Jan, 13th - 14th 2020

Neosperience
Empathy in Technology

—

Computer Vision Applications

Who am I?

Luca Bianchi

Chief Technology Officer @ **Neosperience**

Chief Technology Officer @ **WizKey**

Serverless Meetup Italy manager

ServerlessDays co-organizer

github.com/aletheia

https://it.linkedin.com/in/lucabianchipavia

@bianchiluca

https://speakerdeck.com/aletheia

Neosperience

Who am I?

Alessandro Bitetto

PhD student @ **FinTech Lab Pavia**

PhD intern @ **Neosperience**

https://www.linkedin.com/in/alessandrobitetto

Neosperience

Neosperience Cloud

WE ARE HIRING

A cloud platform built on AWS to deliver DCX projects

Neosperience Cloud is the technology platform that allows creating personalized experiences for your customers that drive loyalty and faster paths to purchase. Unlike existing technologies that rely only on demographics data, we use proprietary models, developed with AI, to personalize your offering to the right segment. A compelling experience for each customer at the right time, place, and situational context.

- Deeply understand their customers and be more useful to them by delivering relevant digital experiences.
- Delight customers by delivering relevant experiences across mobile, web, in-store.
- Maintain their Brand identity and increase value as platforms like Amazon, Google and Facebook drive up disintermediation and make companies unintentional utilities.
- Keep pace with the variety of devices and interaction models available to customers to overcome complexity and costs associated with the alignment of apps, web apps, social media and conversational interfaces.

…which means fast time to market, machine learning and scalability by design.

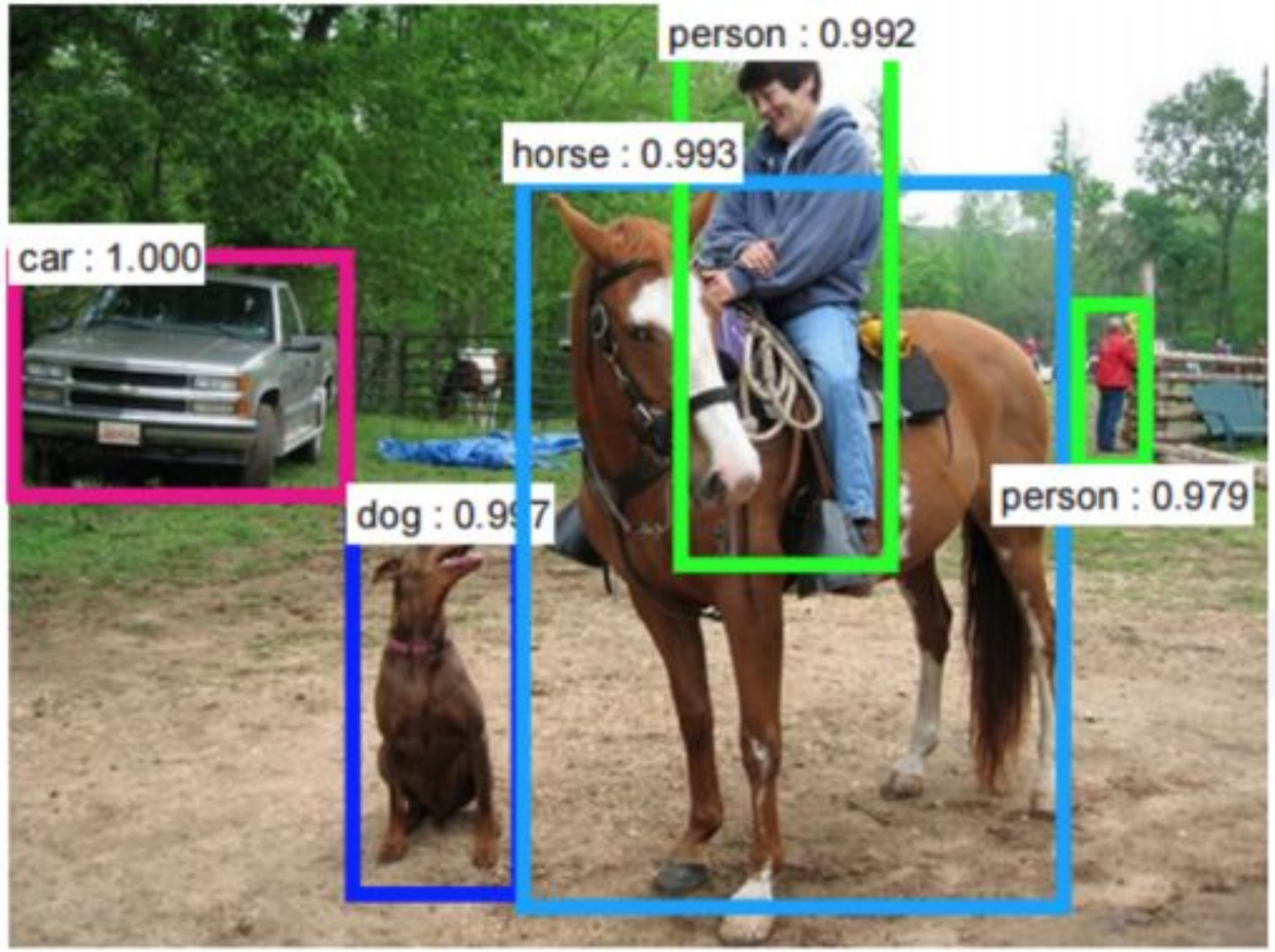
Part 1

Convolutional Neural Networks (CNNs)

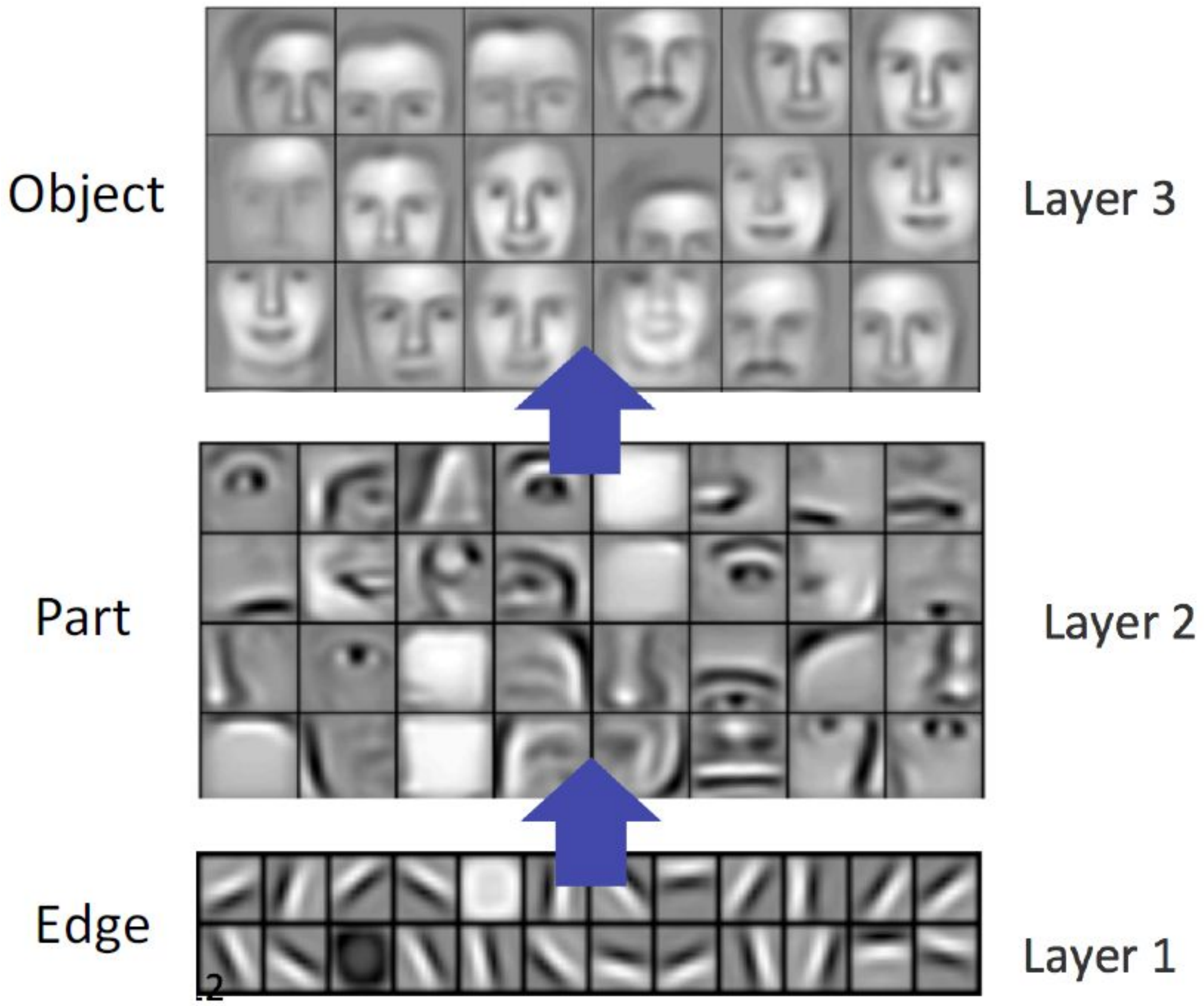
Neosperience

Convolutional Neural Network

Convolutional Neural Networks (ConvNets or CNNs) are a category of Neural Networks that have proven very effective in areas such as image recognition and classification.

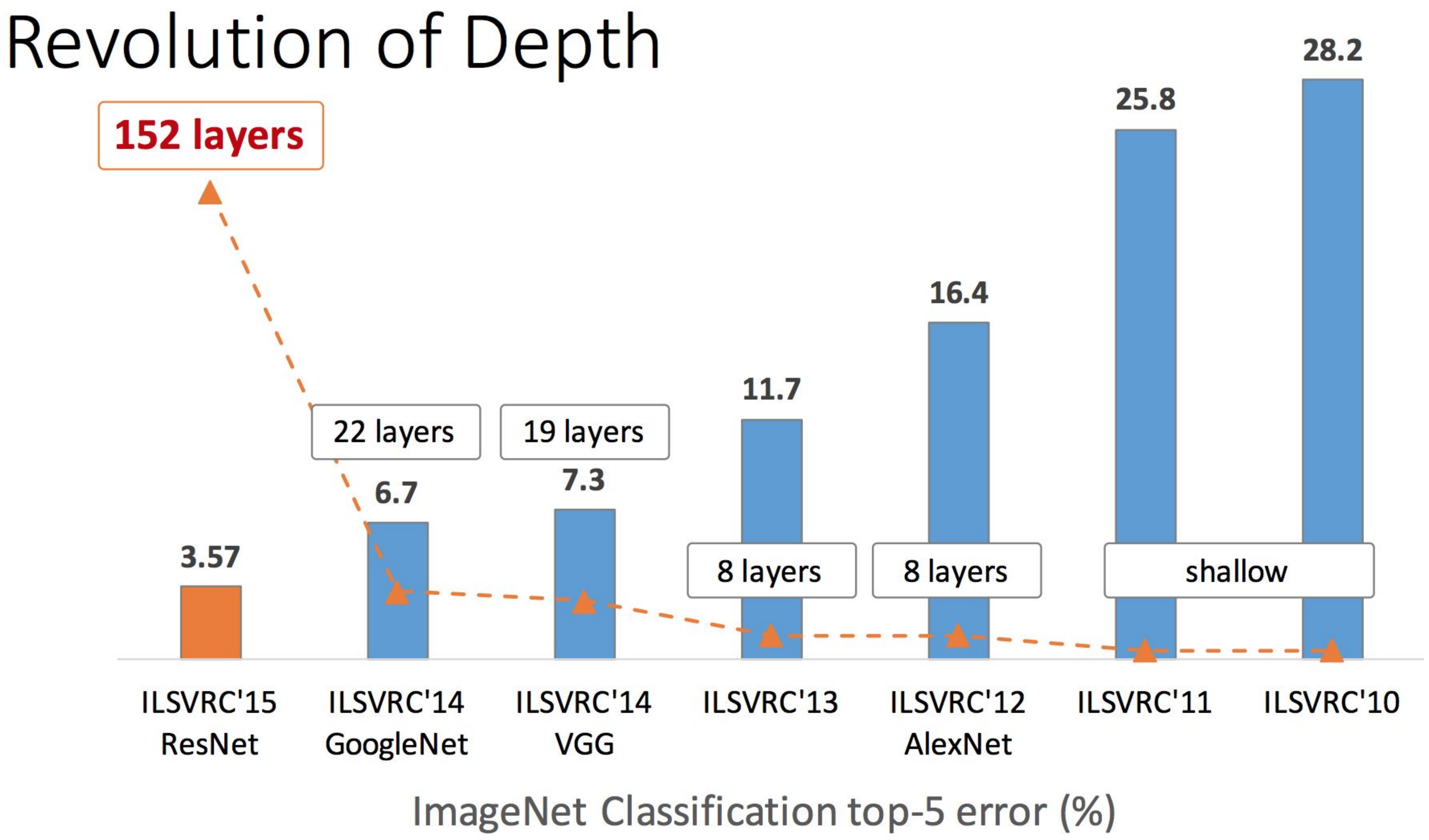


CNNs are based on **Hierarchical Compositionality**: we start from a low level input (pixel) and then we aggregate informations up to an higher interpretation level.



Convolutional Neural Network

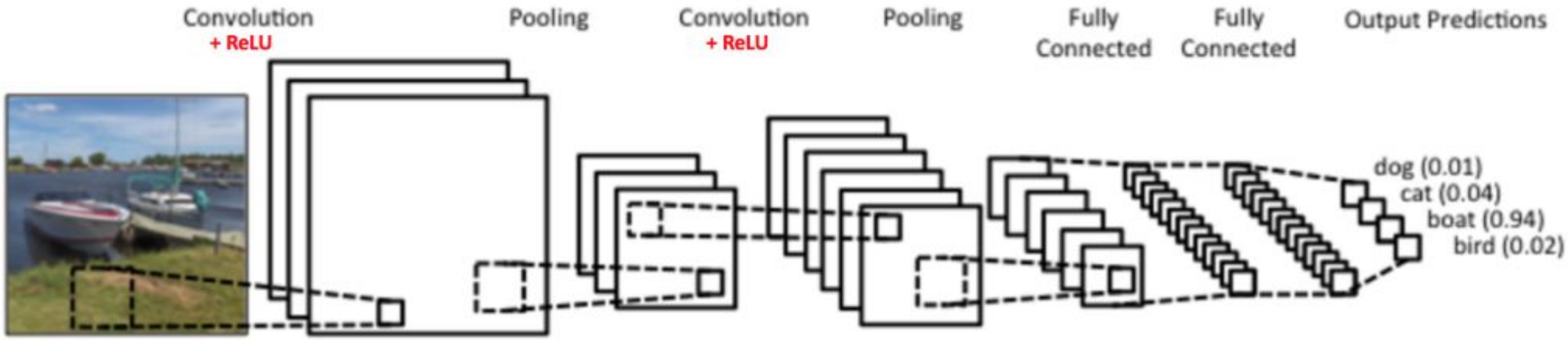
First CNN was developed by Yann LeCun on 1988, called **LeNet**, but CNNs became popular when in 2012 **AlexNet** was the first CNN to win the ImageNet Large Scale Visual Recognition Challenge (ILSVCR). Since then, only DNN model where used (and won) the following editions.



Convolutional Neural Network

Key components of a CNN are the following:

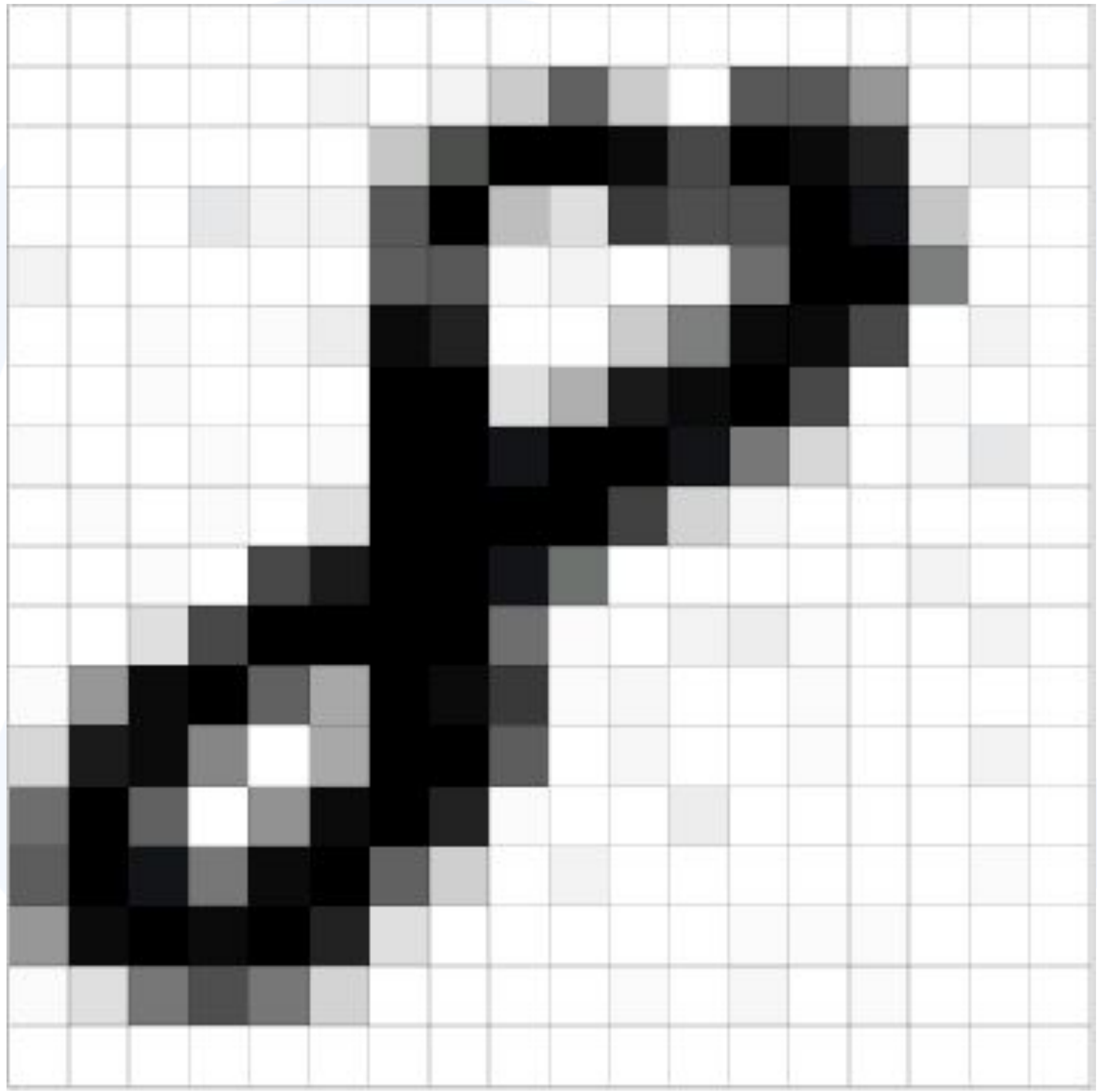
- Convolution
- Non Linearity (activation function)
- Pooling or Sub-sampling
- Classification (fully connected layer) and training

