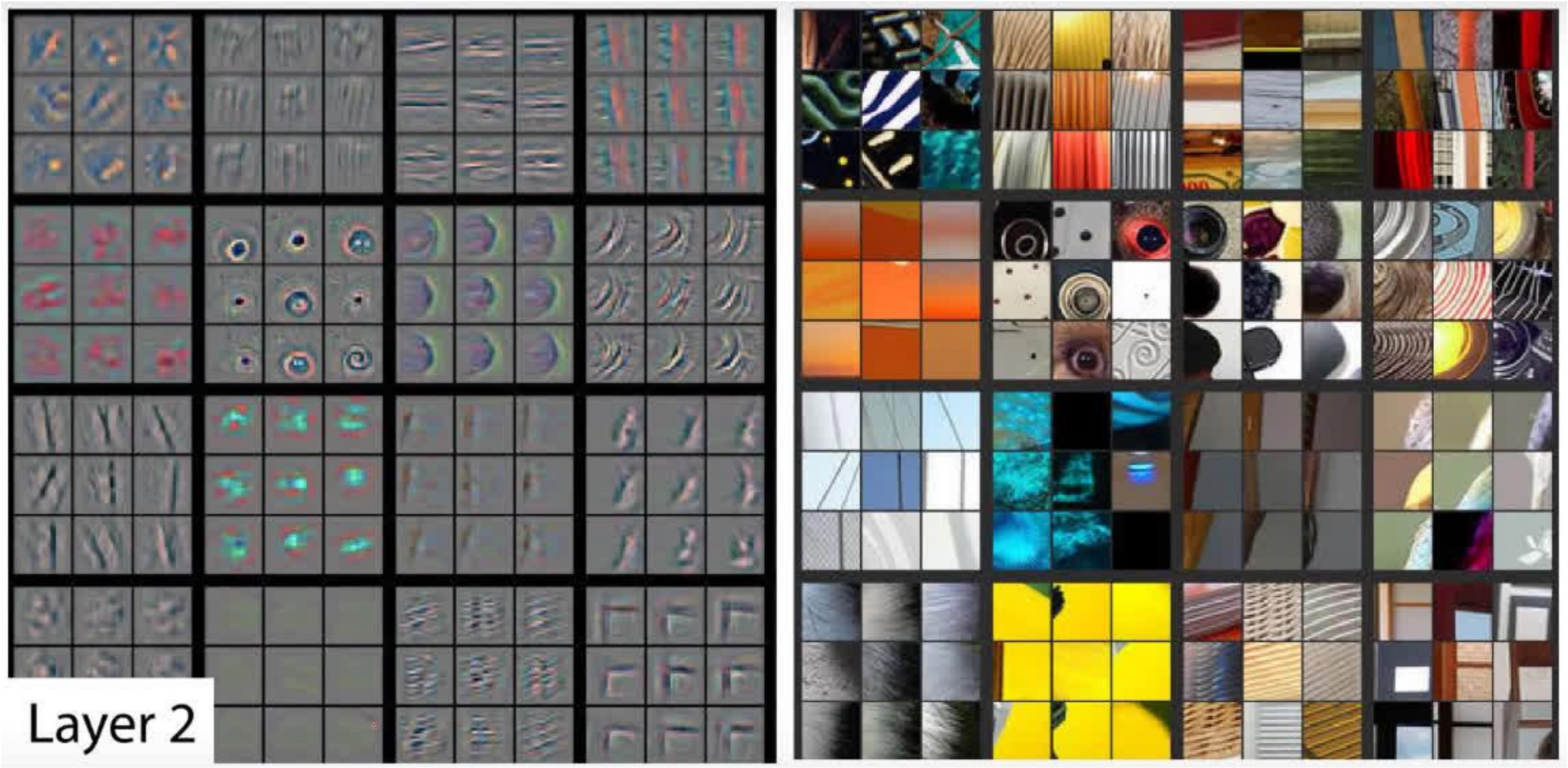
Lin, T.-Y. et al. (2014). Microsoft COCO: Common Objects in Context. ECCV 2014, pp. 740–755.

