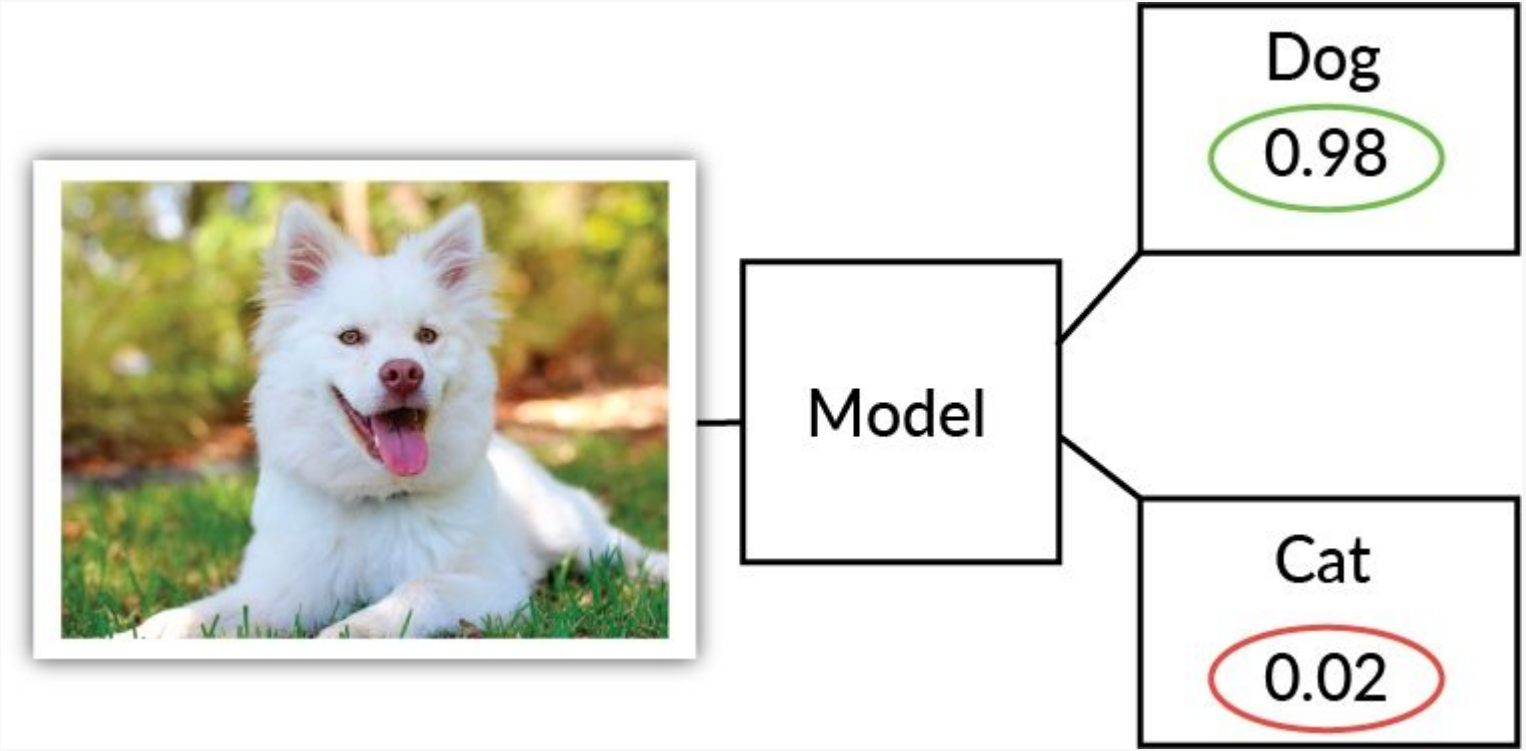
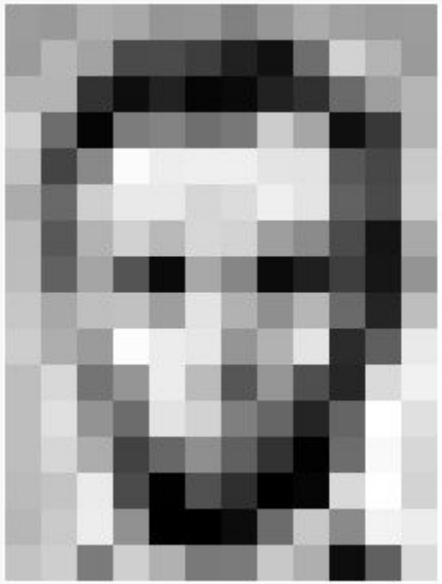
Module 4