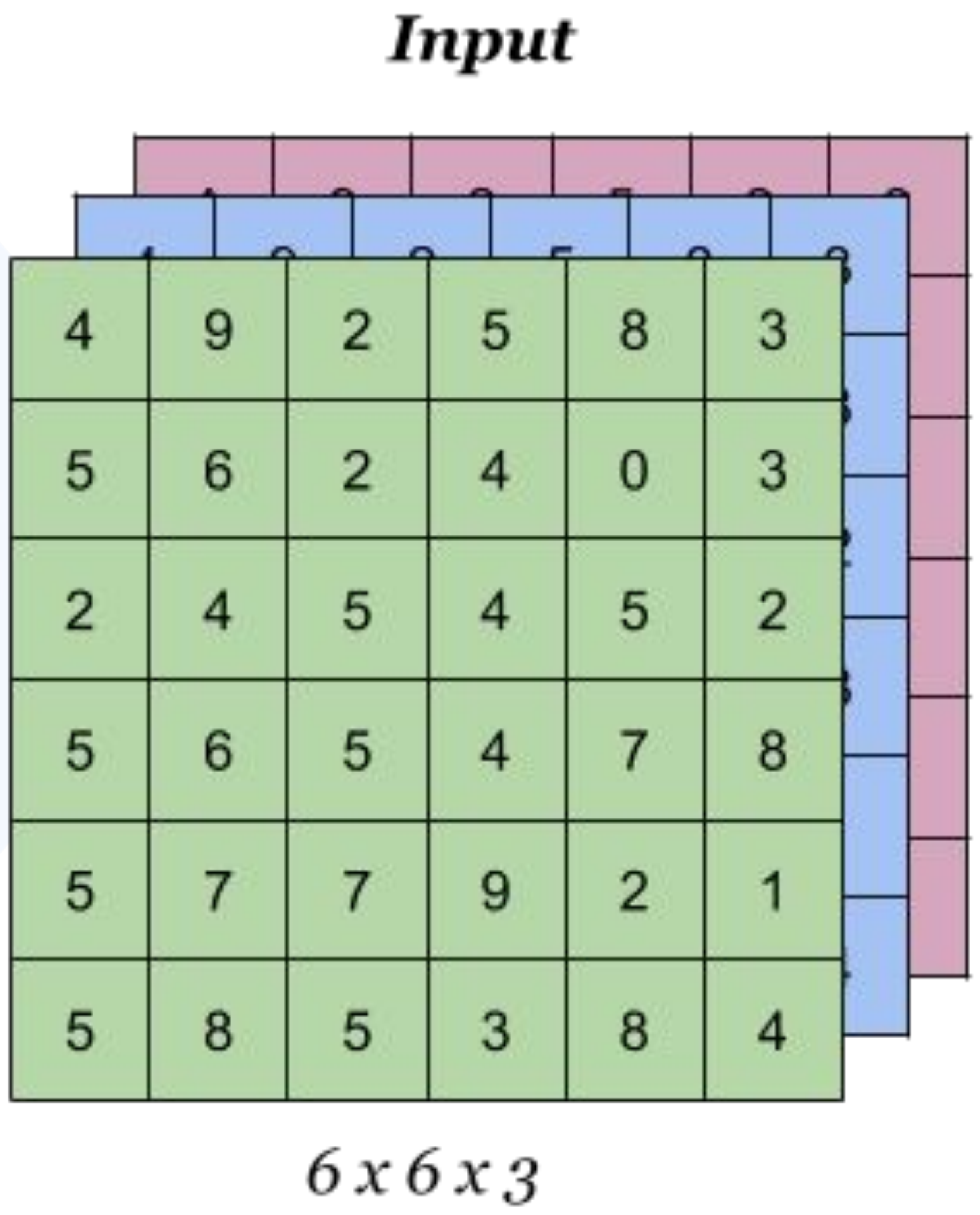


LeNet

Input

Every image can be represented as matrices of pixels, one for each channel (RGB, HSV, etc)





Convolution filter

We chose a **filter** (or **Kernel**) to be passed on the image. Every cell of the filter is multiplied elementwise with the corresponding area of each channel and then summed up. Outcome is called **Convolved Feature** or **Feature Map**

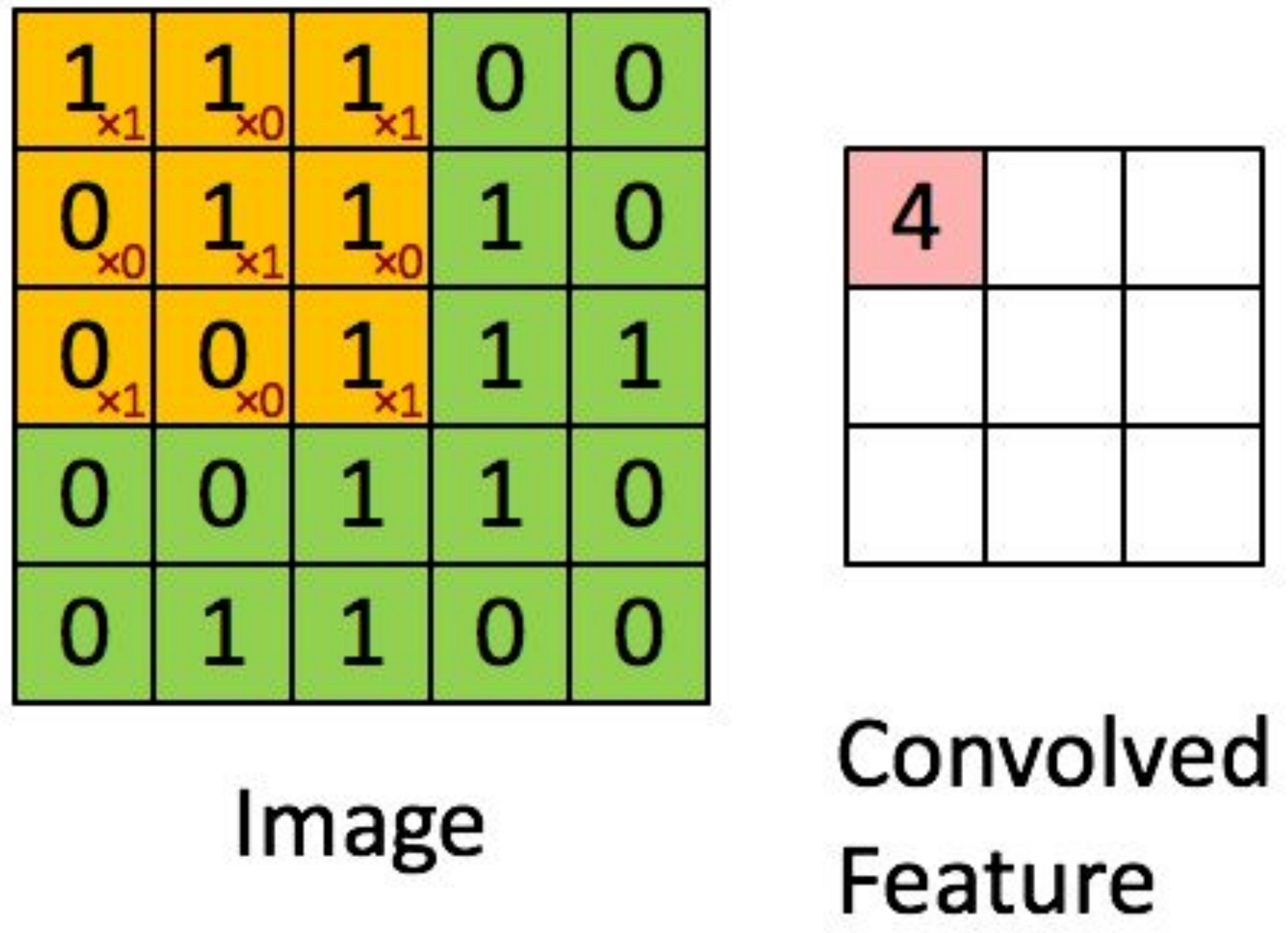
Input

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

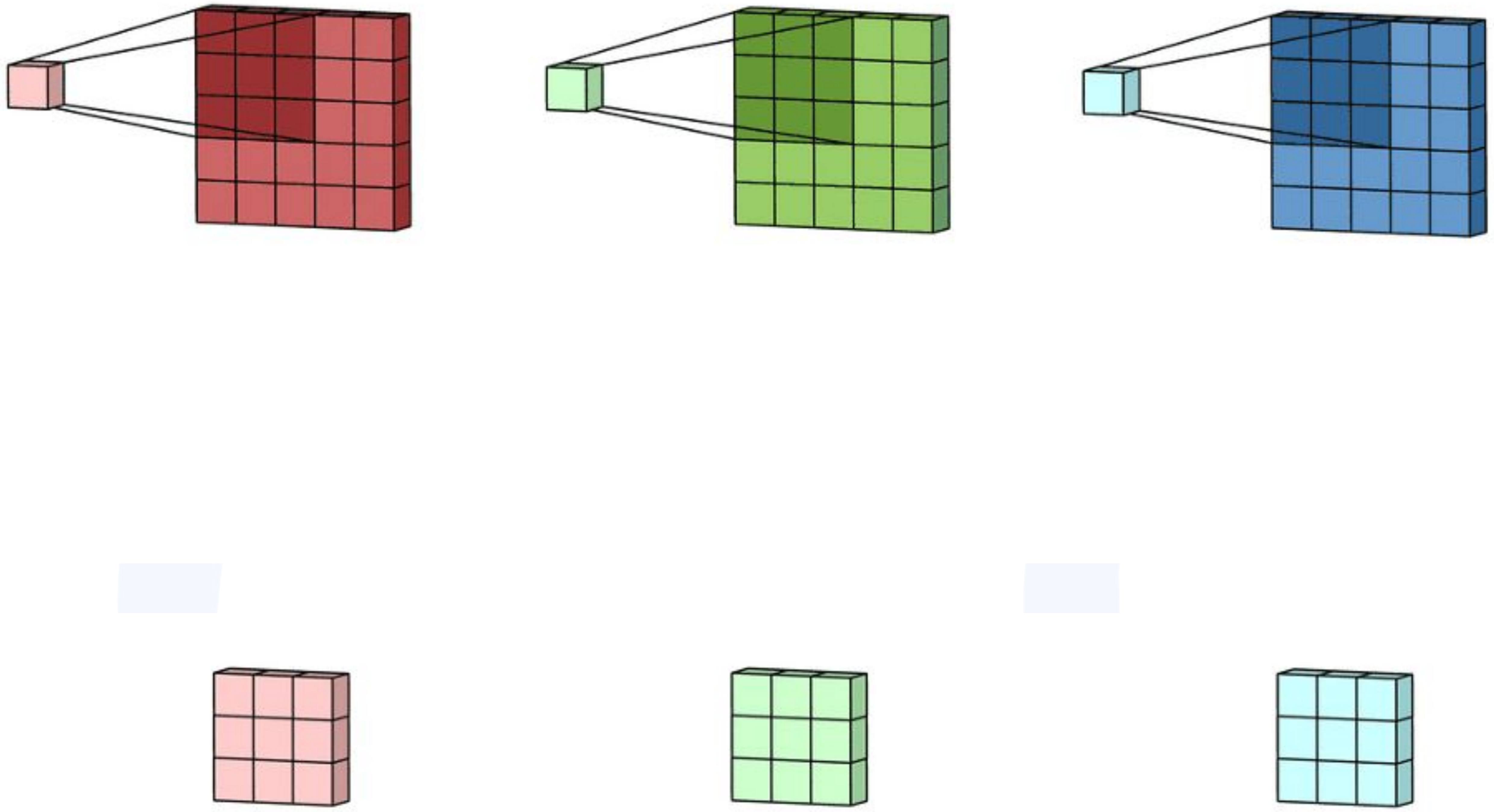
Filter

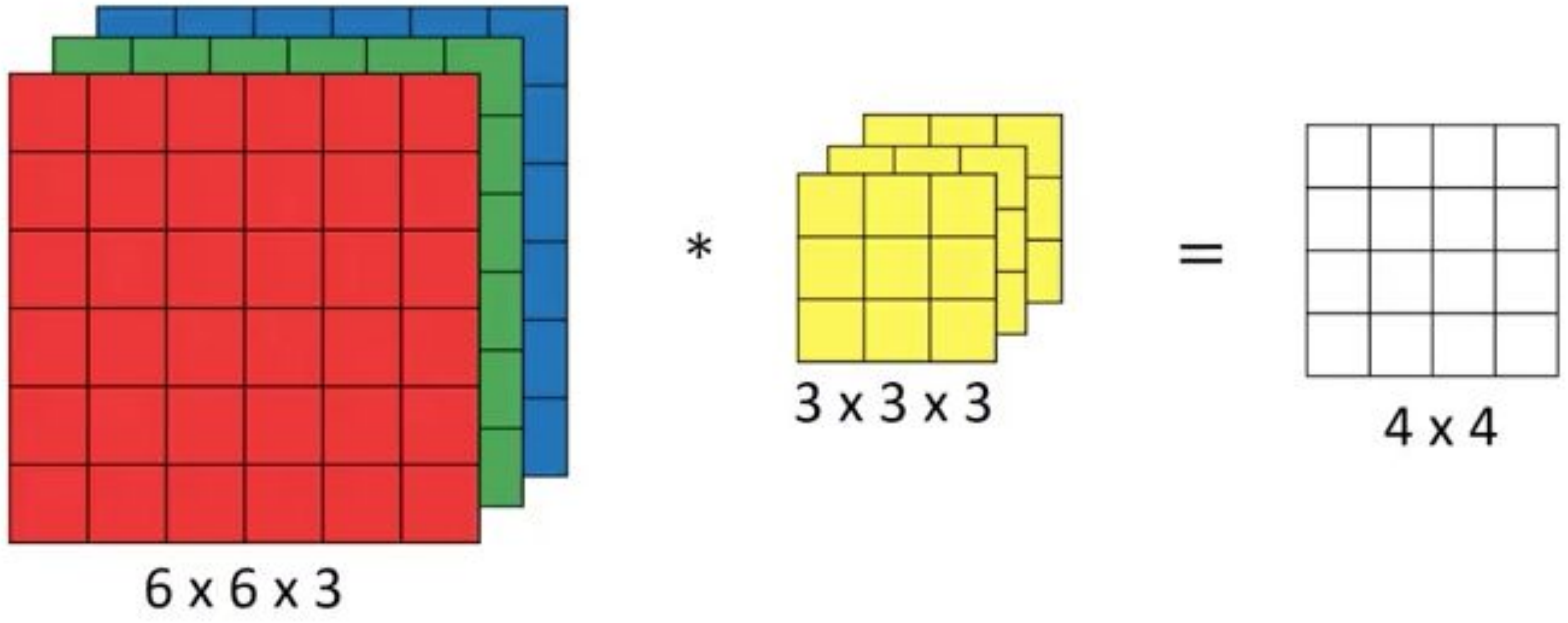
1	0	1
0	1	0
1	0	1

Convolution filter



Convolution filter - 3 channel example

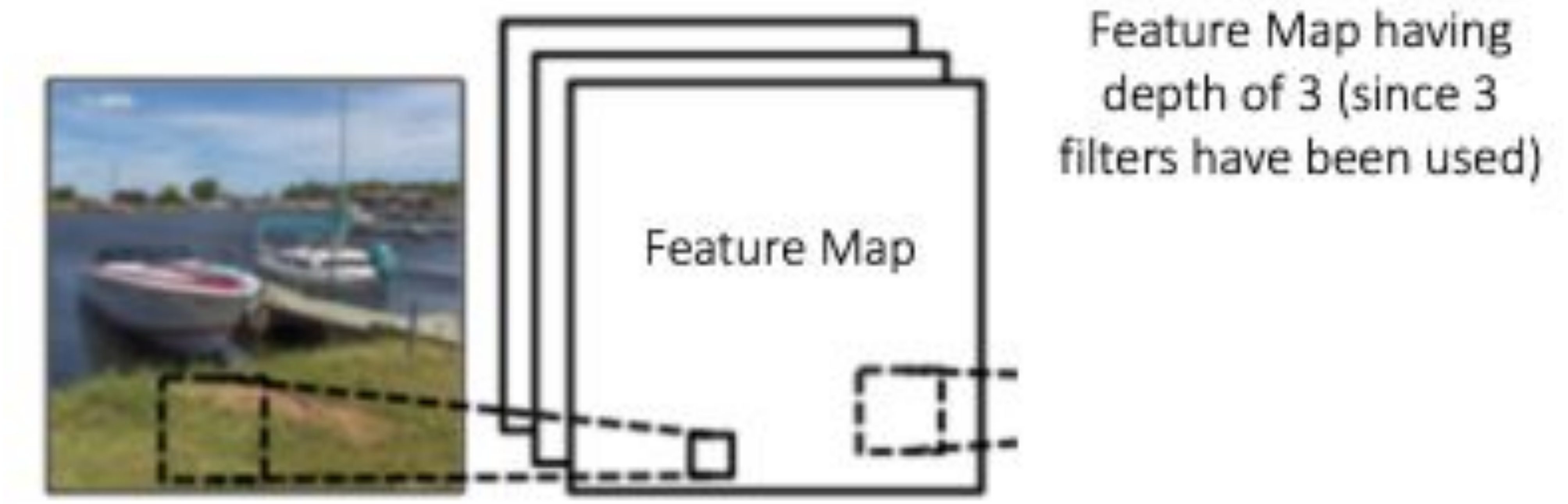


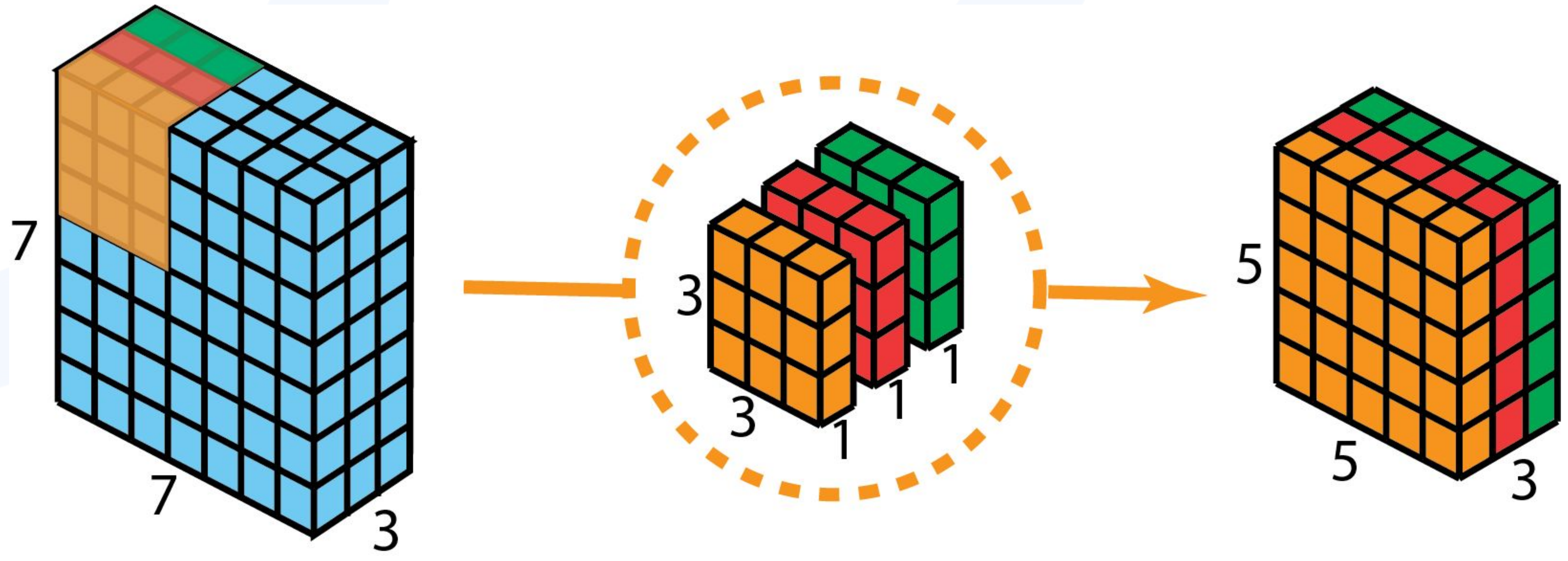


Convolution filter parameters

Each filter is characterized by the following parameters:

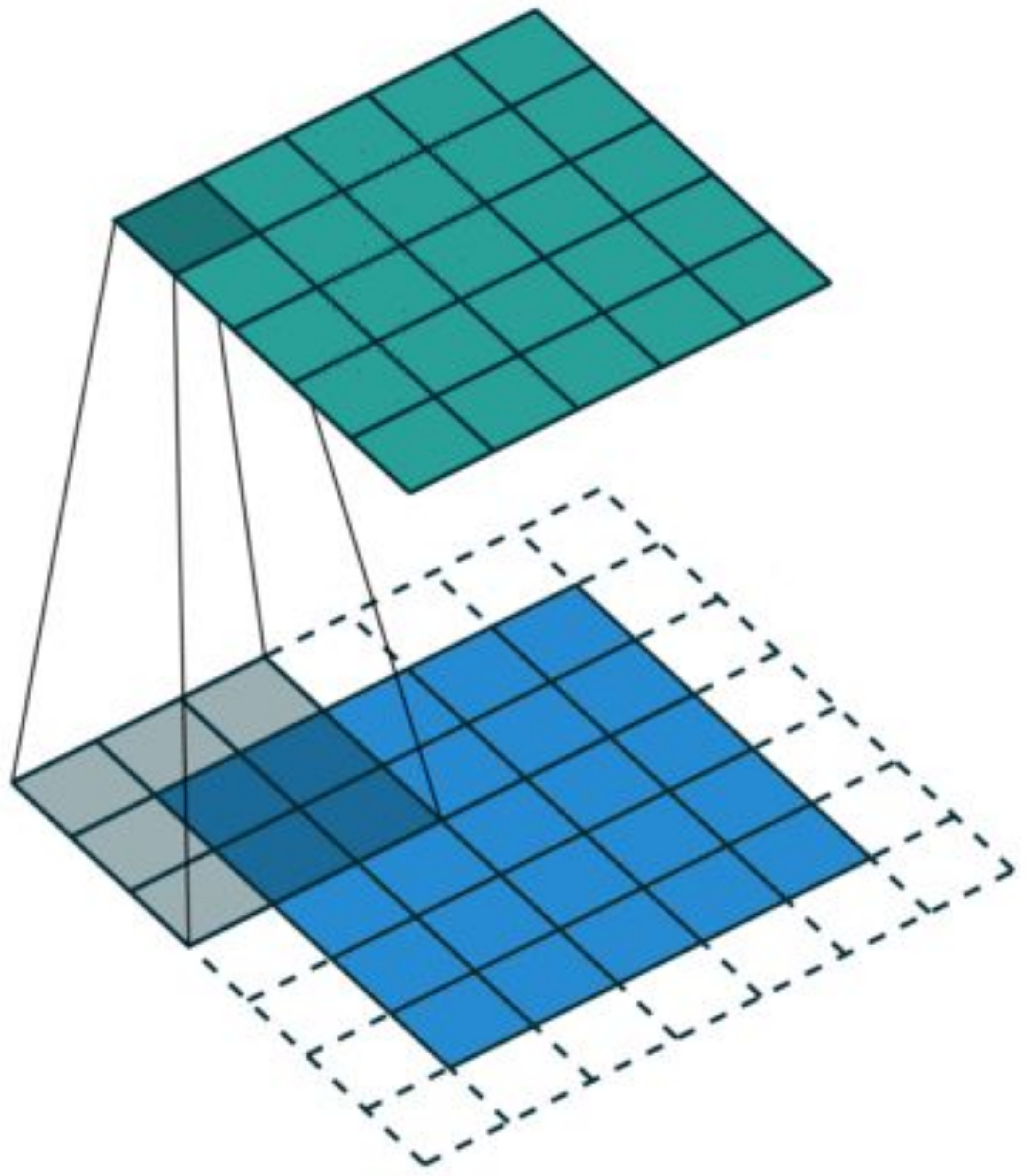
- **Depth:** number of distinct filters we use for the convolution operation. Multiple filters are used to detect different "features" of the images



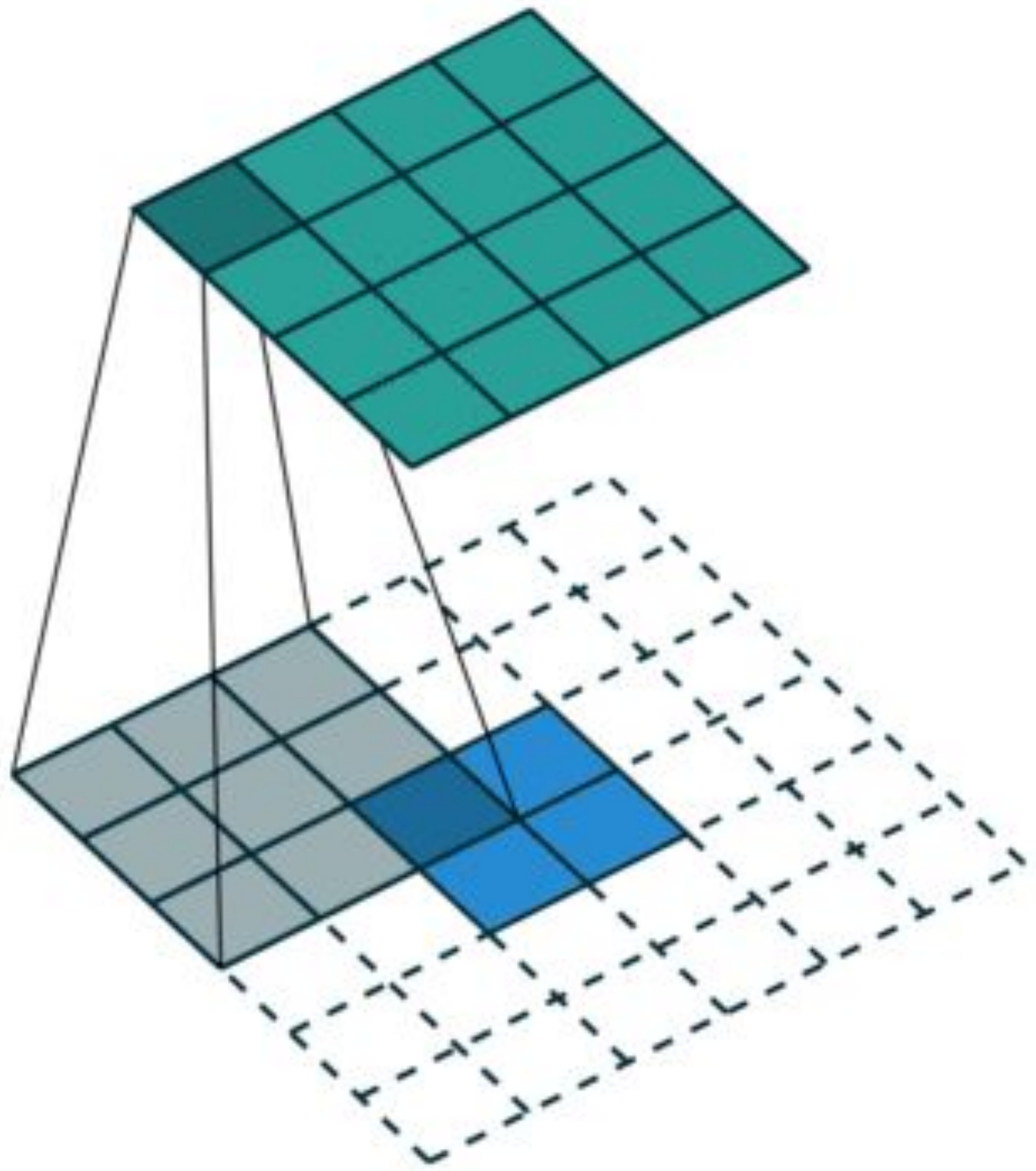


Convolution filter parameters

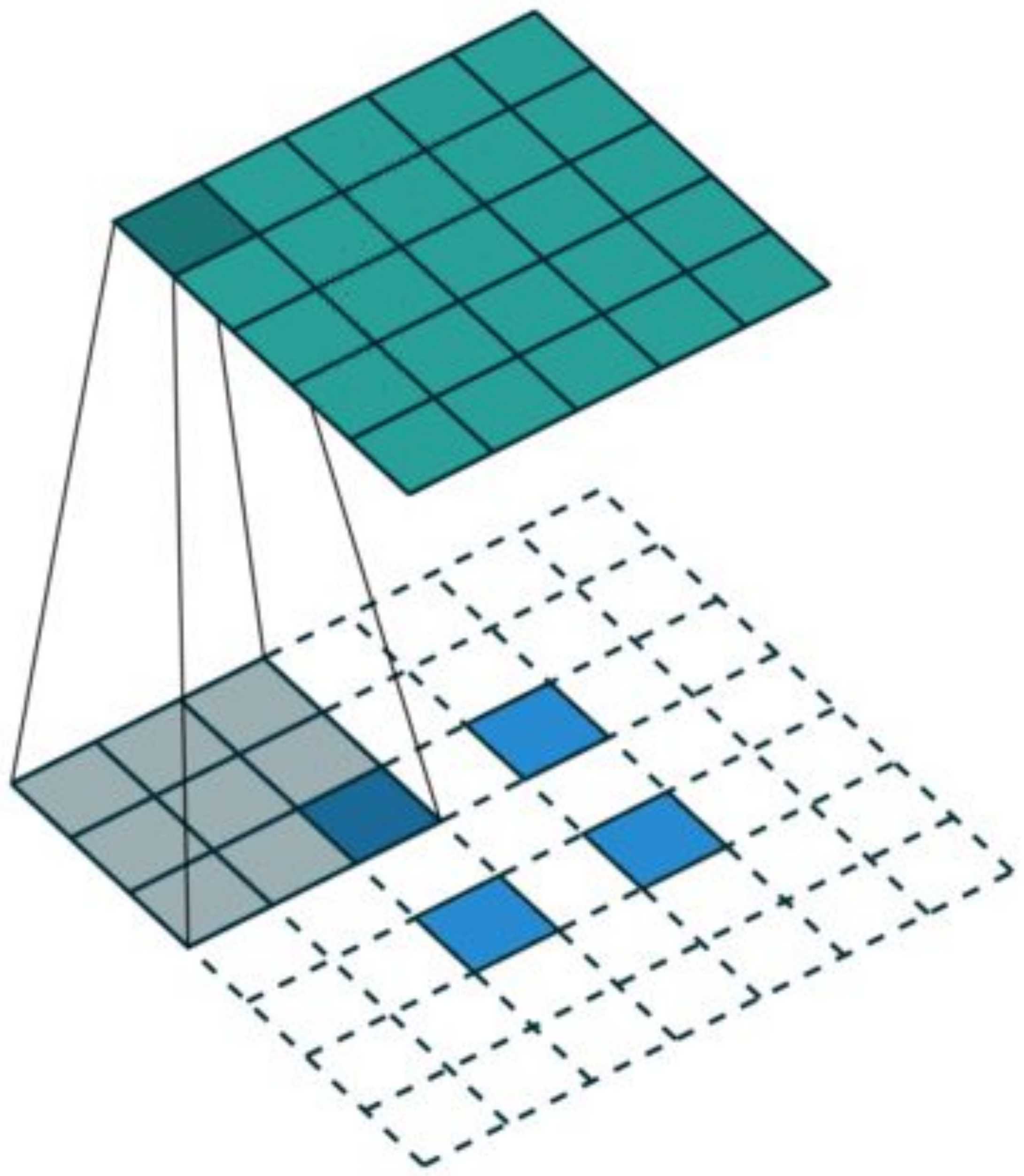
- **Zero-Padding:** pad the input matrix with zeros around the border. it allows us to control the size of the feature maps