What computers see

206	205	247	245	244	253	247	245	136	151	255	255	255	255	255	255	234	207	231	255	254	254	255	255	254	255	252	255	255	254	255	247
244	161	137	244	254	255	254	255	118	103	209	228	155	153	236	193	74	52	68	173	255	254	254	255	255	255	254	255	254	253	244	184
192	154	75	200	249	255	255	255	110	98	84	61	35	44	89	53	44	45	43	54	140	213	253	255	255	255	255	245	187	186	176	223
90	109	98	143	223	255	255	252	117	75	41	35	31	24	25	38	45	44	44	46	81	118	148	234	252	254	255	248	231	248	255	254
67	69	107	196	236	255	255	255	104	25	34	35	29	20	25	34	32	30	32	34	53	85	100	142	231	242	247	249	255	255	255	255
55	51	45	134	218	251	255	232	51	12	28	33	24	24	46	75	82	78	71	68	58	53	67	90	136	228	208	158	253	246	249	255
79	58	56	75	224	255	255	118	11	27	74	99	91	106	140	162	173	173	173	172	158	137	92	46	78	187	217	206	254	222	233	255
38	43	47	52	147	255	229	56	41	81	129	145	160	169	169	172	178	179	178	179	177	177	172	110	31	82	209	238	255	244	249	255
40	40	33	36	90	245	171	32	65	110	139	145	151	162	171	174	178	179	182	184	187	183	173	162	71	45	167	255	254	255	254	255
37	44	44	31	69	250	158	36	70	129	143	142	153	162	171	175	177	178	182	191	194	188	180	170	120	51	137	255	254	250	254	255
34	45	51	64	116	237	181	53	116	138	140	143	154	164	176	178	174	177	183	186	185	185	183	178	140	66	141	254	252	225	249	255
34	36	52	74	71	188	156	63	131	134	144	155	160	161	173	179	178	179	189	193	190	185	187	182	156	93	148	250	254	214	247	255
32	38	52	54	159	250	126	57	129	138	138	140	151	156	166	168	171	178	180	187	186	185	185	183	180	102	136	242	255	255	254	254
36	32	72	129	212	228	115	65	121	104	102	104	94	103	134	158	170	162	125	108	121	143	155	190	191	104	134	230	253	253	255	251
61	82	116	107	179	247	124	60	101	90	111	119	103	81	94	147	191	178	126	98	123	153	147	161	200	92	100	222	207	167	227	215
144	178	167	231	210	232	170	67	115	88	76	62	83	85	88	139	192	190	135	80	53	99	141	165	201	97	79	192	245	235	248	249
127	145	149	195	204	213	197	95	133	122	117	133	126	108	110	139	191	197	167	129	127	148	147	171	188	110	121	228	233	180	215	212
87	112	100	79	85	82	65	75	142	148	151	153	138	125	120	149	191	190	193	175	174	193	198	190	208	127	163	239	219	149	198	195
63	83	109	134	129	106	39	78	132	142	155	159	139	111	124	164	195	200	186	192	191	195	200	202	200	143	217	253	249	242	238	234
69	78	78	113	97	74	43	106	127	140	152	155	125	97	112	150	185	194	174	183	196	198	202	208	209	166	247	254	255	254	254	254
72	44	63	59	46	52	49	74	127	137	146	149	132	103	78	90	134	141	168	165	199	207	204	203	216	193	236	244	251	$\boxed{242}$	236	243
55	20	69	73	59	80	46	74	117	127	144	161	148	124	105	120	156	187	193	162	189	206	201	205	214	194	174	185	197	188	183	193
65	49	77	89	50	68	43	61	109	127	141	147	113	100	121	145	148	169	181	176	181	201	201	205	202	174	166	169	178	183	188	184
82	76	92	79	54	58	37	47	90	121	132	116	89	78	111	146	163	149	122	124	180	197	197	198	178	149	146	152	155	157	159	168
104	107	122	123	105	79	27	33	66	111	122	120	114	114	147	175	190	196	163	101	170	200	187	185	156	146	145	139	137	141	140	145
117	124	127	133	135	105	21	28	37	88	115	121	128	128	141	142	168	202	212	153	164	186	180	168	154	146	144	149	151	151	147	144
119	118	118	125	128	111	21	29	28	58	100	118	131	140	151	159	186	201	205	192	180	168	149	166	119	144	147	143	140	141	144	148
117	119	125	130	139	106	18	29	44	58	70	102	133	147	168	197	212	215	210	195	177	152	133	195	57	59	126	151	145	143	142	141
115	123	126	134	145	102	27	54	52	38	45	69	105	135	175	189	193	216	206	166	139	111	164	203	74	5	121	151	142	142	143	146
101	108	123	121	132	105	44	40	31	35	57	44	58	101	147	144	138	163	145	94	90	145	196	187	84	48	165	160	142	144	142	145
98	97	97	98	104	76	34	33	30	48	41	49	51	58	74	53	55	66	63	89	150	188	209	156	62	108	140	149	125	133	131	131
102	102	97	88	73	35	30	23	42	50	65	41	90	60	59	51	57	82	123	157	187	205	169	62	98	151	106	101	154	135	130	129