Computer Vision - Working with Images



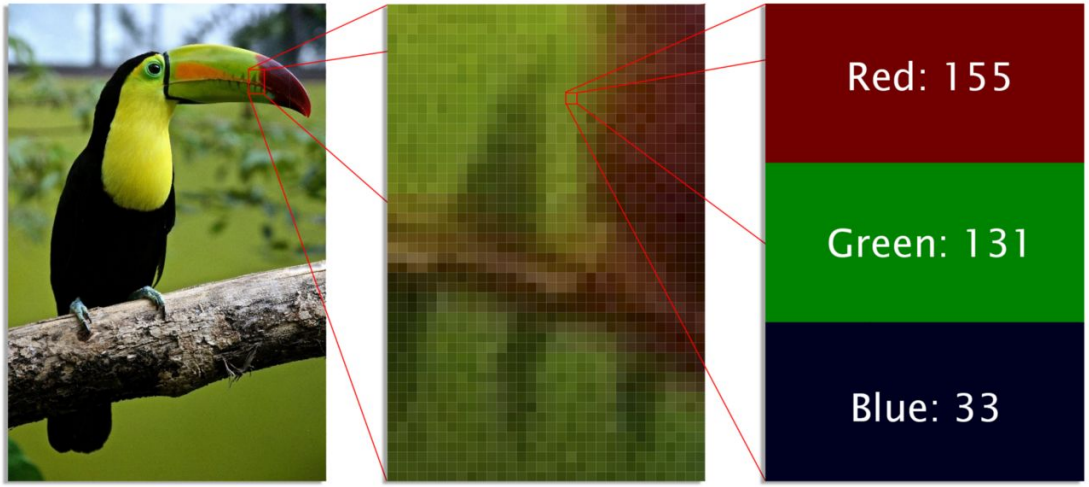
Working with images

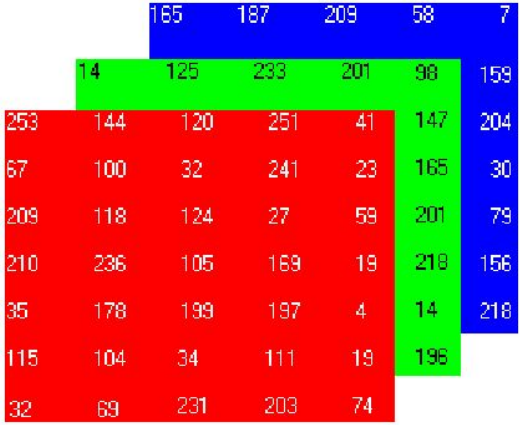


157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Color channels





Let’s review Keras sequential models

Core concepts:

train_test_split

Neurons

Layers (Dense)

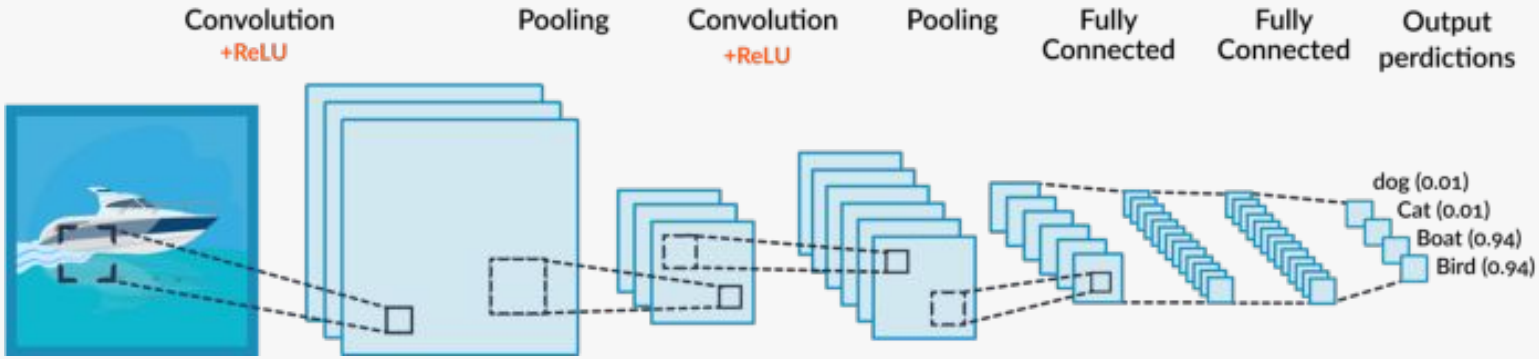
Activation functions (Sigmoid, ReLu)

Loss Functions (MSE, Negative log loss, Cross entropy)

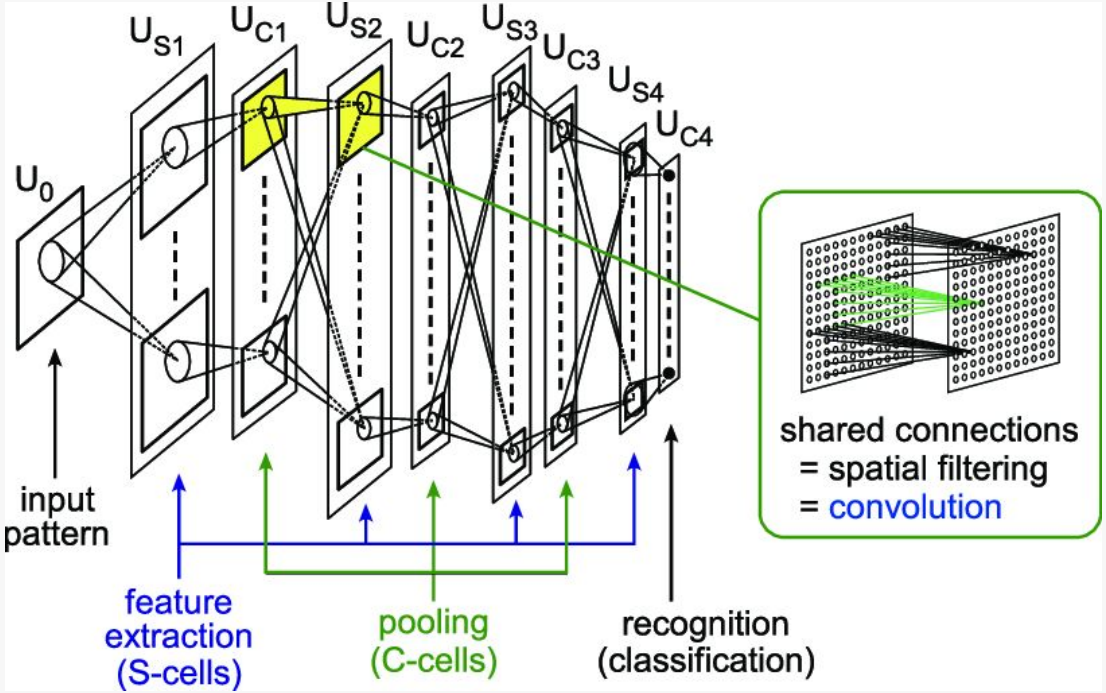
Optimizers (Adam)

Training (Epochs, Batchsize)