1-padding



2-padding

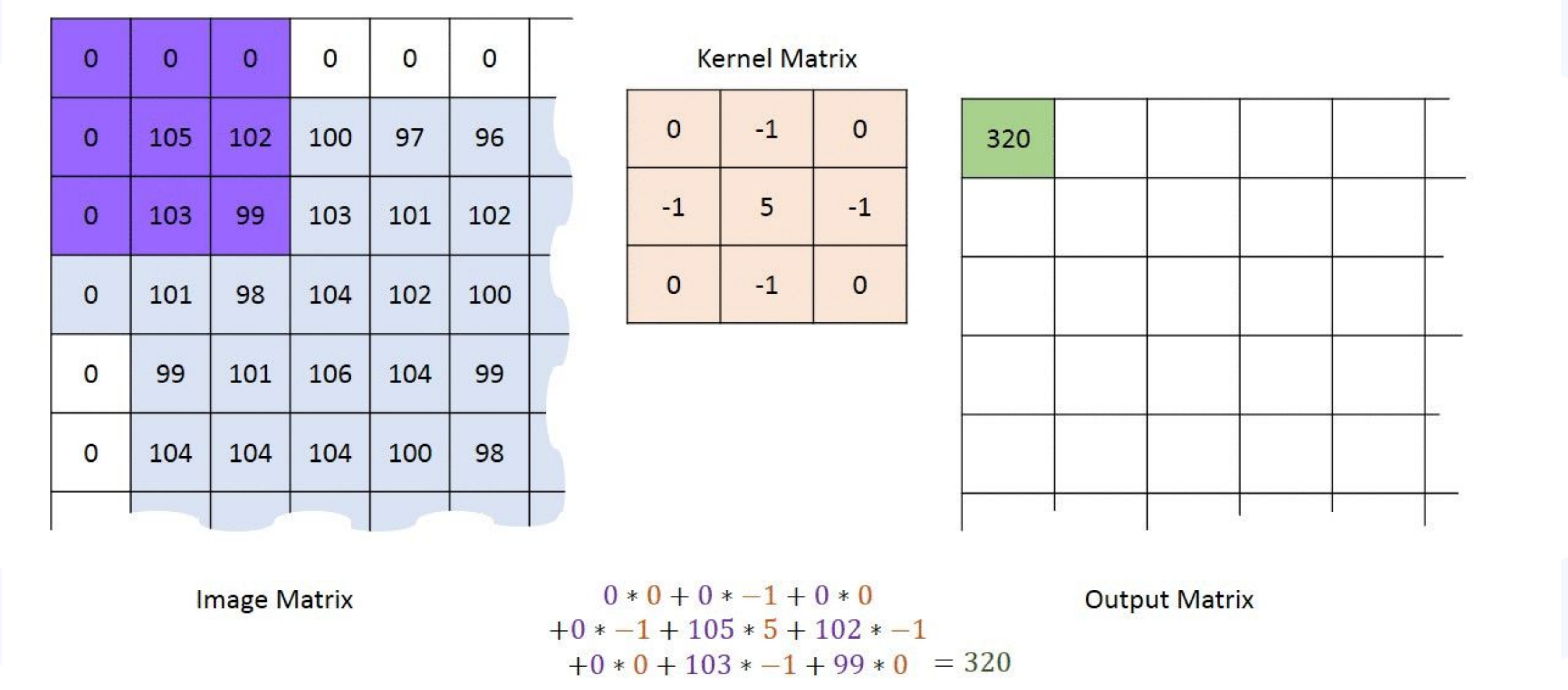


2-padding with up-sampling

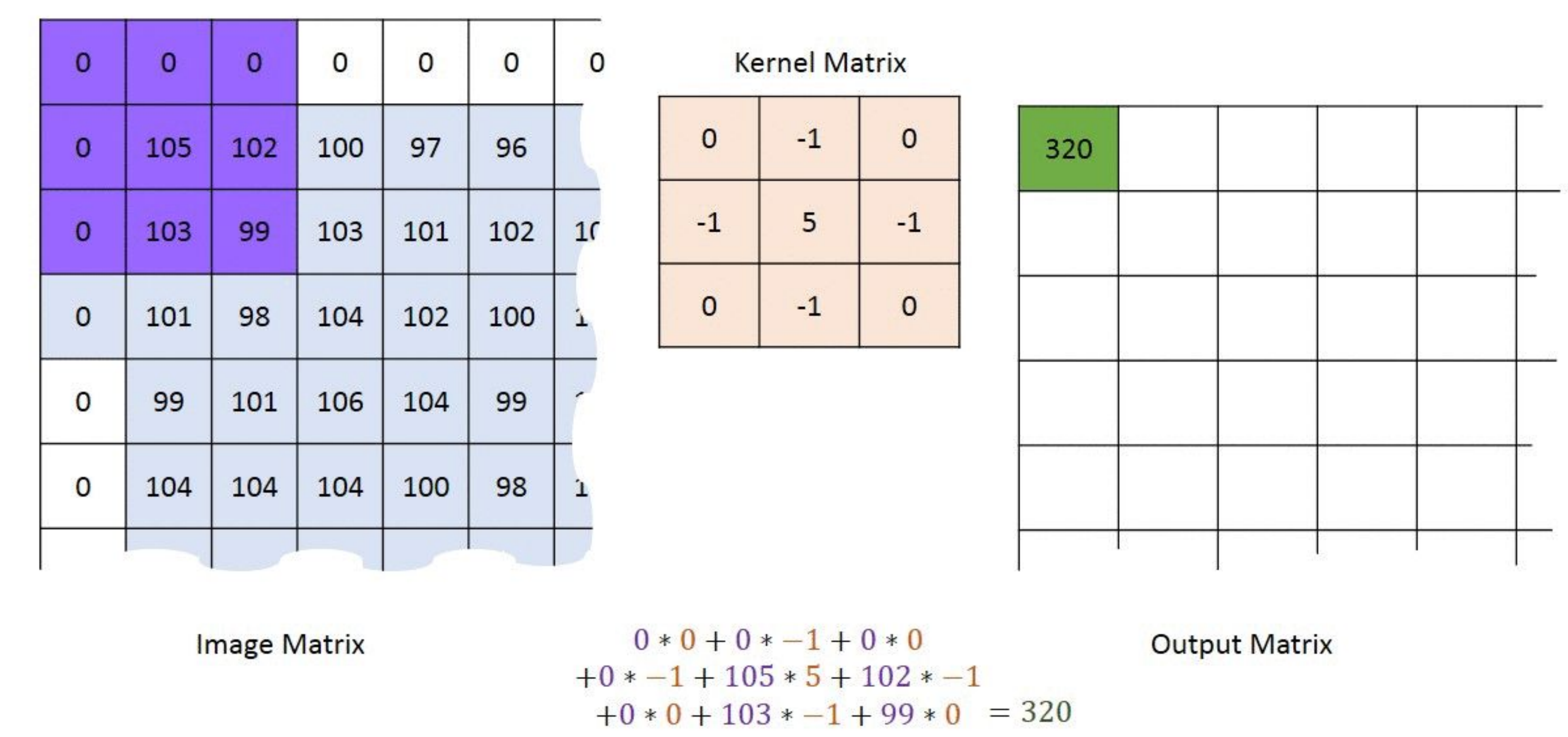
Neosperience

Convolution filter parameters

- **Stride:** number of pixels by which we slide our filter matrix. Having a larger stride will produce smaller feature maps



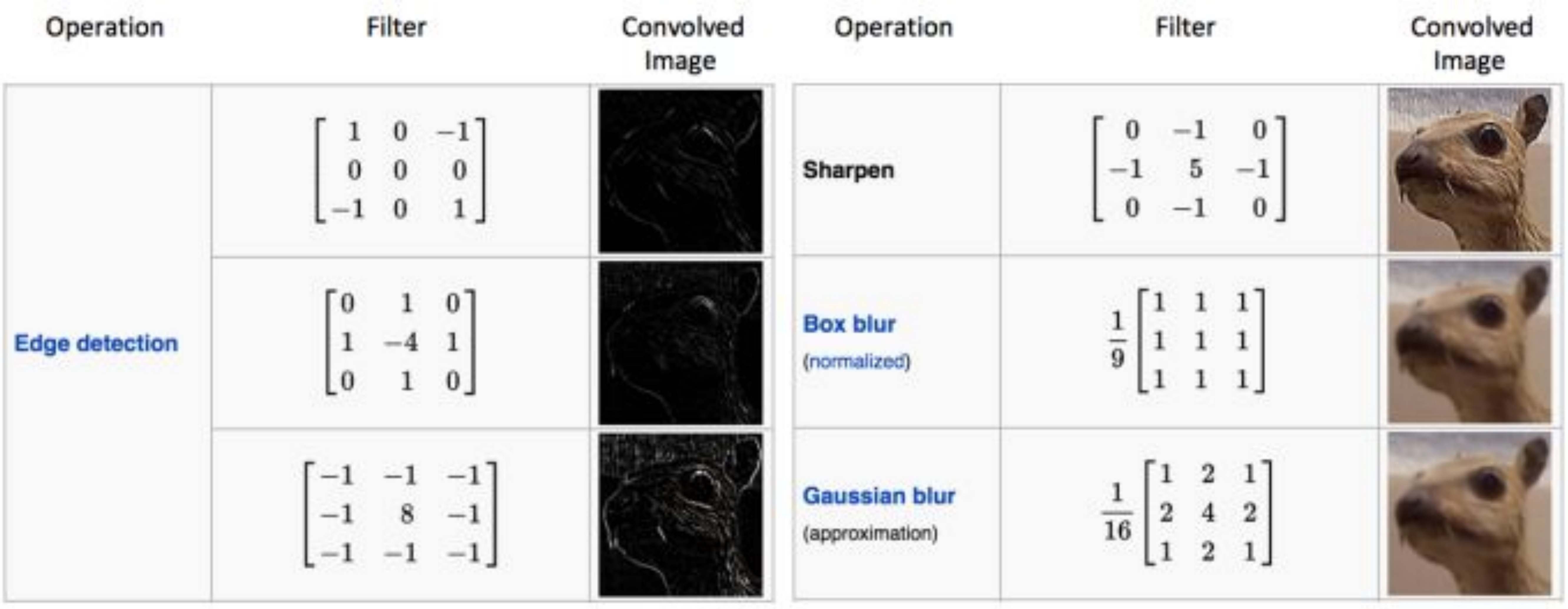
Convolution with horizontal and vertical strides = 1



Convolution with horizontal and vertical strides = 2

Classic Computer Vision filters

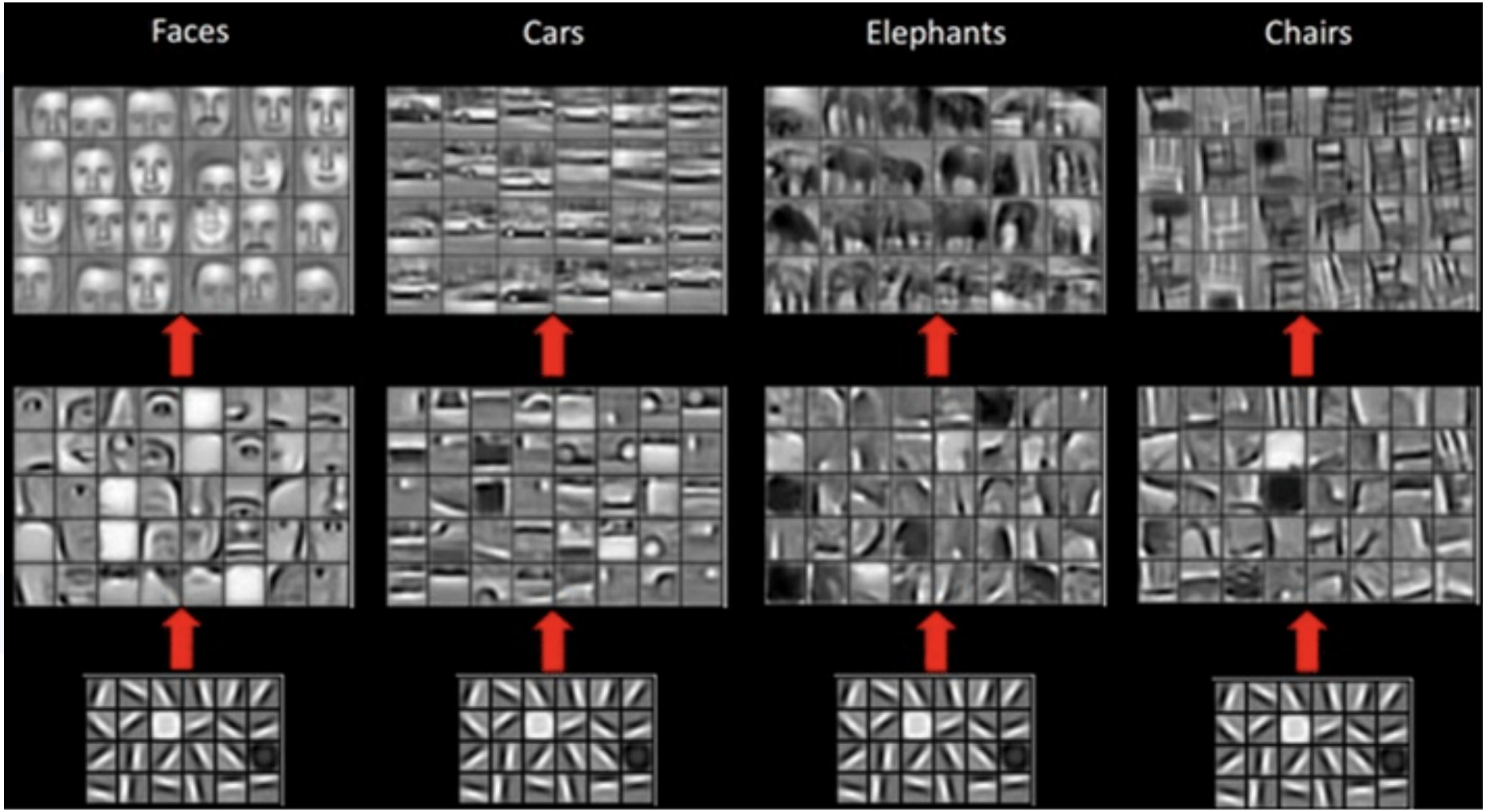
Classic CV filters are set by the model designer and are “experience based”, depending on the context of the images and the task to be achieved.



Operation	Filter	Convolved Image	Operation	Filter	Convolved Image
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$		Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$		Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$		Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

CNN learned filters

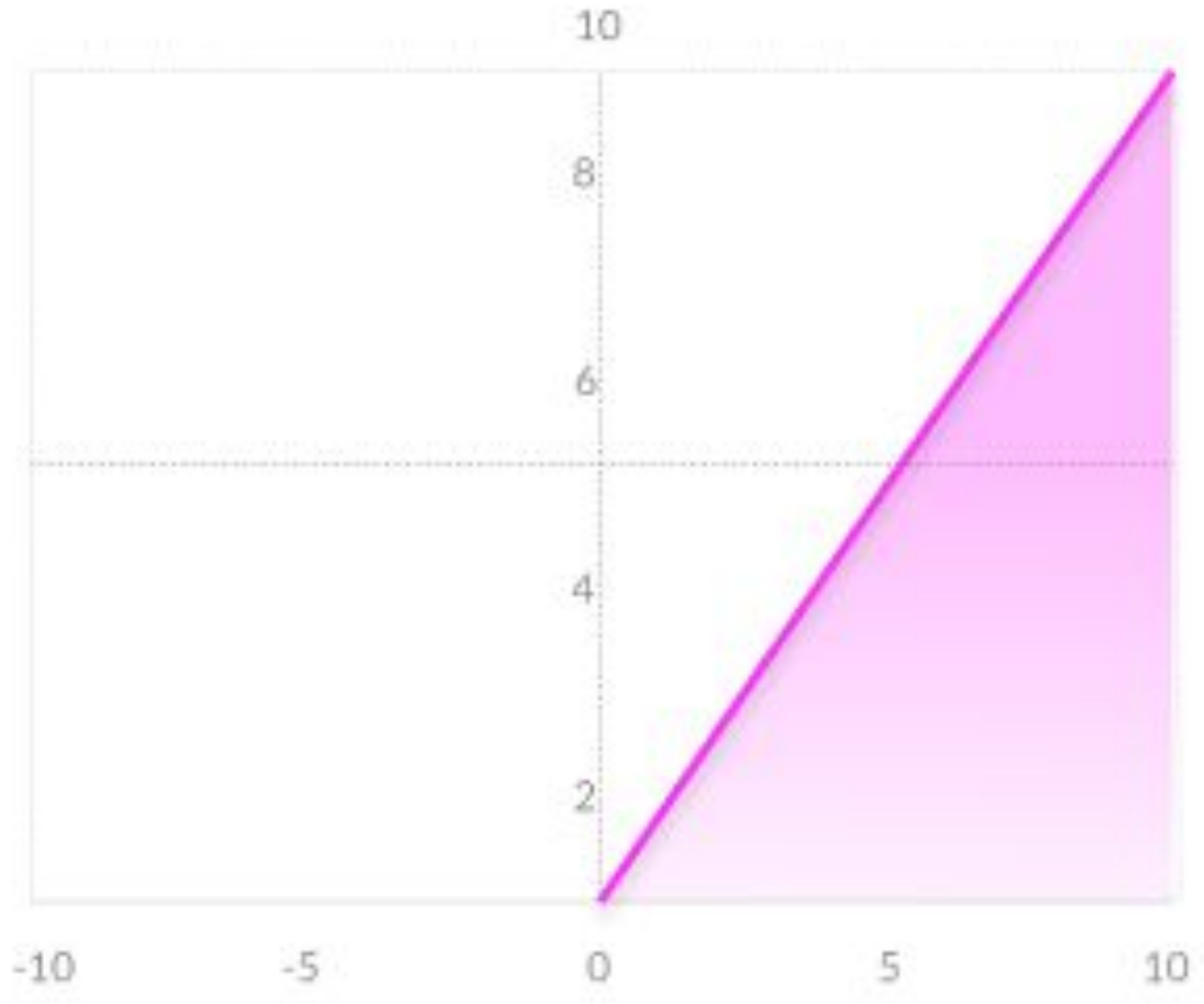
CNN filters are learned by the network itself, surprisingly identifying understandable context features



Neosperience

Non linearity

A commonly used activation function is the **Rectified Linear Unit** (ReLU), a non-linear function and element wise operation (applied per pixel) that replaces all negative pixel values in the feature map by zero.



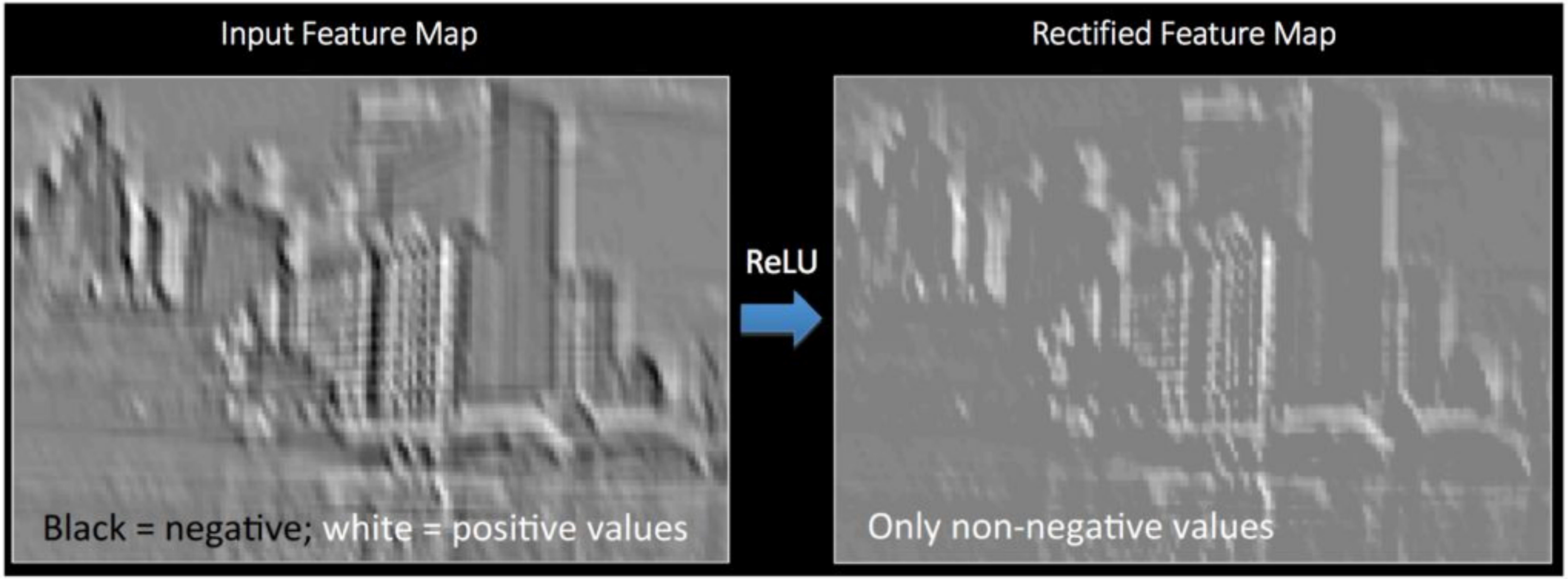
ReLU function

$$\phi(x) = \begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases}$$

ReLU derivative

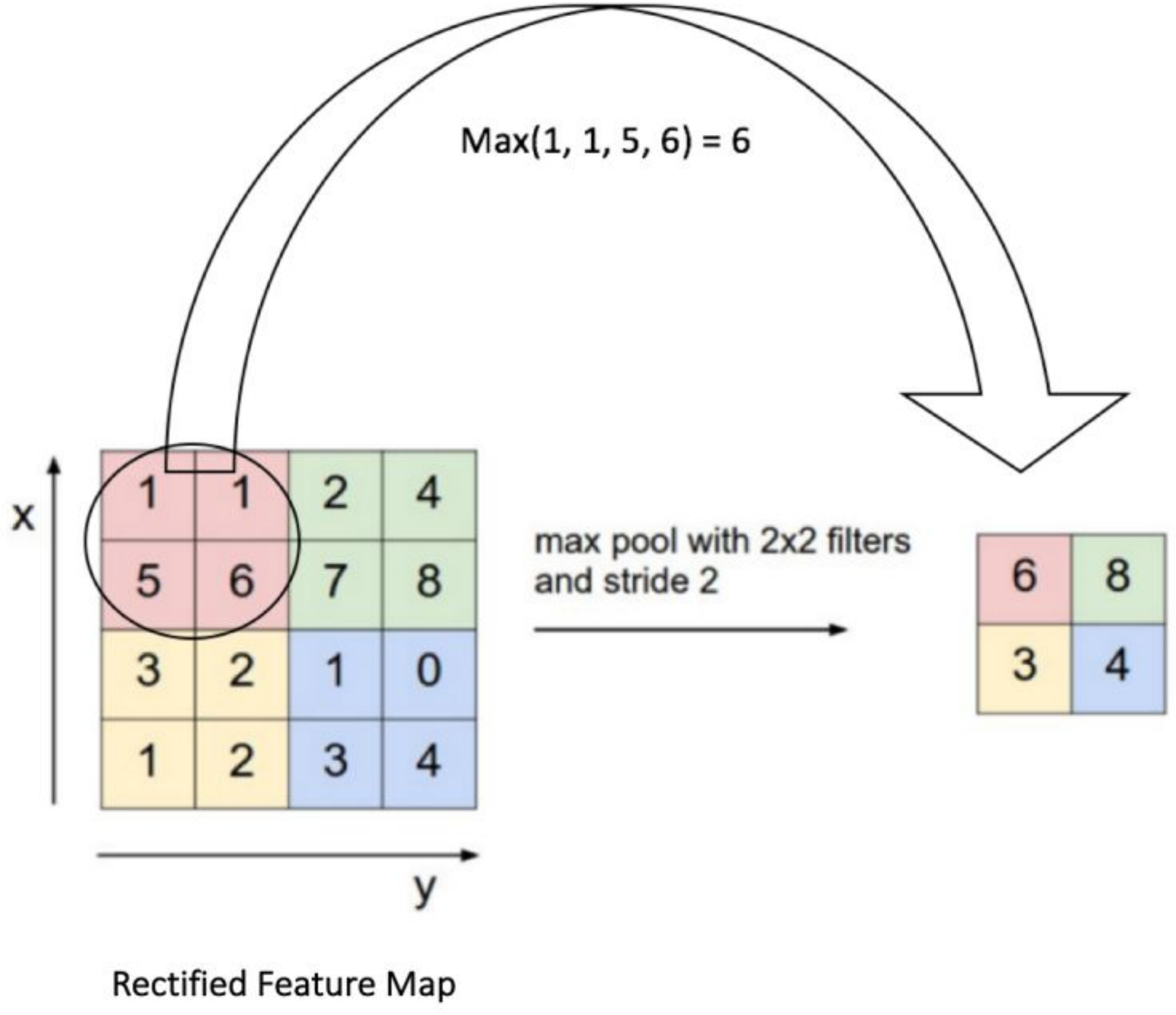
$$\phi'(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases}$$