What computers see

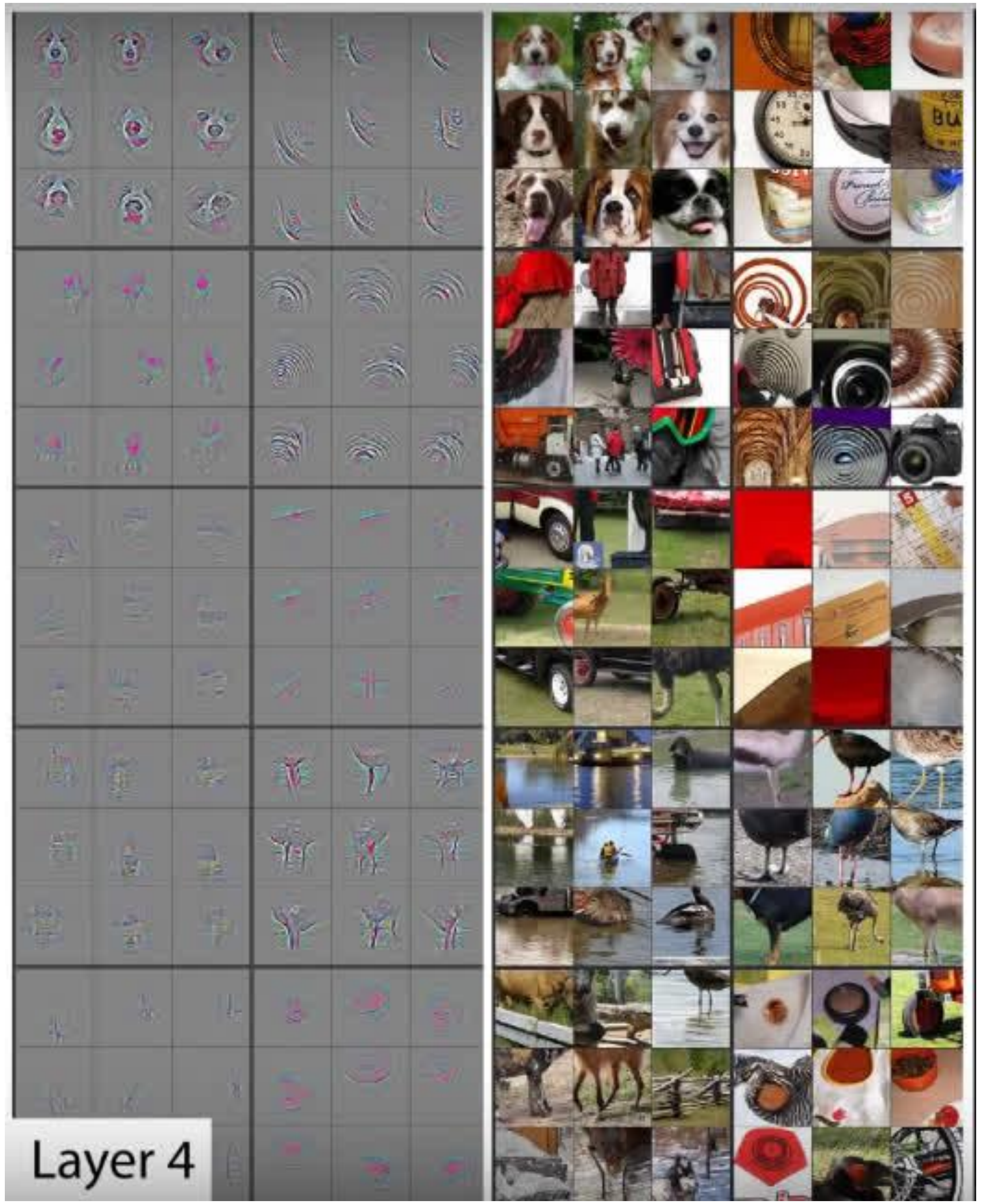
On the left, each square is a different feature map (convolutional filter) with the activations shown for the corresponding images on the right (these images are selected such that they maximize the activation of the filter.



Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833)

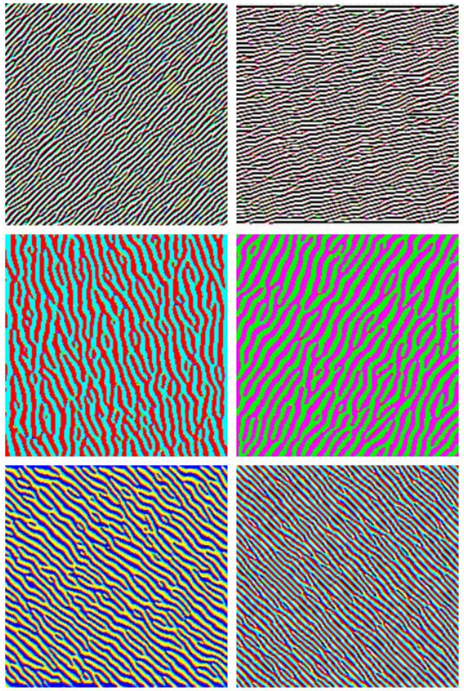
What computers see

The patterns that the filters detect become more complex as we go deeper into the network.

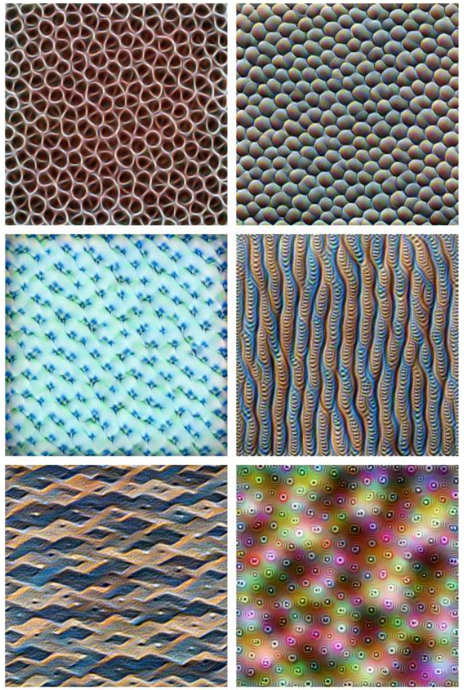


Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833)