Convolutional Neural Networks

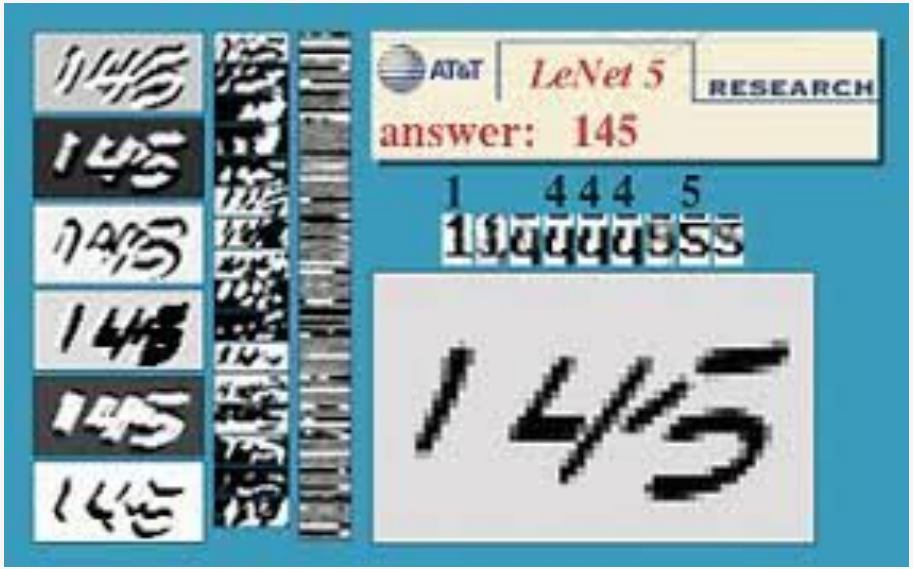


Sharing weights



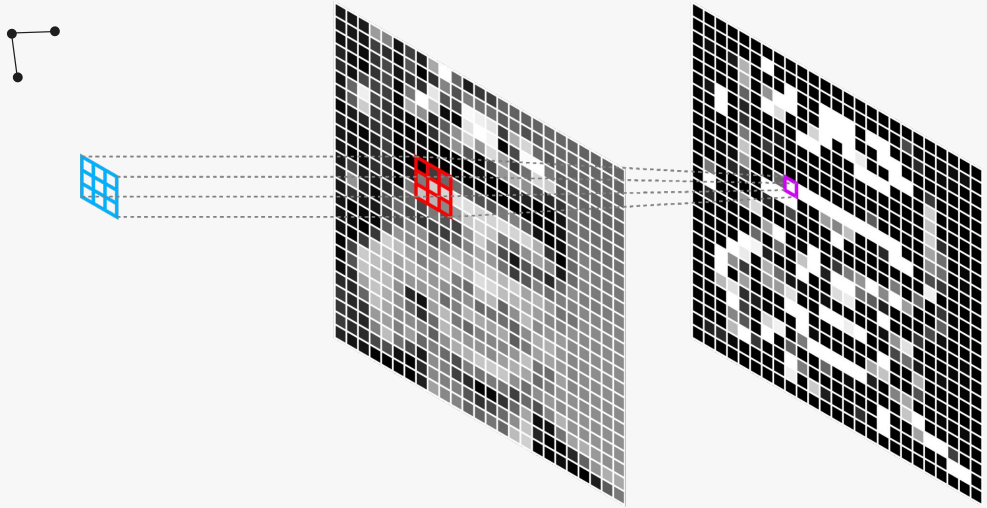
Fukushima and NeoCognitron

The LeNet model



LeNet model demo - Bell Labs Youtube Video

Convolutional Neural Networks



-1	-1	-1
-1	8	-1
-1	-1	-1

Kernel

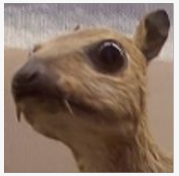
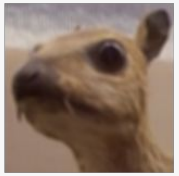
45	81	87
194	203	215
164	116	131