Non linearity



Pooling

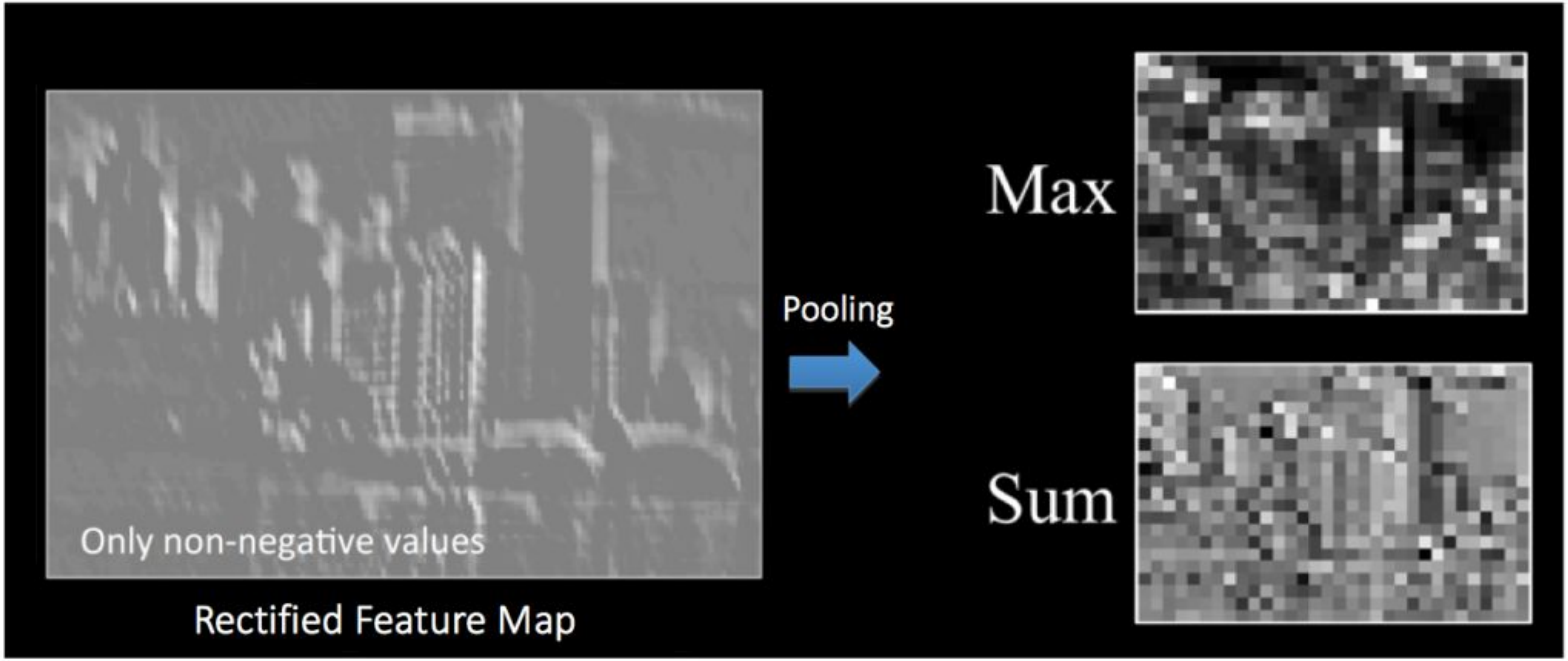
Spatial Pooling (also called subsampling or downsampling) reduces the dimensionality of each feature map but retains the most important information. Spatial Pooling can be of different types: Max, Average, Sum etc.



- makes the **input representations** (feature dimension) **smaller** and more manageable
- **reduces the number of parameters** and computations in the network
- makes the network **invariant to small transformations**, distortions and translations in the input image (a small distortion in input will not change the output of Pooling – since we take the maximum / average value in a local neighborhood)
- helps to arrive at an almost **scale invariant** (equivariant) **representation** of our image. This is very powerful since we can detect objects in an image no matter where they are located

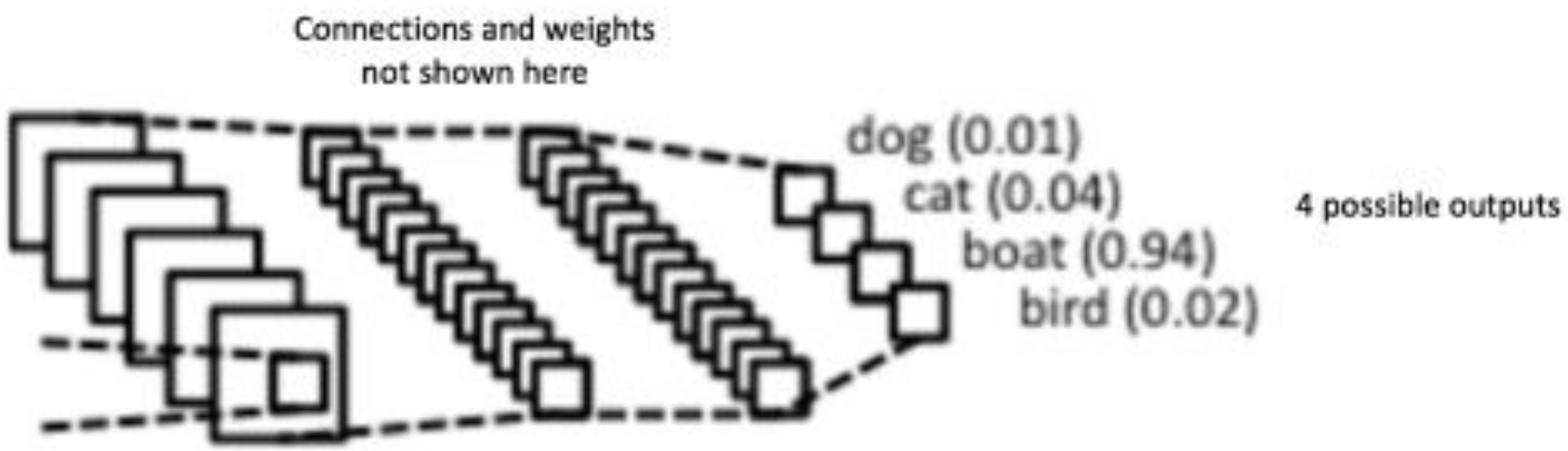
Neosperience

Pooling


Only non-negative values
Rectified Feature Map
Pooling
Max
Sum

Training and loss function

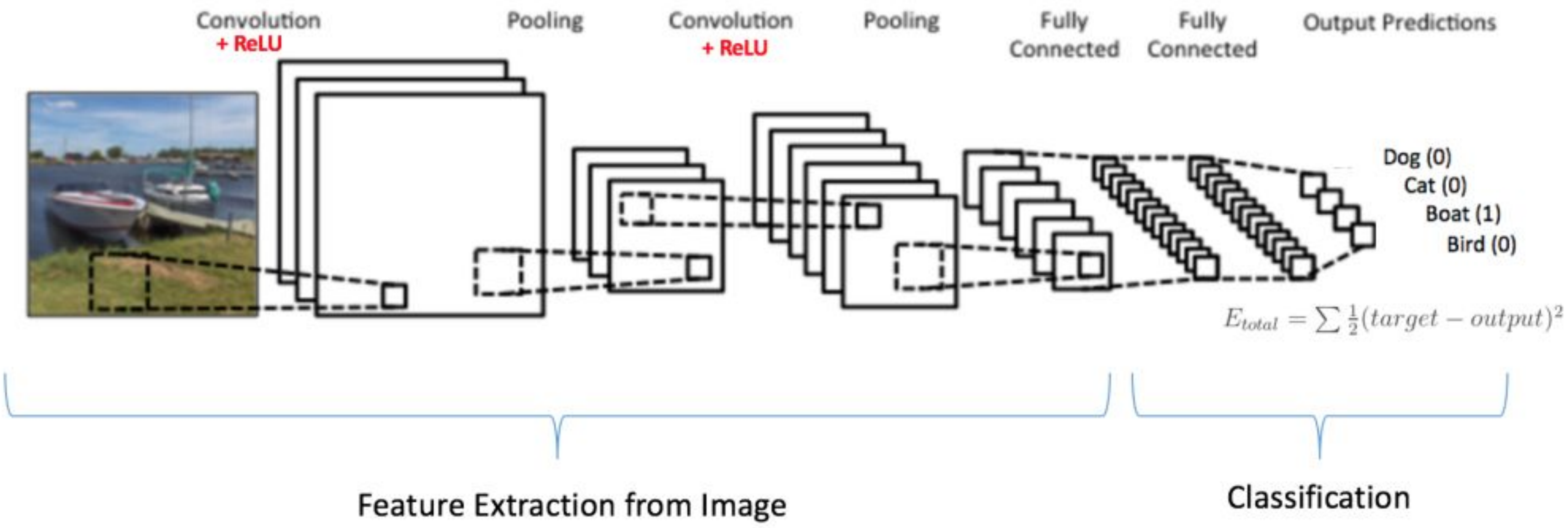
- The Fully Connected layer is a traditional Multi Layer Perceptron that uses a **Softmax** activation function in the output layer, flattening the output of convolutional and pooling layers
- The output from the convolutional and pooling layers represent high-level features of the input image
- The purpose of the Fully Connected layer is to use these features for classifying the input image into various classes based on the training dataset.
- This is also a cheap way of learning non-linear combinations of these features. Most of the features from convolutional and pooling layers may be good for the classification task, but combinations of those features might be even better



$$\mathrm{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

Training and loss function

Now we have all the building blocks to train our neural network



Neosperience

Training and loss function

Training (tuning of the weights) consist of the following steps:

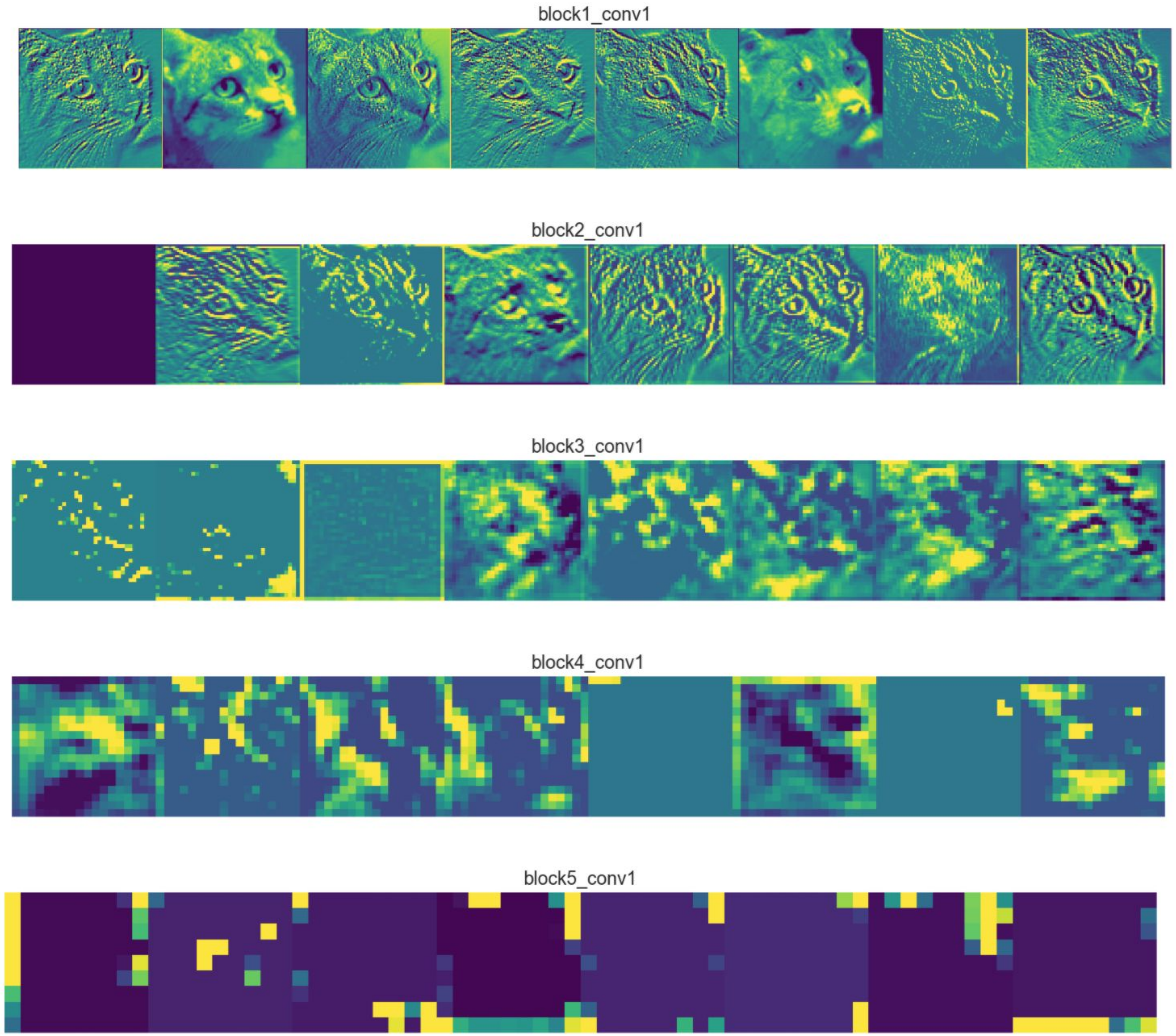
1) initialize all filters and parameters (weights) with random values
2) The network takes a training image as input, goes through the forward propagation step (convolution, ReLU and pooling operations along with forward propagation in the Fully Connected layer) and finds the output probabilities OP_i for each class (normalized with the softmax)
3) Calculate the total error (**Loss Function**) at the output layer comparing the target probabilities TP_i with the output ones. Two commonly used metrics are:

Mean Squared Error $$\frac{1}{n}\sum_{i=1}^{n}(TP_i - OP_i)^2$$

Cross-Entropy $$-\sum_{i=1}^{n} TP_i \log(OP_i)$$

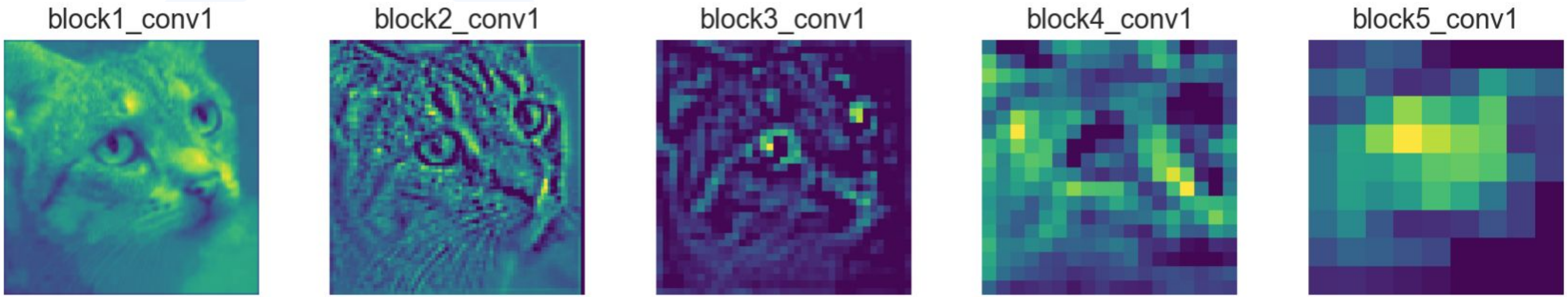
4) Use Backpropagation to calculate the gradients of the error with respect to all weights in the network and use gradient descent to update all weights and parameter values to minimize the output error
5) Repeat steps 2-4 with all images in the training set