What computers see

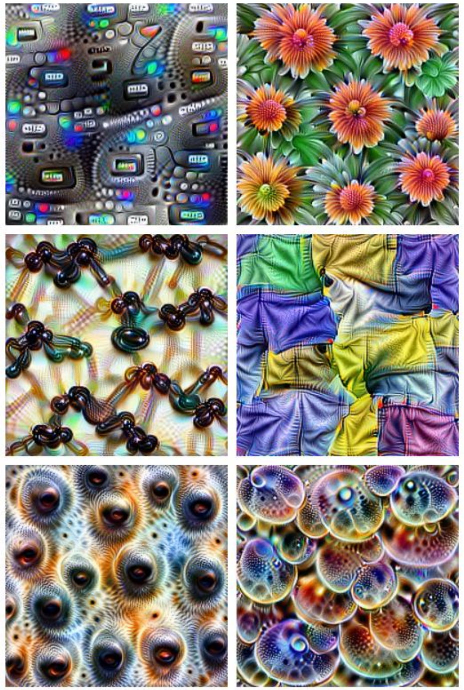


Edges (layer conv2d0)



Textures (layer mixed3a)

Patterns (layer mixed4a)

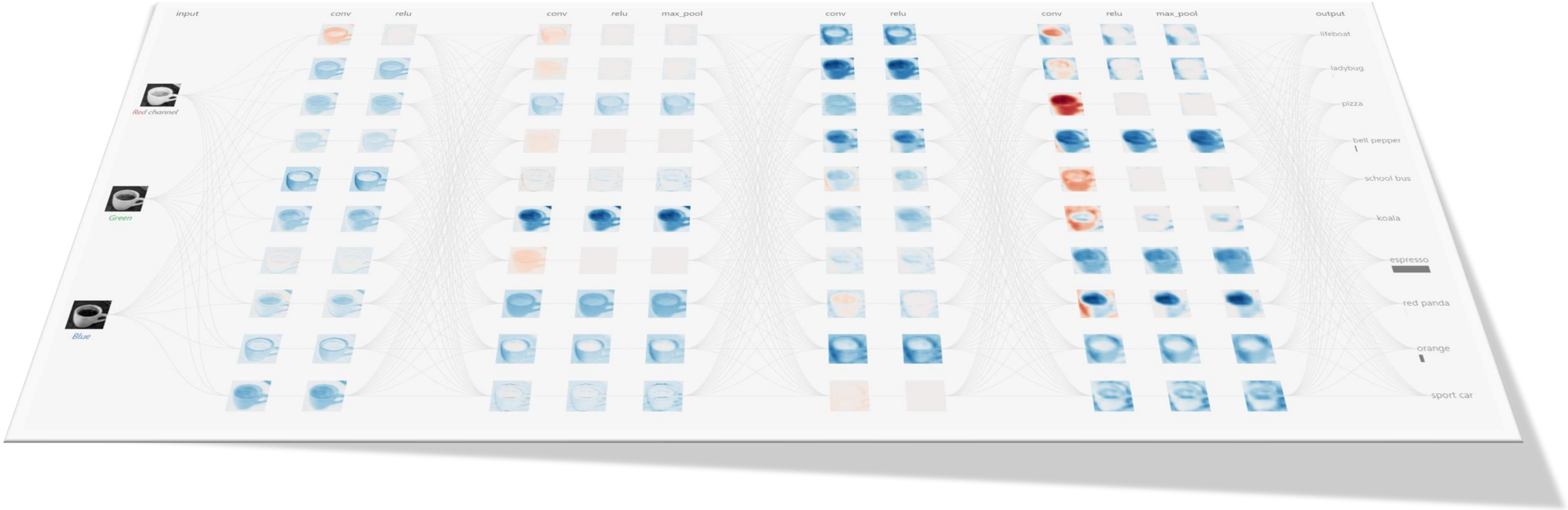


Parts (layers mixed4b & mixed4c)

Objects (layers mixed4d & mixed4e)

https://distill.pub/2017/feature-visualization/

What computers see



https://poloclub.github.io/cnn-explainer/

Activation maximization

Activation Maximization is a technique used to visualize what a CNN has learned by generating an input that **maximally activates** a specific neuron or filter.