Input

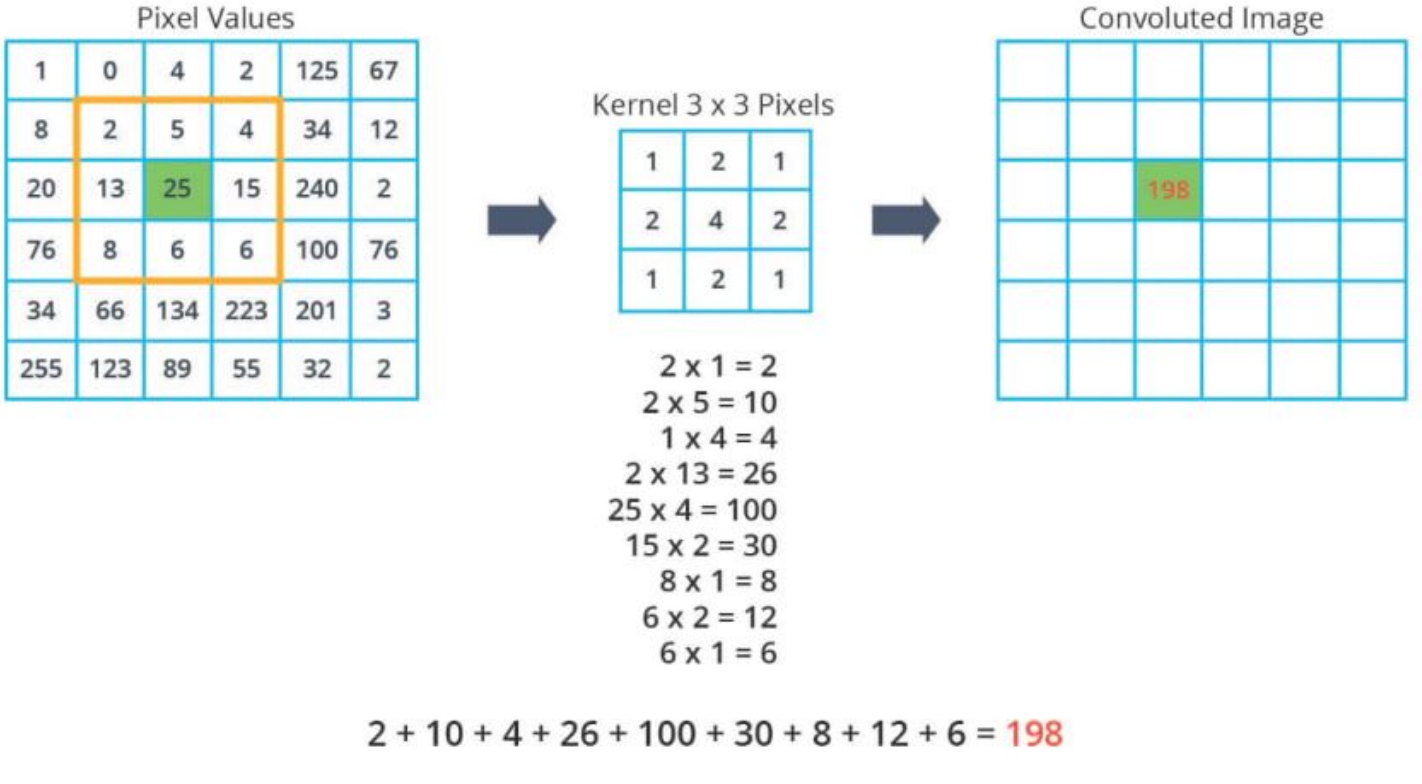
255*

* = max(255, 657)

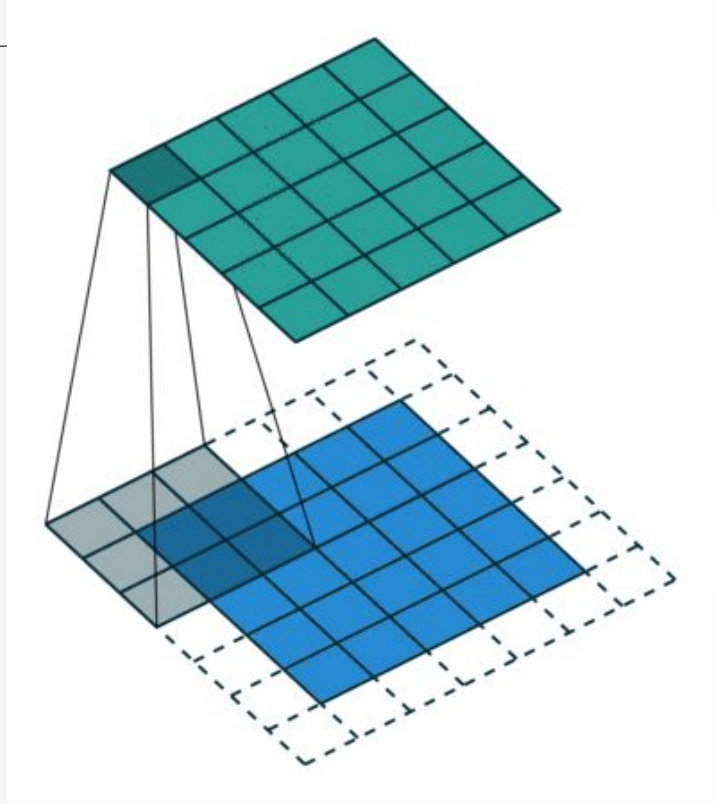
Output

Input	Convolution filter	Feature
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	 Edge
	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	 Blurred
	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	 Sharpen