Visualizing CNN

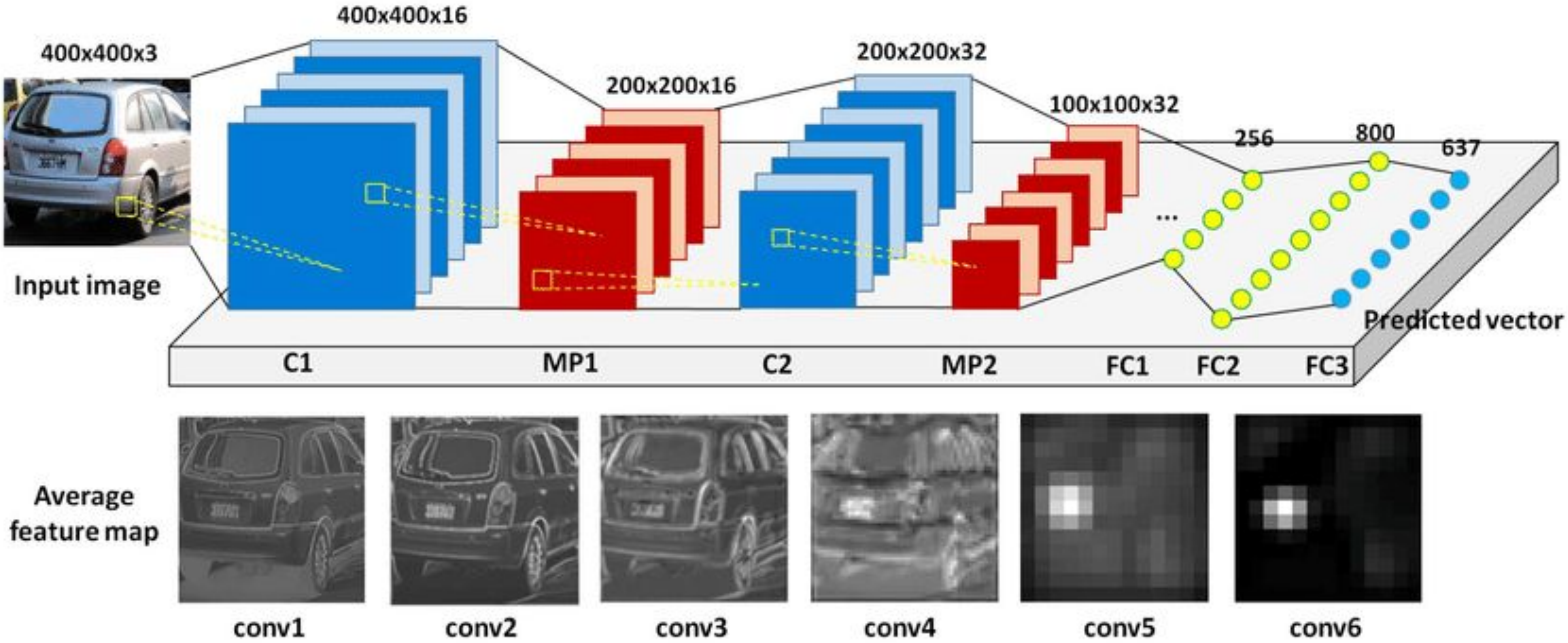


Neosperience

Visualizing CNN



Visualizing CNN



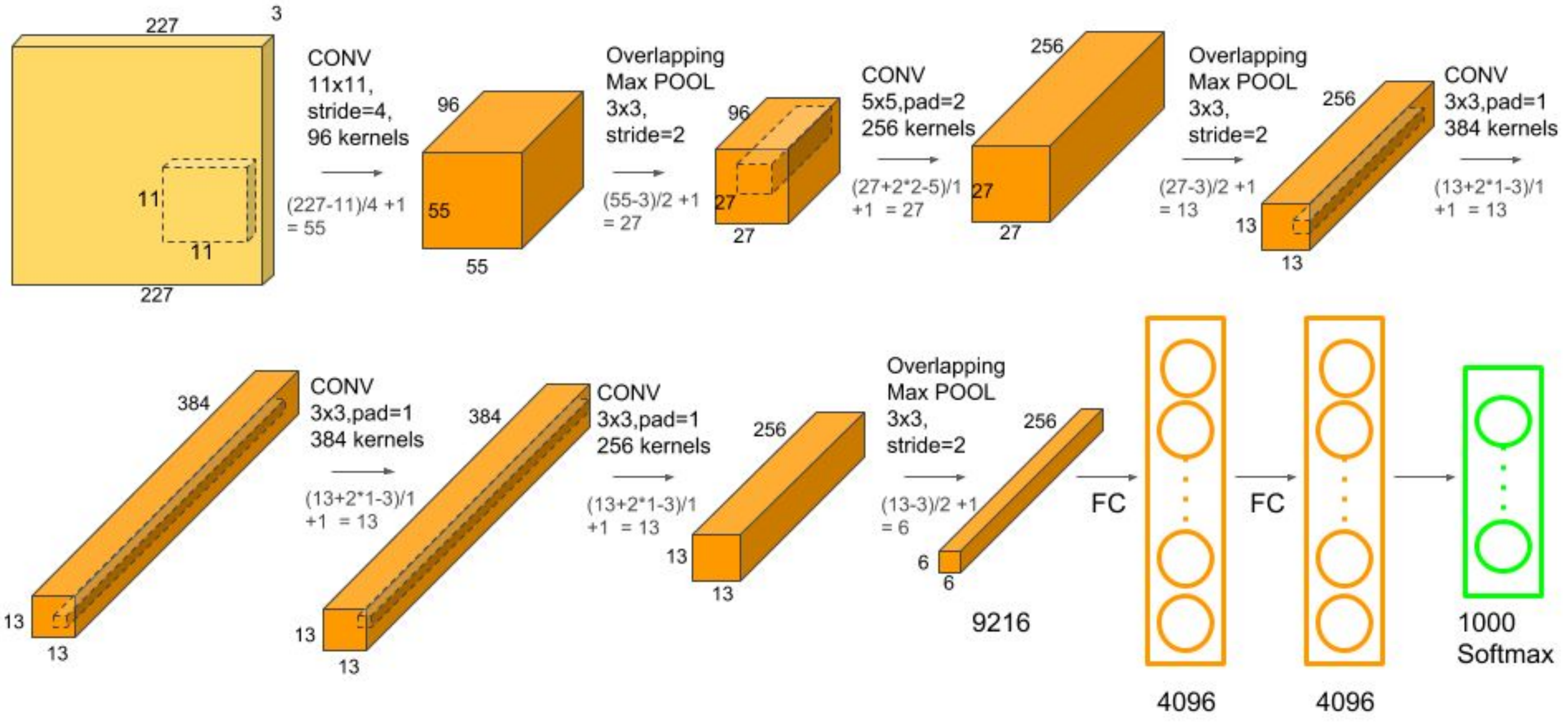
Useful implementation tips

Overfitting is a common problem for all Neural Network. Following tips may prevent the problem and speed up training time:

- Preprocessing:
  - Local Mean Subtraction
  - Normalization
- Better optimization methods:
  - (Batch) Stochastic Gradient Descent (SGD)
  - ADAptive Moment (ADAM) that combines momentum and RMSprop
- Regularization
  - Weight decay
  - Dropout
- Data augmentation

CNN Architectures: AlexNet (Alex Krizhevsky - 2012)

- AlexNet was much larger than previous CNNs. It has 60 million parameters and 650,000 neurons and took five to six days to train on two GTX 580 3GB GPUs.
- consists of **5 Convolutional Layers** and **3 Fully Connected Layers**

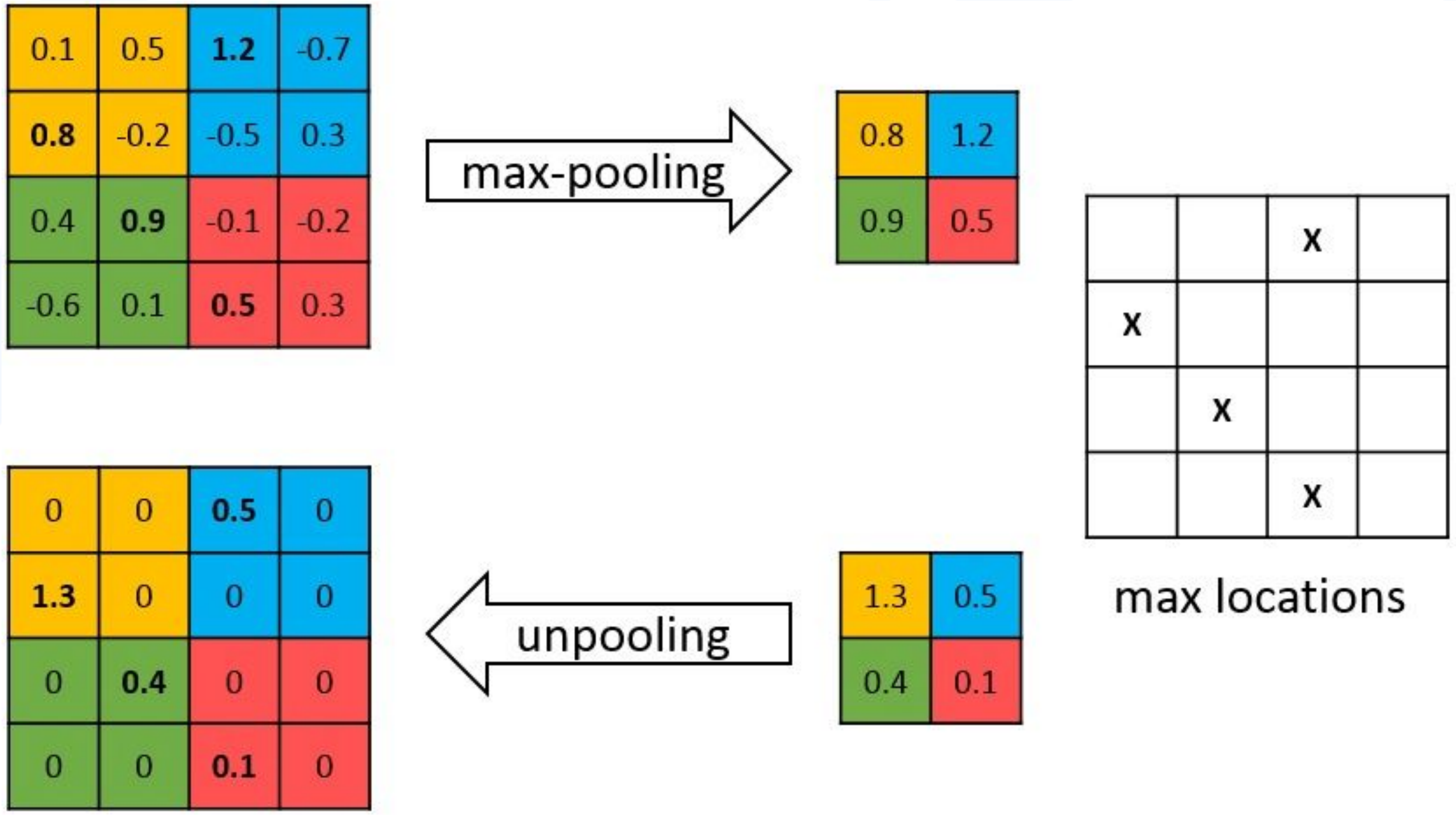


Neosperience

CNN Architectures: ZFnet (Zeiler & Fergus - 2013)

- Before this model CNN were black boxes. This model provides insights into how CNNI networks are learning internal representations
- Main idea is to improve AlexNet introducing **DeconvNet**, a deconvolutional net that acts as the opposite of convolution and **Unpooling** (inverse of pooling)