It involves optimizing an input image using **gradient ascent** to enhance the activation of a target neuron, layer, or class score while applying regularization techniques (e.g., total variation loss, L2 norm) to ensure interpretability.

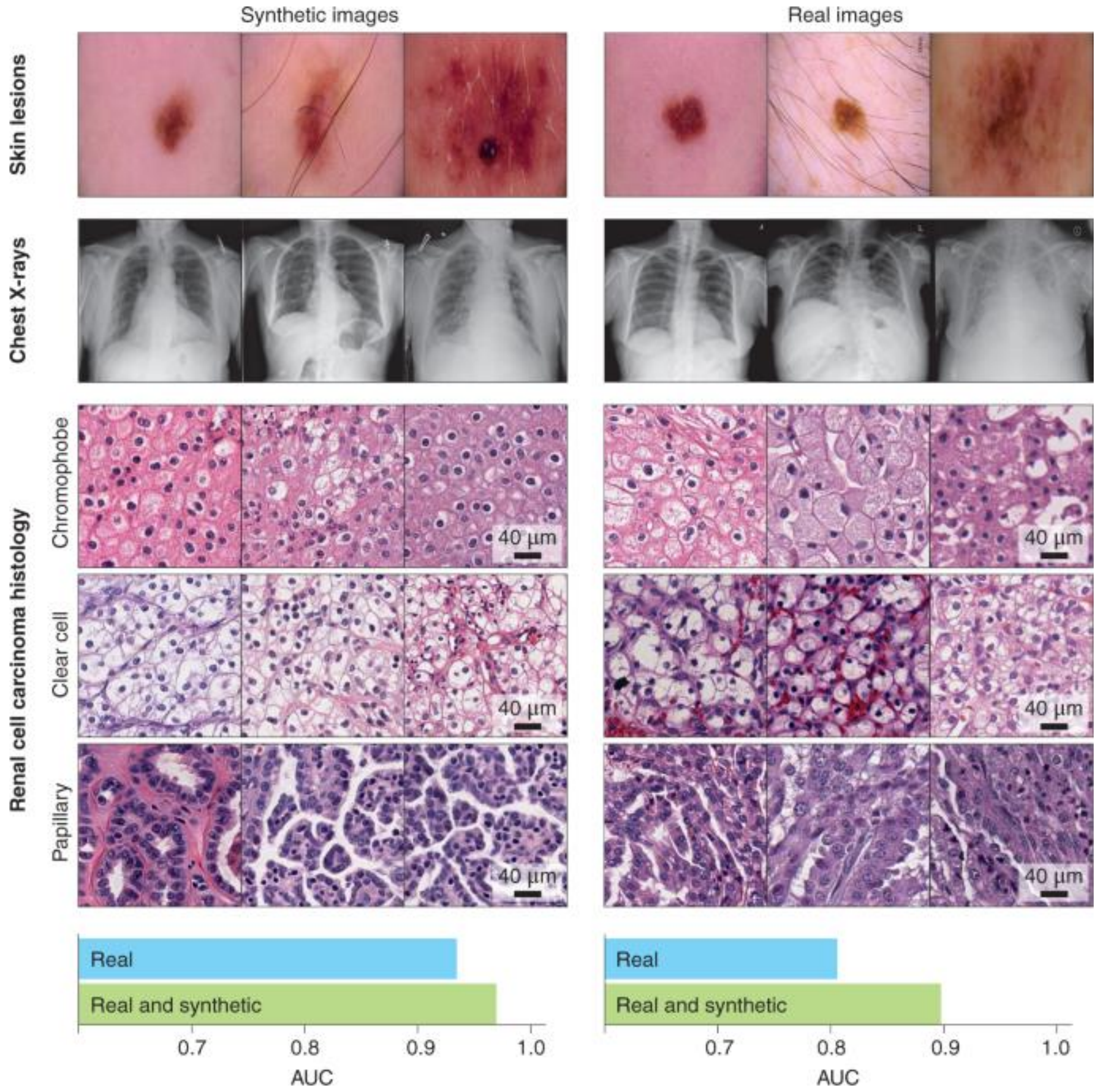
Activation maximization

Activation maximization

X-rays, MRIs, pathology slides.