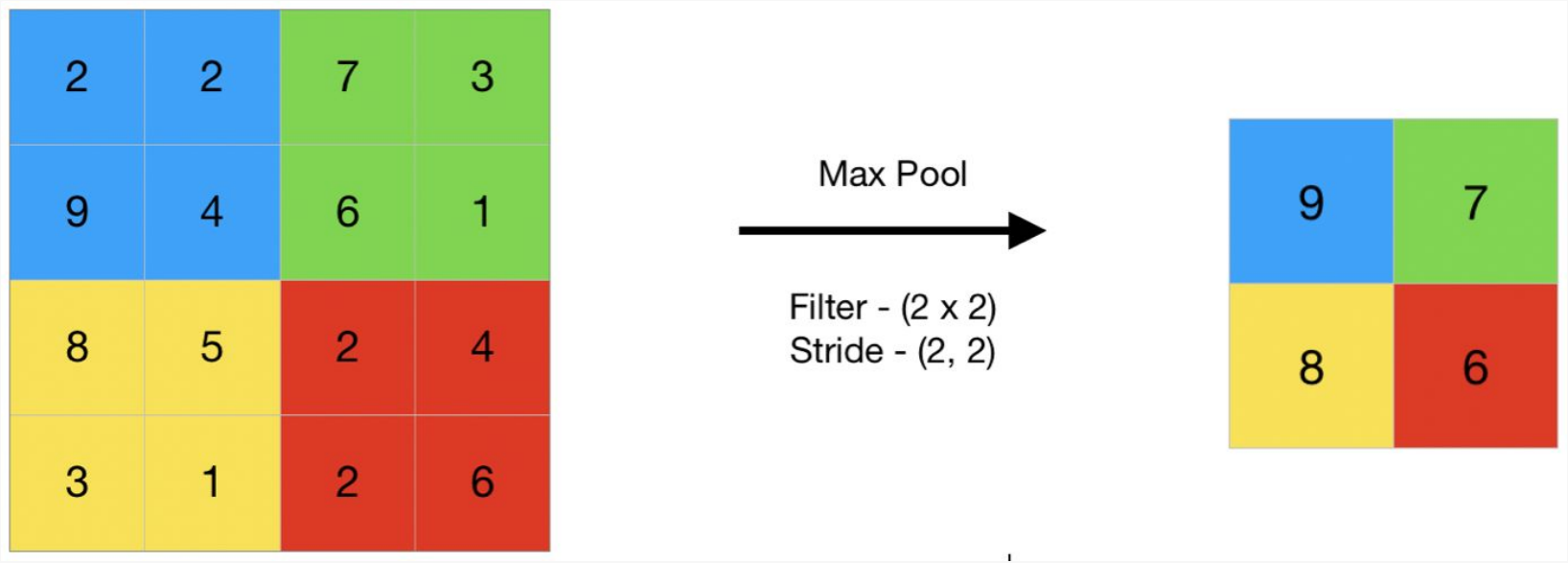
Convolutional Neural Networks



Sliding a kernel over a padded image



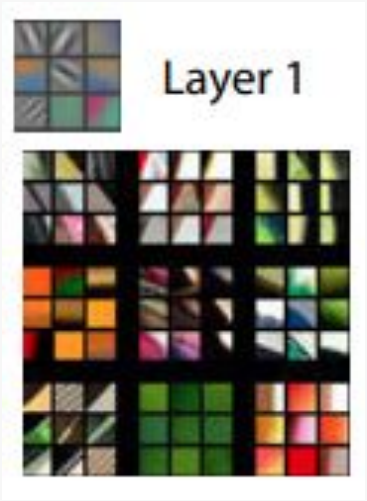
Pooling layer

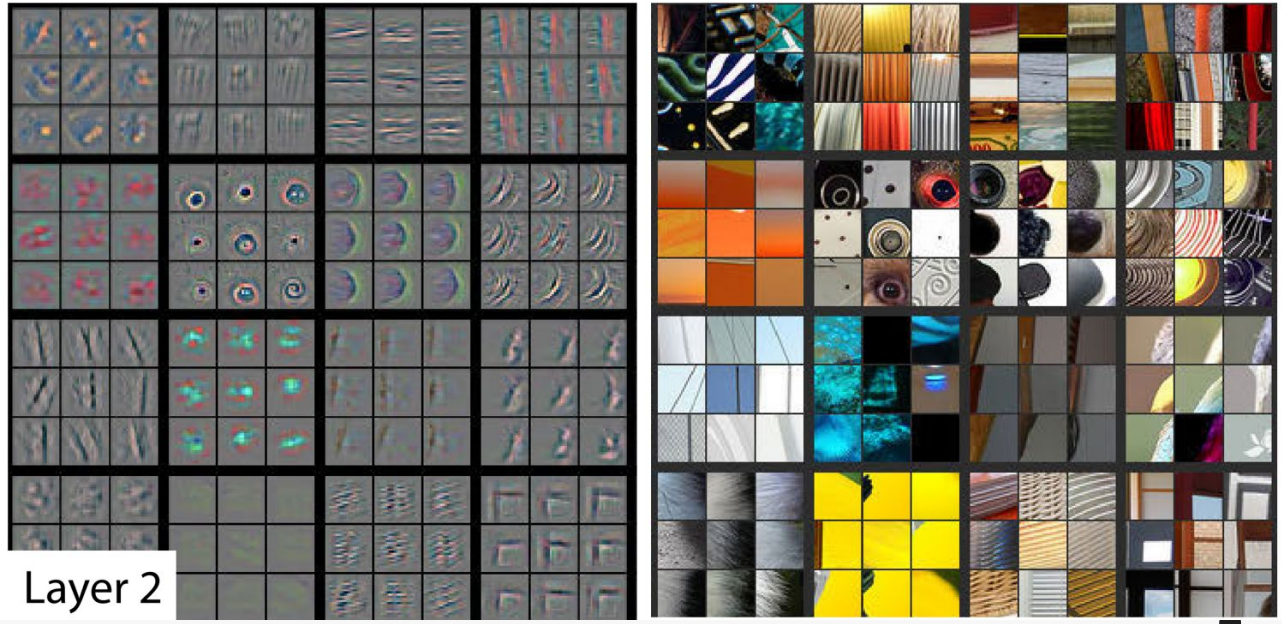


3D demo of NN Architectures