Unpooling

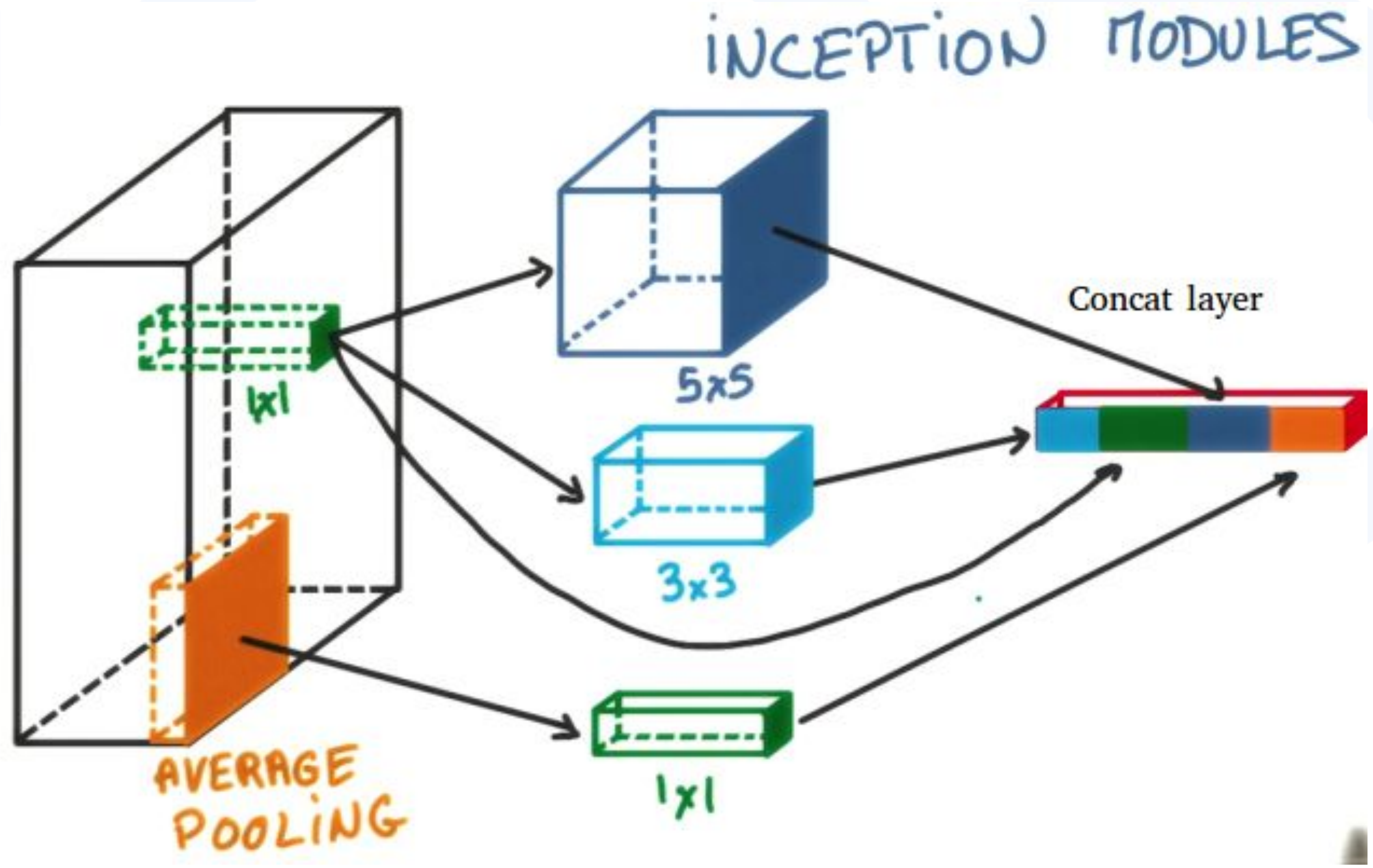


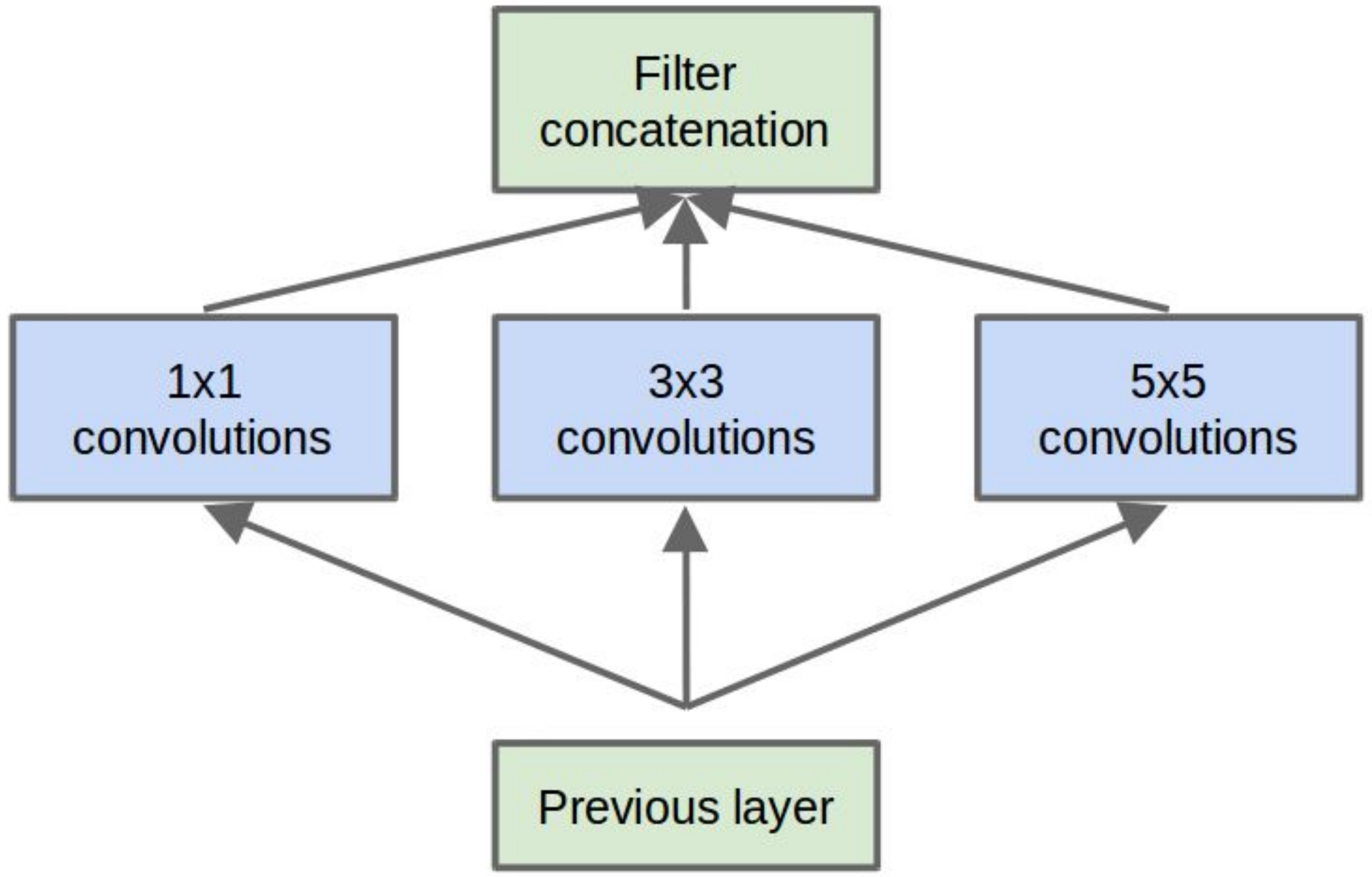
Deconvolution

Blue is input, cyan is output

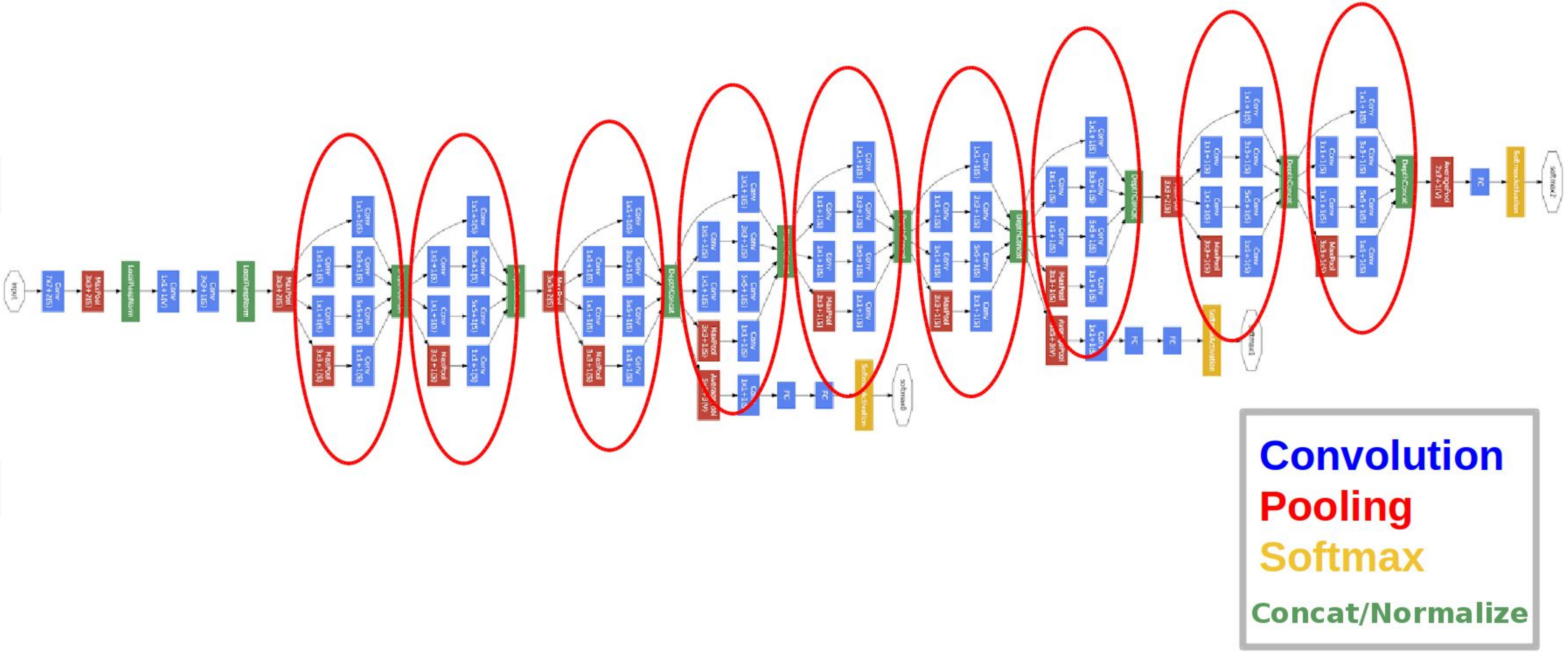
CNN Architectures: GoogLeNet (2014)

- Introduced **Inception** layer, convolving in parallel different sizes from the most accurate detailing (1x1) to a bigger one (5x5)
- The idea is that a series of filters with different sizes, will handle better multiple objects scales with the advantage that all filters on the inception layer are learnable.



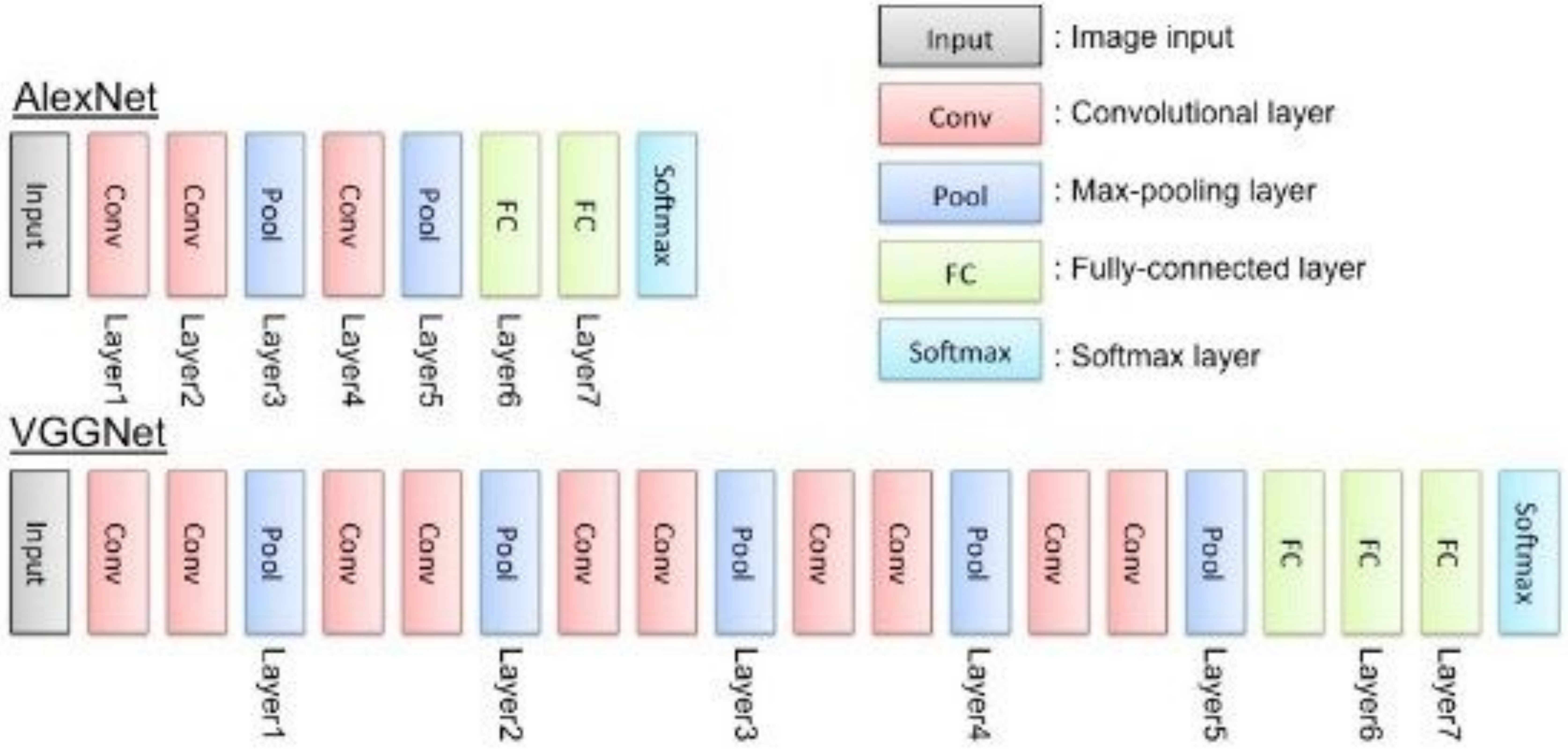


CNN Architectures: GoogLeNet (2014)



CNN Architectures: VGGNet (2014)

- Improved AlexNet using more convolutional filter blocks but with smaller size
- Main contribution was in showing that the depth of the network (number of layers) is a critical component for good performance



Neosperience

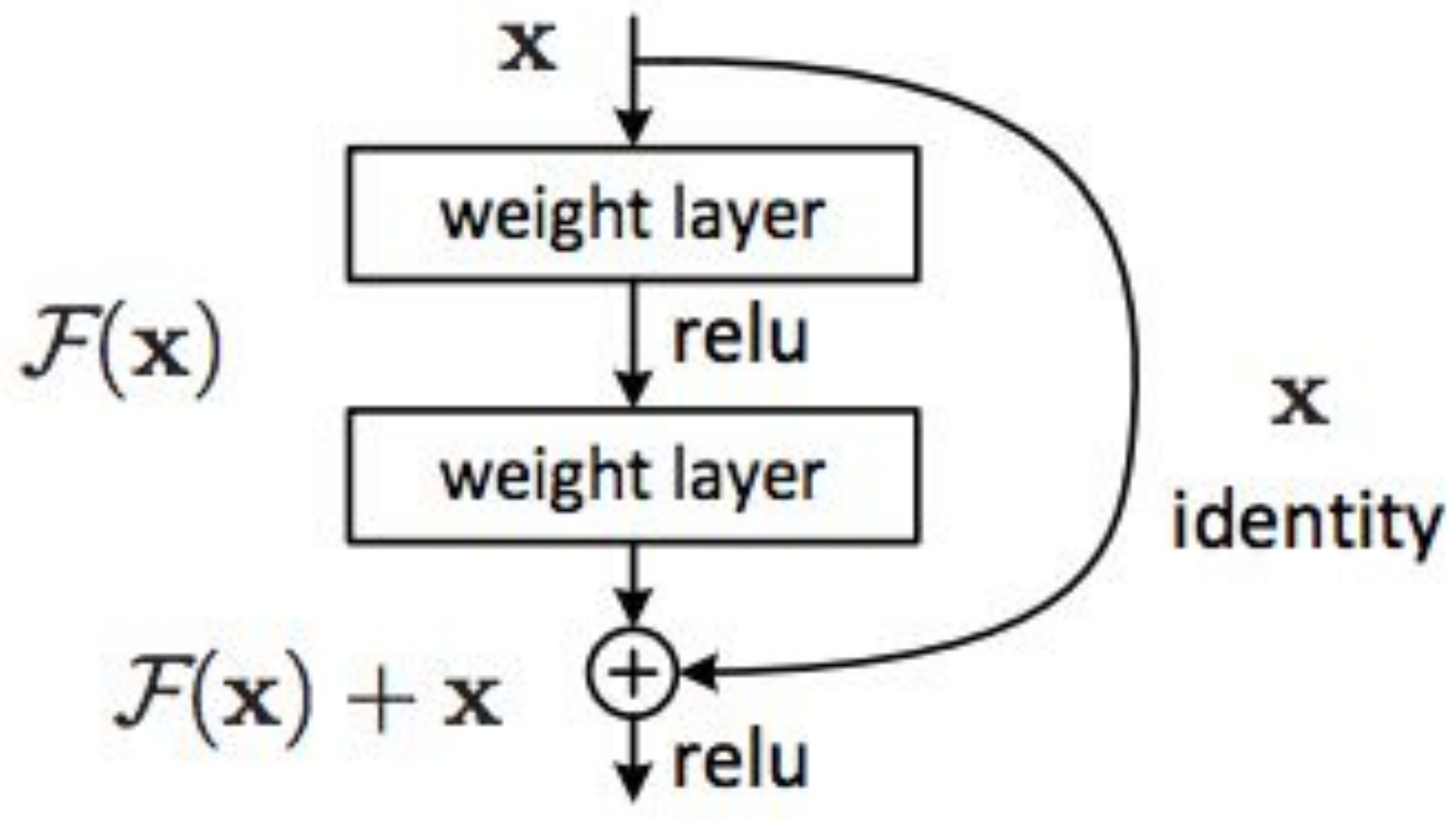
CNN Architectures: ResNets (2015)

- Faces the **vanishing gradient** problem, allowing to increase the number of layers
- Neural networks are good function approximators, they should be able to easily solve the identify function, where the output of a function becomes the input itself

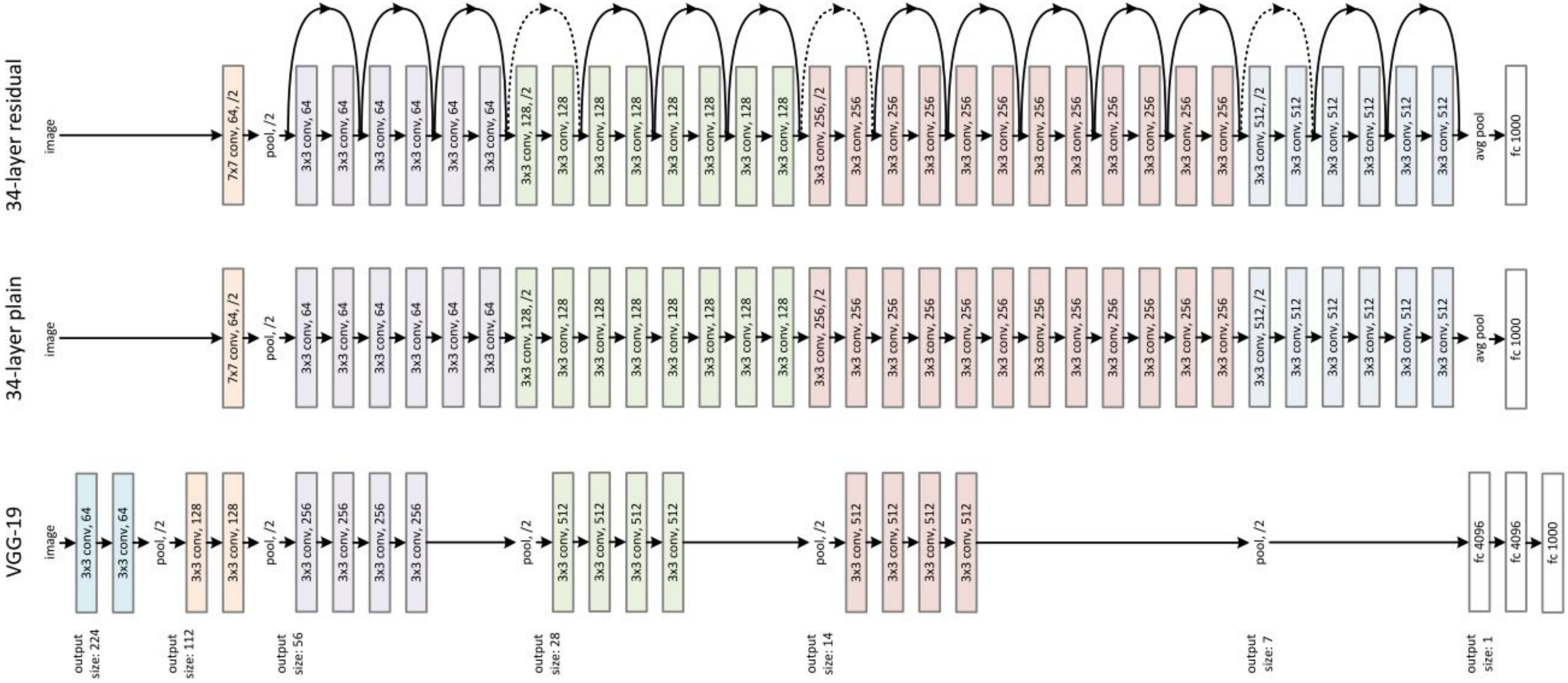
$$f(x) = x$$

- Following the same logic, if we bypass the input to the first layer of the model to be the output of the last layer of the model, the network should be able to predict whatever function it was learning before with the input added to it

$$f(x) + x = h(x)$$



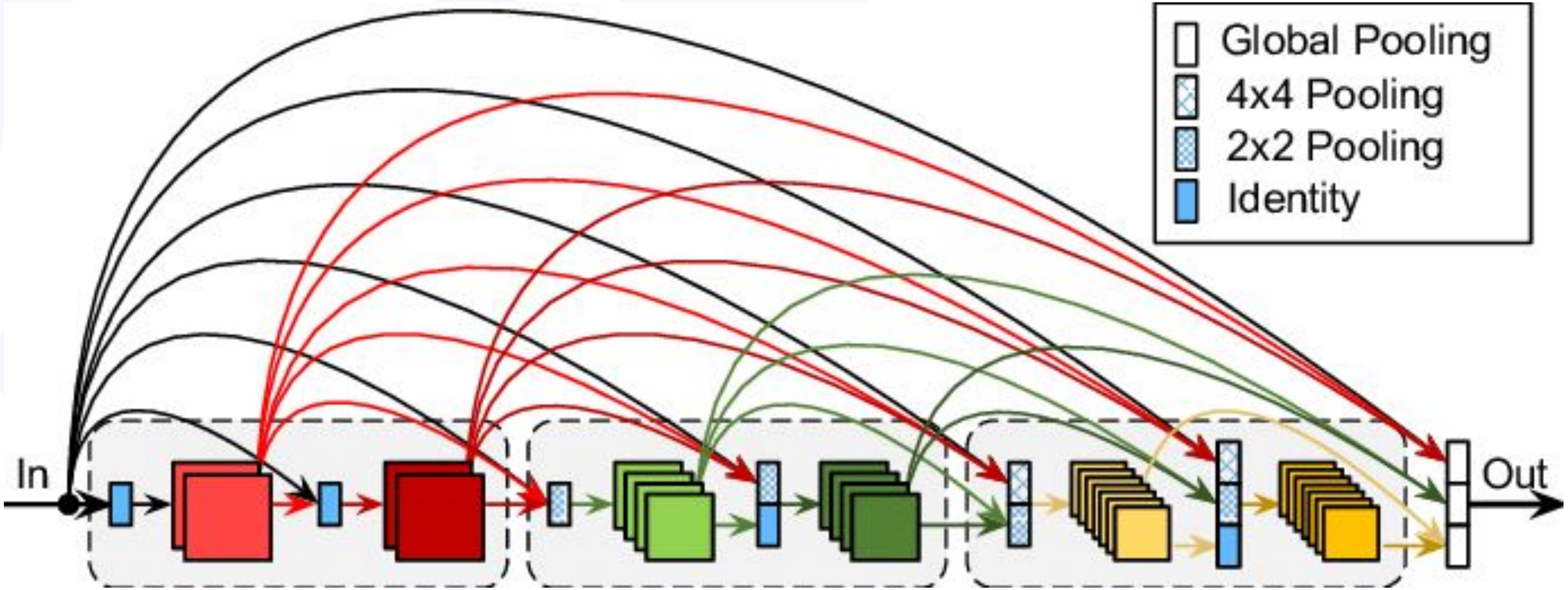
CNN Architectures: ResNets (2015)