By generating **synthetic** images that **maximally activate certain neurons**, we can visualize the features a model uses to detect tumors, fractures, or diseased tissues.

Generating synthetic images offers an ethical alternative to using **sensitive patient data**, helping with education and training without compromising patient privacy



Activation maximization

DNA/RNA sequences or protein structures: Generating **synthetic sequences** that maximize disease-associated neuron activations.

Optimizing molecular structures: **Chemical structures or molecular fingerprints**, e.g. revealing which molecular features trigger activation in neurons predicting drug activity.

Transfer learning

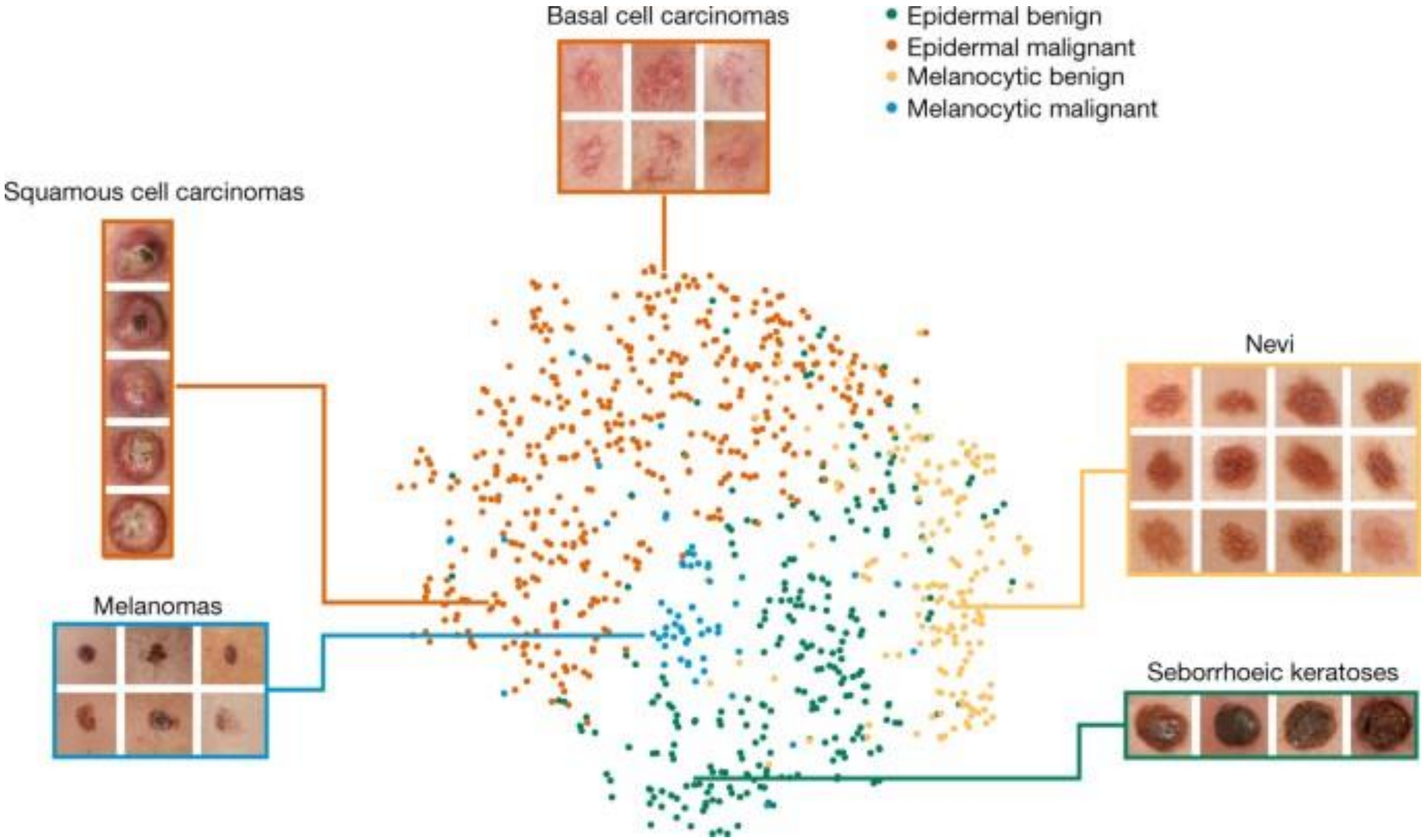
Transfer learning is a machine learning technique where a pre-trained model, trained on a large dataset (e.g., ImageNet for vision or BERT for NLP), is adapted to a new but related task with limited data..