Video link: https://youtu.be/3JQ3hYko51Y

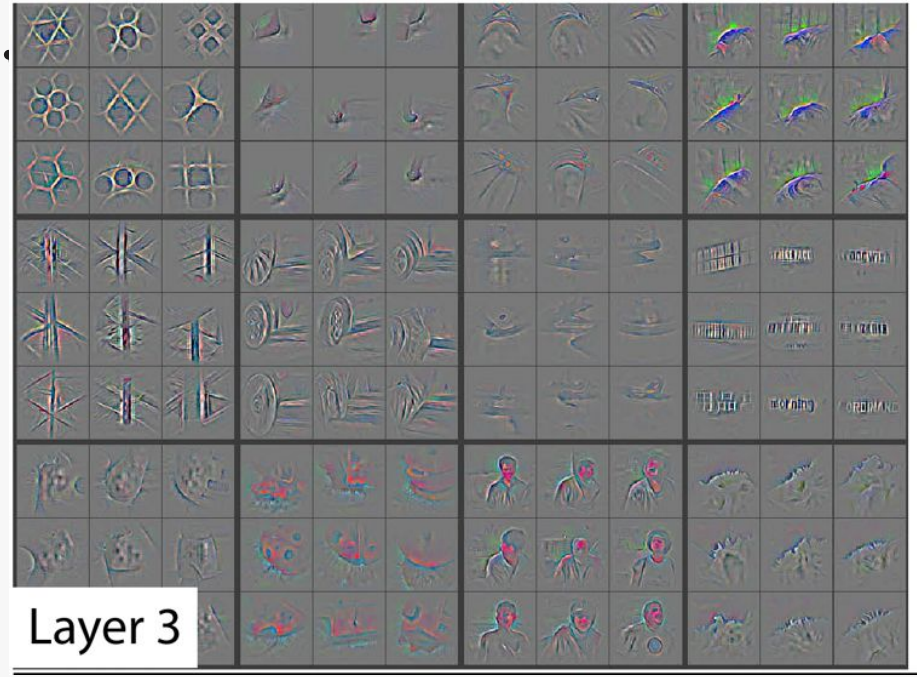
What do the CNN Layers Learn 1&2



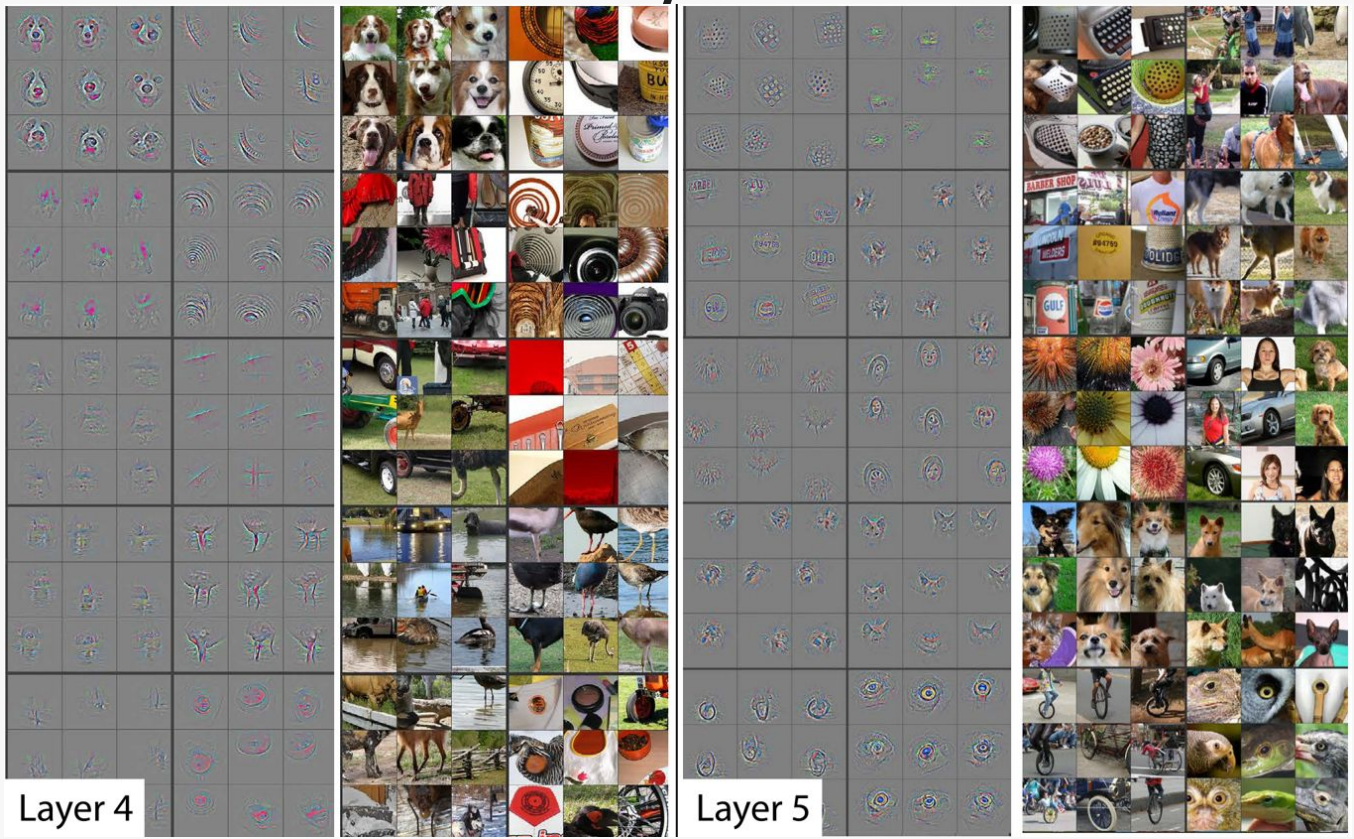


Visualizing and Understanding Convolutional Networks - Zeiler et al.