It significantly **reduces training time** and computational costs by leveraging previously learned features instead of training a model from scratch.

Common approaches include **fine-tuning** (modifying and retraining some layers) and **feature extraction** (using pre-trained layers as fixed feature detectors).

Pre-training positions a deep neural network closer to the optimal solution in the vast, high-dimensional model parameter space, enabling faster convergence and better performance compared to starting from random initialization.

Transfer learning



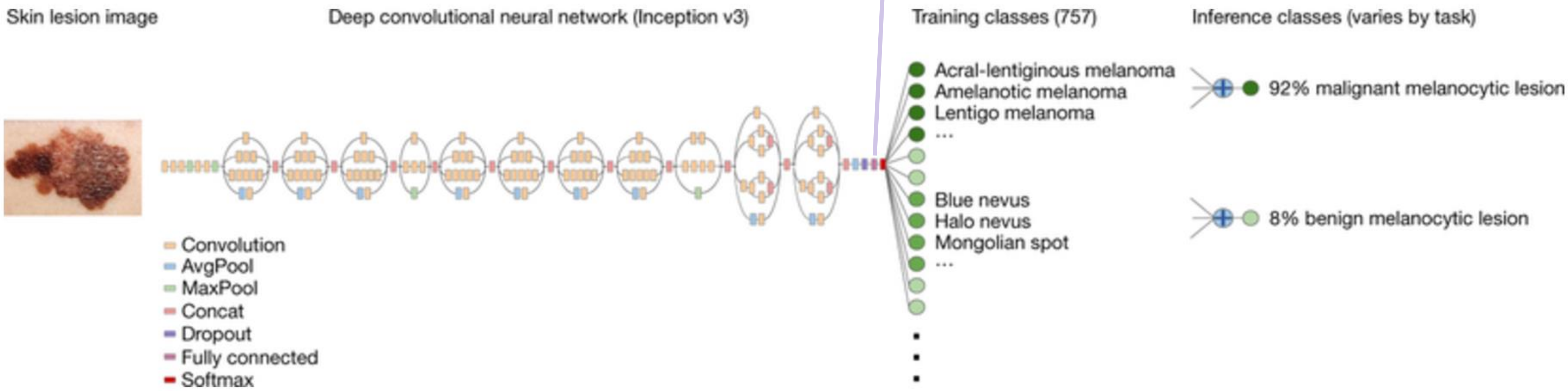
Thrun et al., Dermatologist-level classification of skin cancer with deep neural networks, Nature (2017)

Transfer learning

A **pre-trained** (on ImageNet) Inception v3 CNN was downloaded from GitHub.

The fully connected feed forward neural network module was replaced.

The network was then **fine-tuned** on the available lesion data. This means that the remainder of the modelparameters were used as the network's initialization.



Thrun et al., Dermatologist-level classification of skin cancer with deep neural networks, Nature (2017)