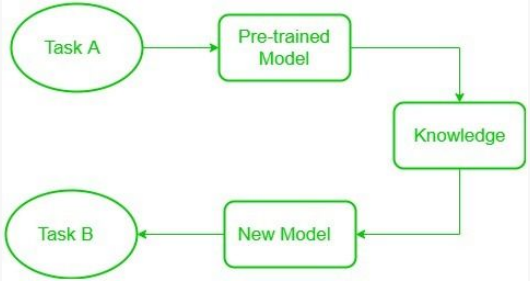
What do the CNN Layers Learn

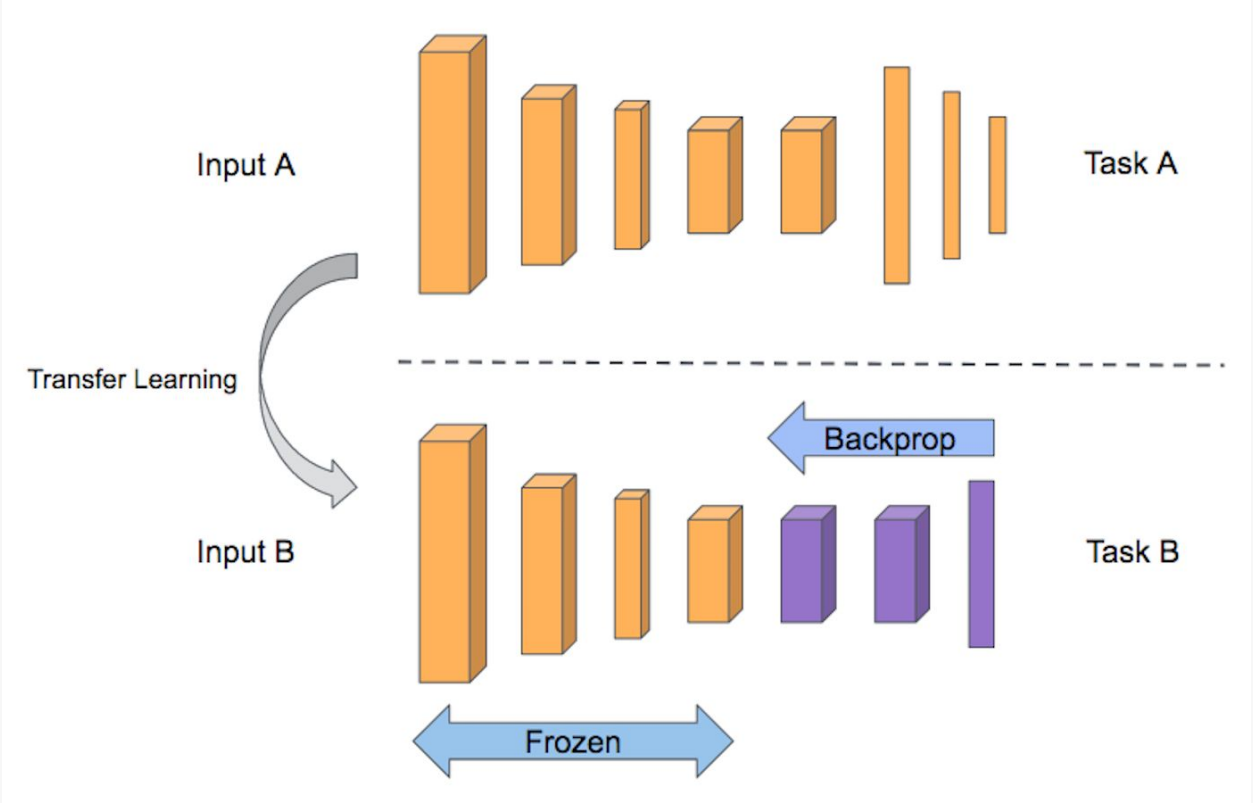


What do the CNN Layers Learn



Transfer Learning





Pre-trained models

RESNET 50

ALEXNET

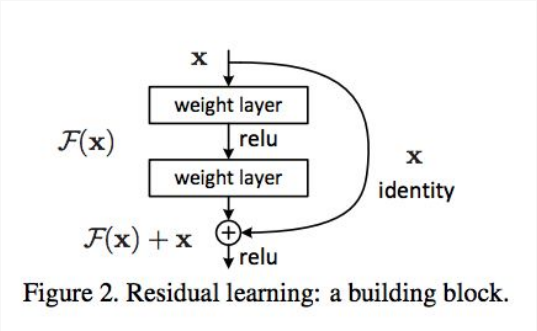
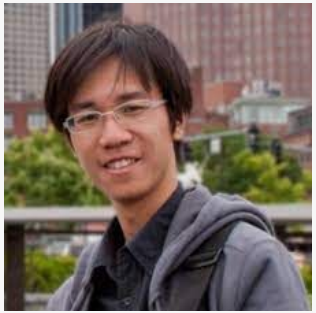


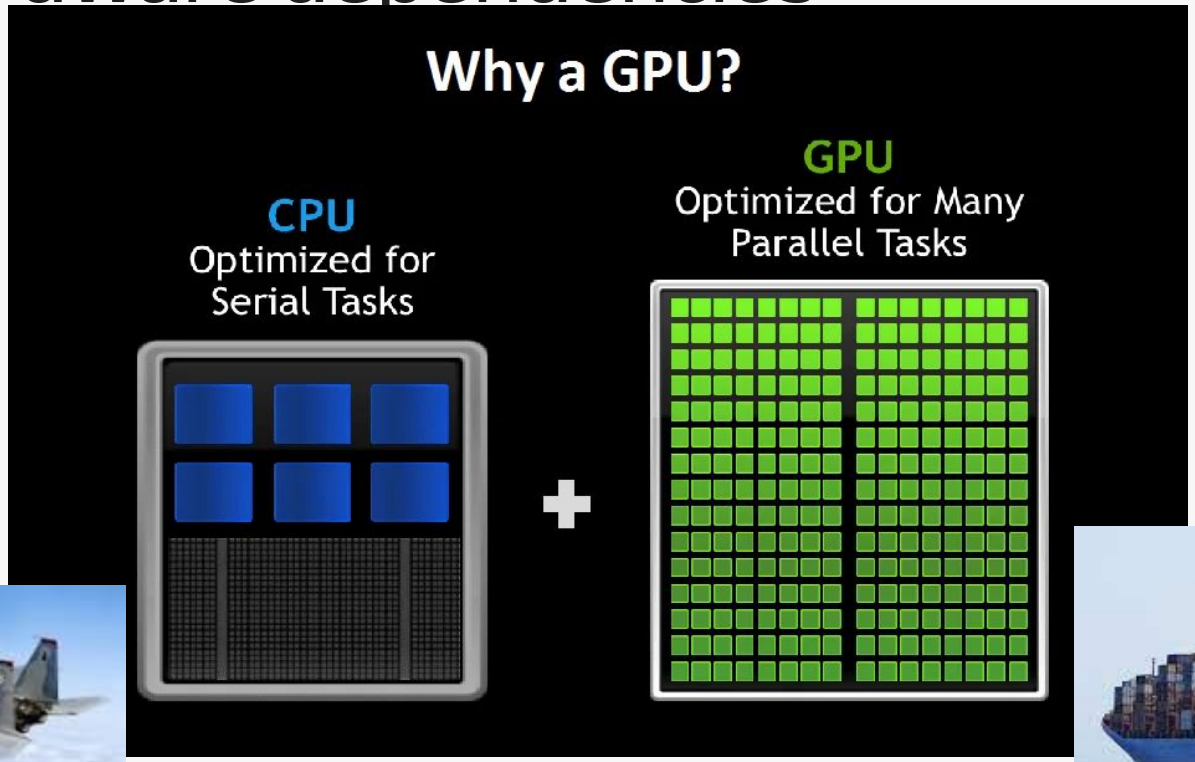
Figure 2. Residual learning: a building block.

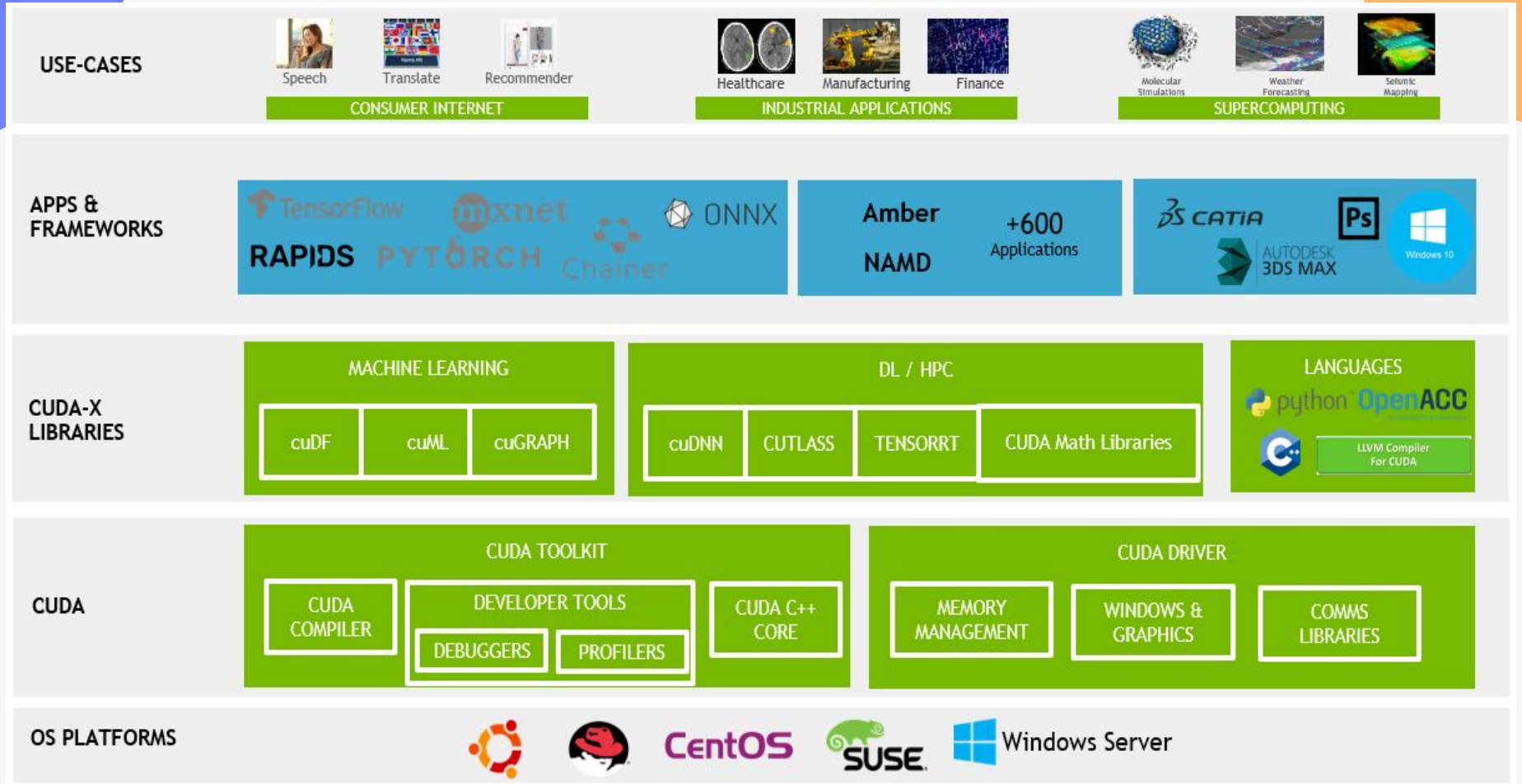


Kaiming He

Annual Imagenet competition

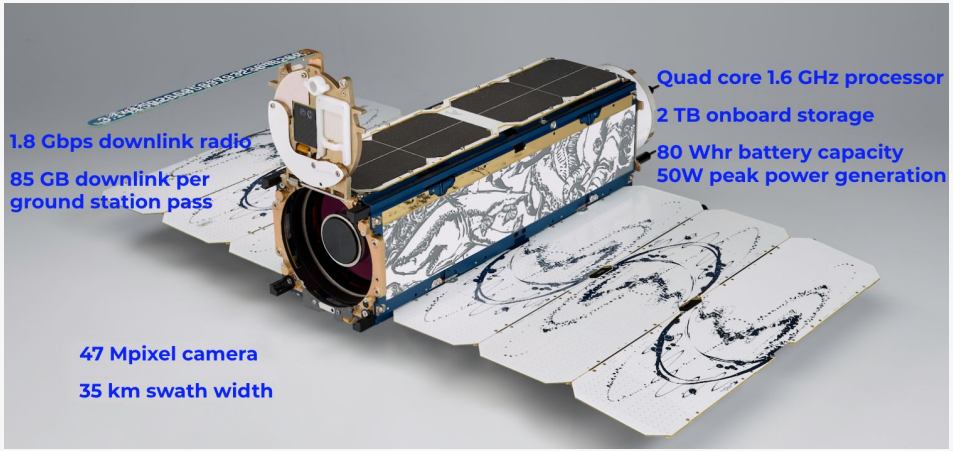
Hardware dependencies

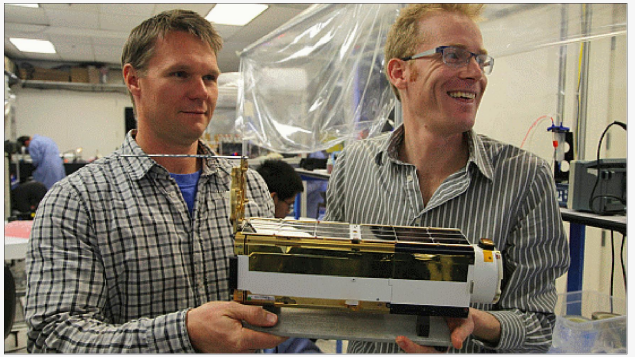



USE-CASES
Speech
Translate
Recommender
CONSUMER INTERNET
Healthcare
Manufacturing
Finance
INDUSTRIAL APPLICATIONS
Molecular Simulations
Weather Forecasting
Seismic Mapping
SUPERCOMPUTING
APPS & FRAMEWORKS
TensorFlow
mxnet
ONNX
RAPIDS
PYTORCH
Chainer
Amber
NAMD
+600 Applications
CATIA
Ps
AUTODESK 3DS MAX
Windows 10
CUDA-X LIBRARIES
MACHINE LEARNING
cuDF
cuML
cuGRAPH
DL / HPC
cuDNN
CUTLASS
TENSORRT
CUDA Math Libraries
LANGUAGES
python
OpenACC
LLVM Compiler For CUDA
CUDA
CUDA TOOLKIT
CUDA COMPILER
DEVELOPER TOOLS
DEBUGGERS
PROFILERS
CUDA C++ CORE
CUDA DRIVER
MEMORY MANAGEMENT
WINDOWS & GRAPHICS
COMMS LIBRARIES
OS PLATFORMS
CentOS
SUSE
Windows Server

Nvidia CUDA

Transfer learning CNN Pytorch - hands on





Satellite image classification task.