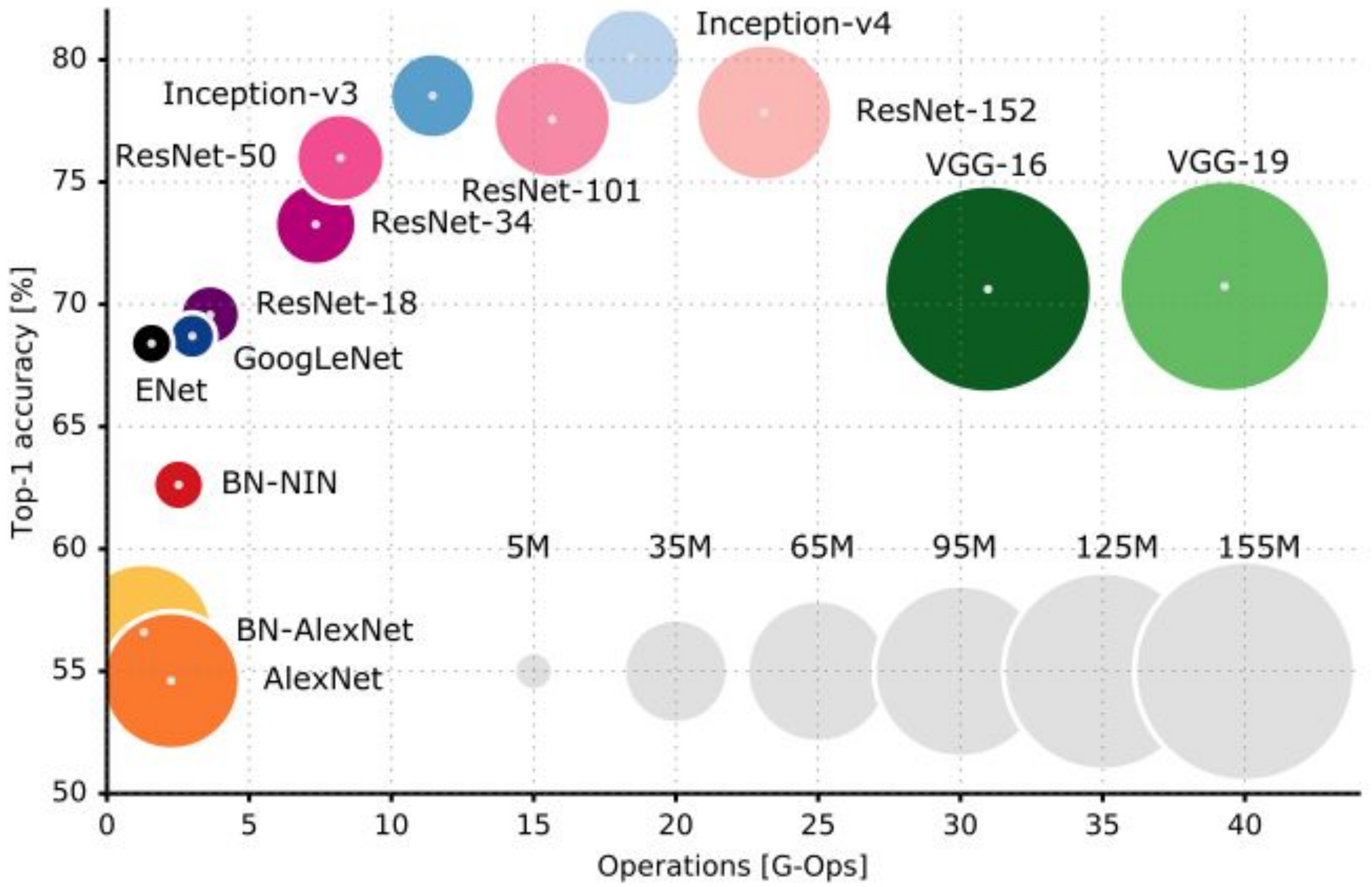
Neosperience

CNN Architectures: DenseNet (2016)

- DenseNet is composed of **Dense blocks**. In those blocks, the layers are densely connected together: each layer receive in input all previous layers output feature maps
- This **extreme use of residual** creates a deep supervision because each layer receive more supervision from the loss function thanks to the shorter connections



CNN Architectures: Complexity vs Accuracy



Building Neural Networks is not easy..

Deep Learning Frameworks

Neosperience

Tensorflow + Keras

Built by Google TensorFlow is the foundation of many DeepLearning services and one of the most used frameworks, even through its high-level library Keras.

Why TensorFlow

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

About →

Easy model building

Build and train ML models easily using intuitive high-level APIs like Keras with eager execution, which makes for immediate model iteration and easy debugging.

Robust ML production anywhere

Easily train and deploy models in the cloud, on-prem, in the browser, or on-device no matter what language you use.

Powerful experimentation for research

A simple and flexible architecture to take new ideas from concept to code, to state-of-the-art models, and to publication faster.

Keras: The Python Deep Learning library

Keras

You have just found Keras.

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. *Being able to go from idea to result with the least possible delay is key to doing good research.*

Use Keras if you need a deep learning library that:

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
- Supports both convolutional networks and recurrent networks, as well as combinations of the two.
- Runs seamlessly on CPU and GPU.

ResNet50

https://github.com/tensorflow/models/blob/master/official/vision/image_classification/resnet_model.py

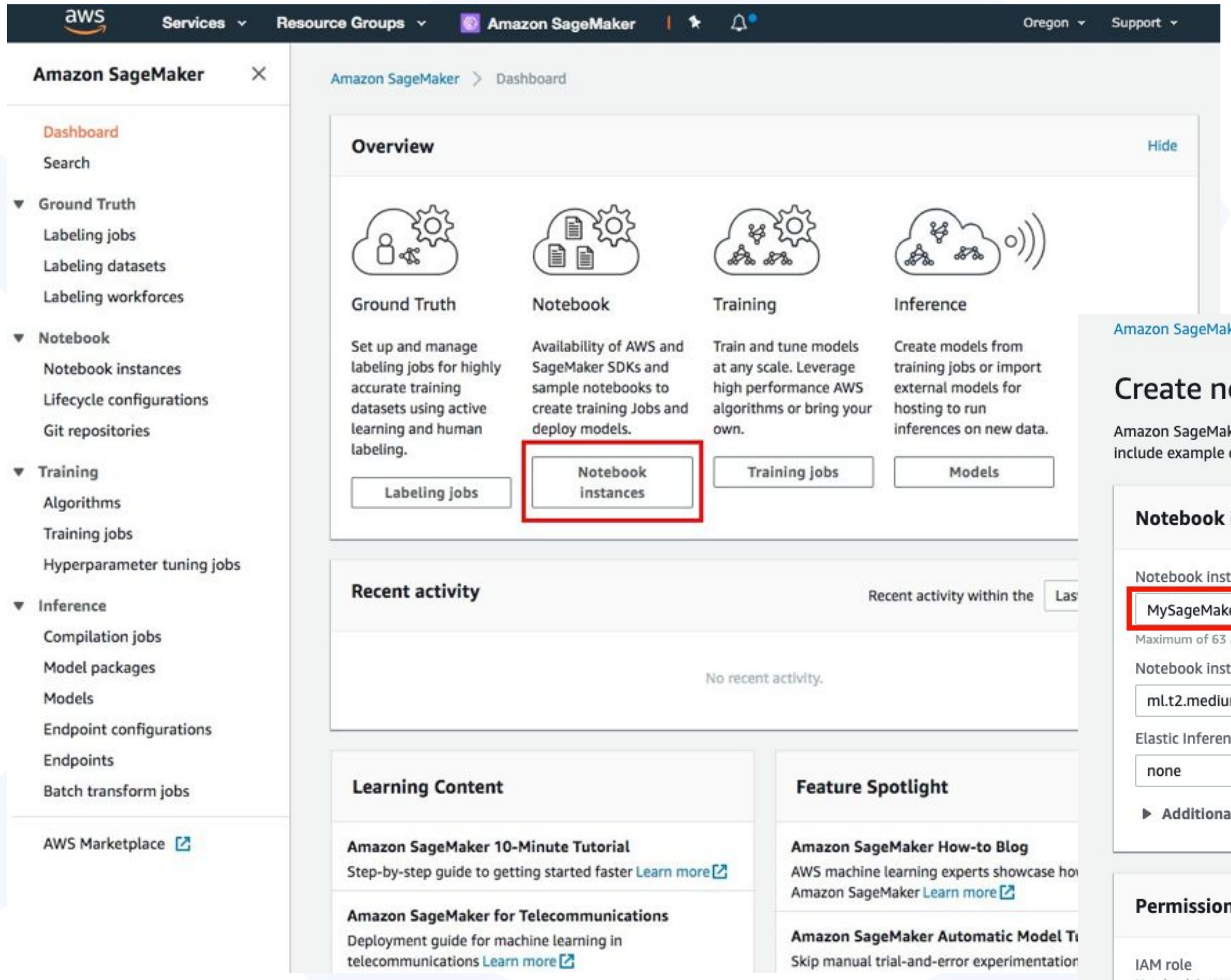
PyTorch + FastAI

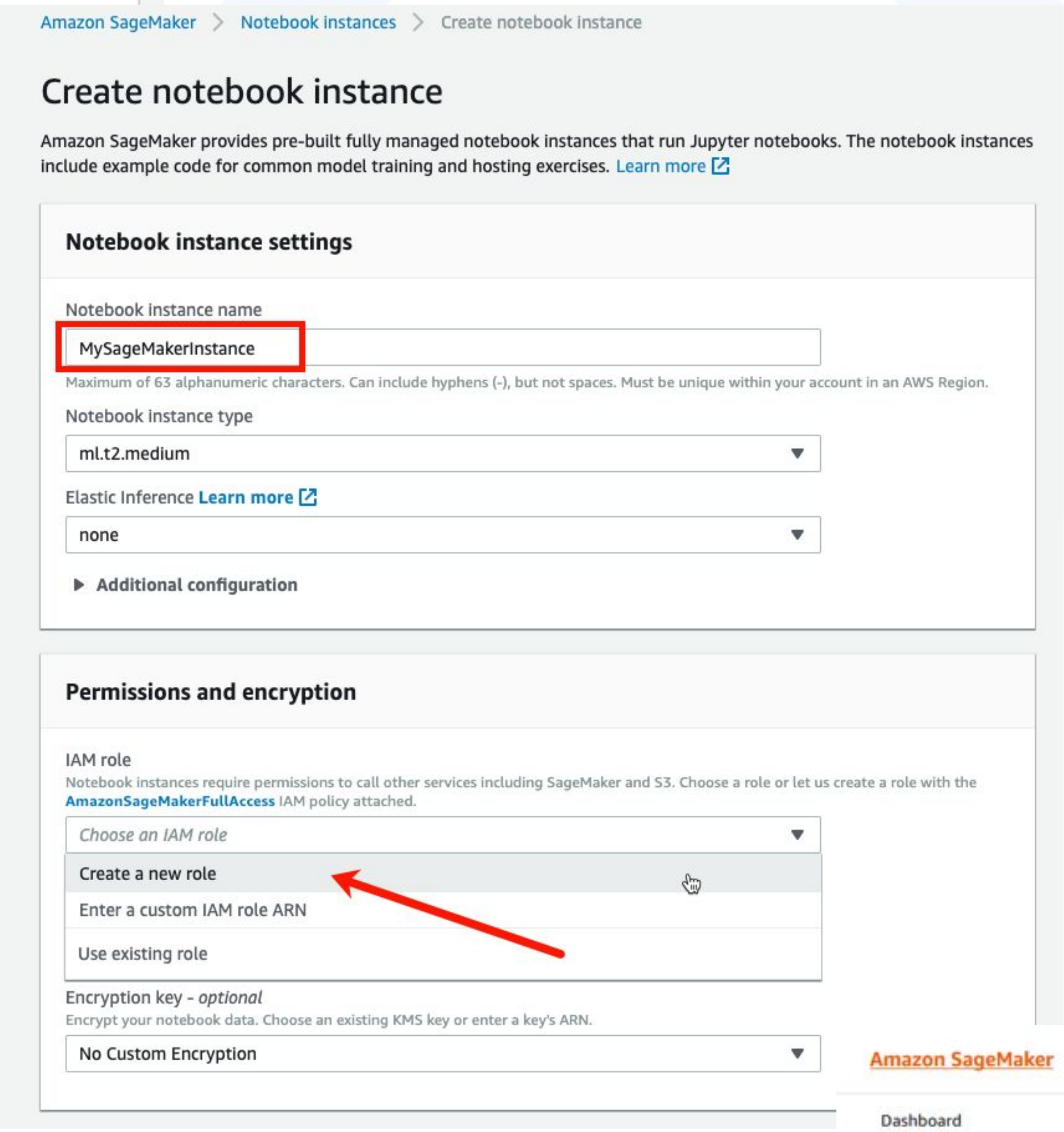
PyTorch enables fast, flexible experimentation and efficient production through a user-friendly front-end, distributed training, and ecosystem of tools and libraries.

Setting up a Deep Learning environment

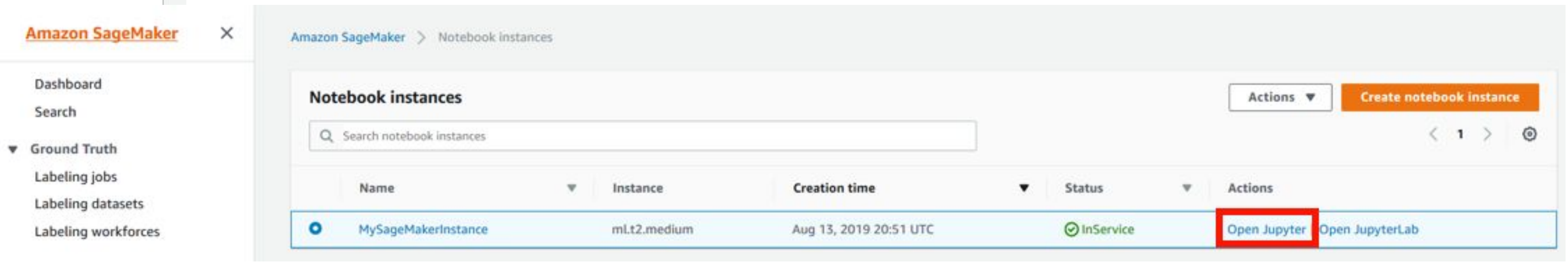
Jupyter Notebooks: an IDE for Machine Learning

Amazon SageMaker -- with Deep Learning instances





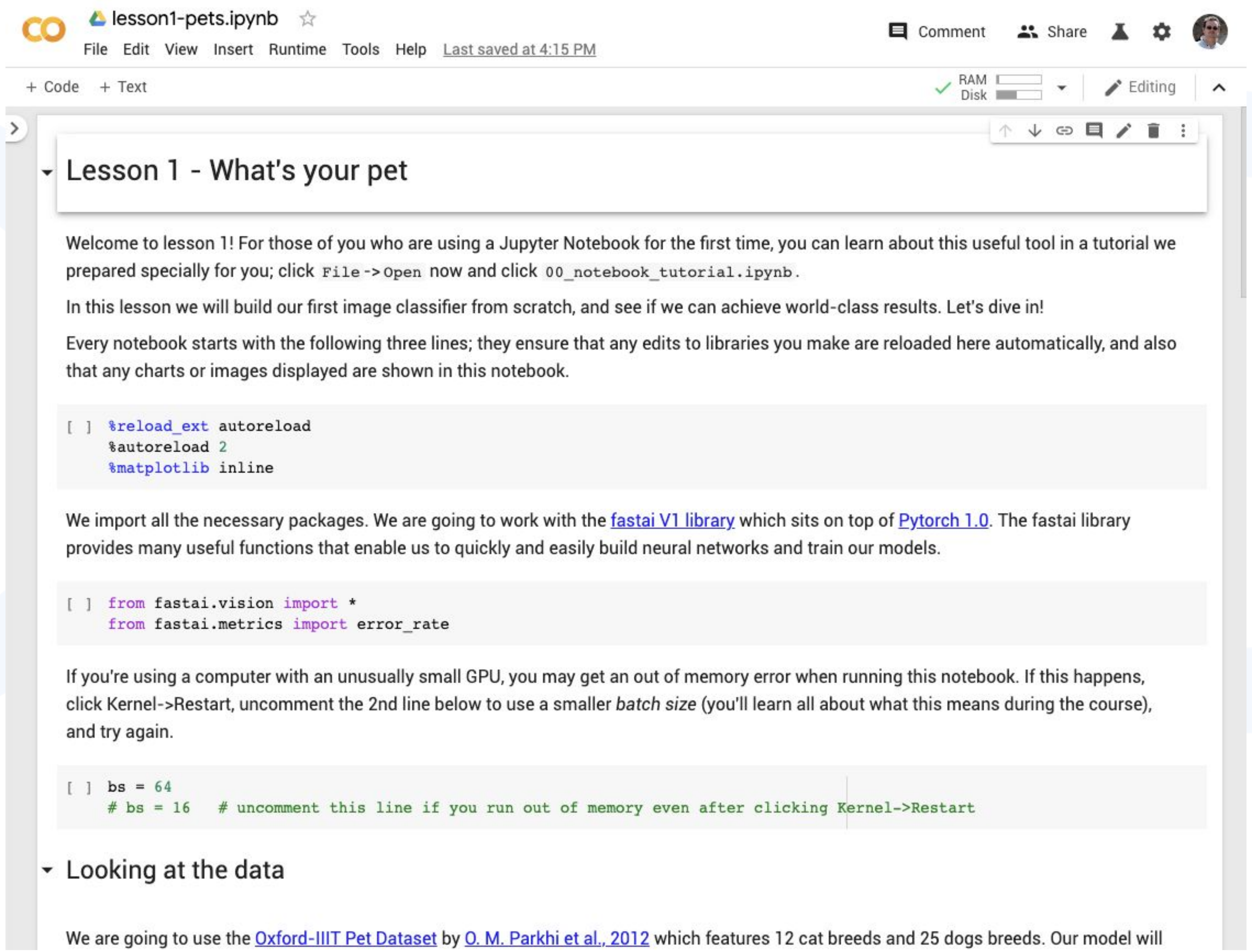
Amazon SageMaker offers fully managed Jupyter Notebook (Jupyter Lab) running on a virtual instances with on board GPUs to easily allow model training and model inference on a variety of trained models



Neosperience

Jupyter Notebooks: an IDE for Machine Learning

Google Colab -- free managed Notebooks



Google Colab is a completely free managed Jupyter Notebook instance, where can select runtime type between GPU and TPU.

When runtime is released, or instance gets disconnected, the environment is reset and local storage is removed

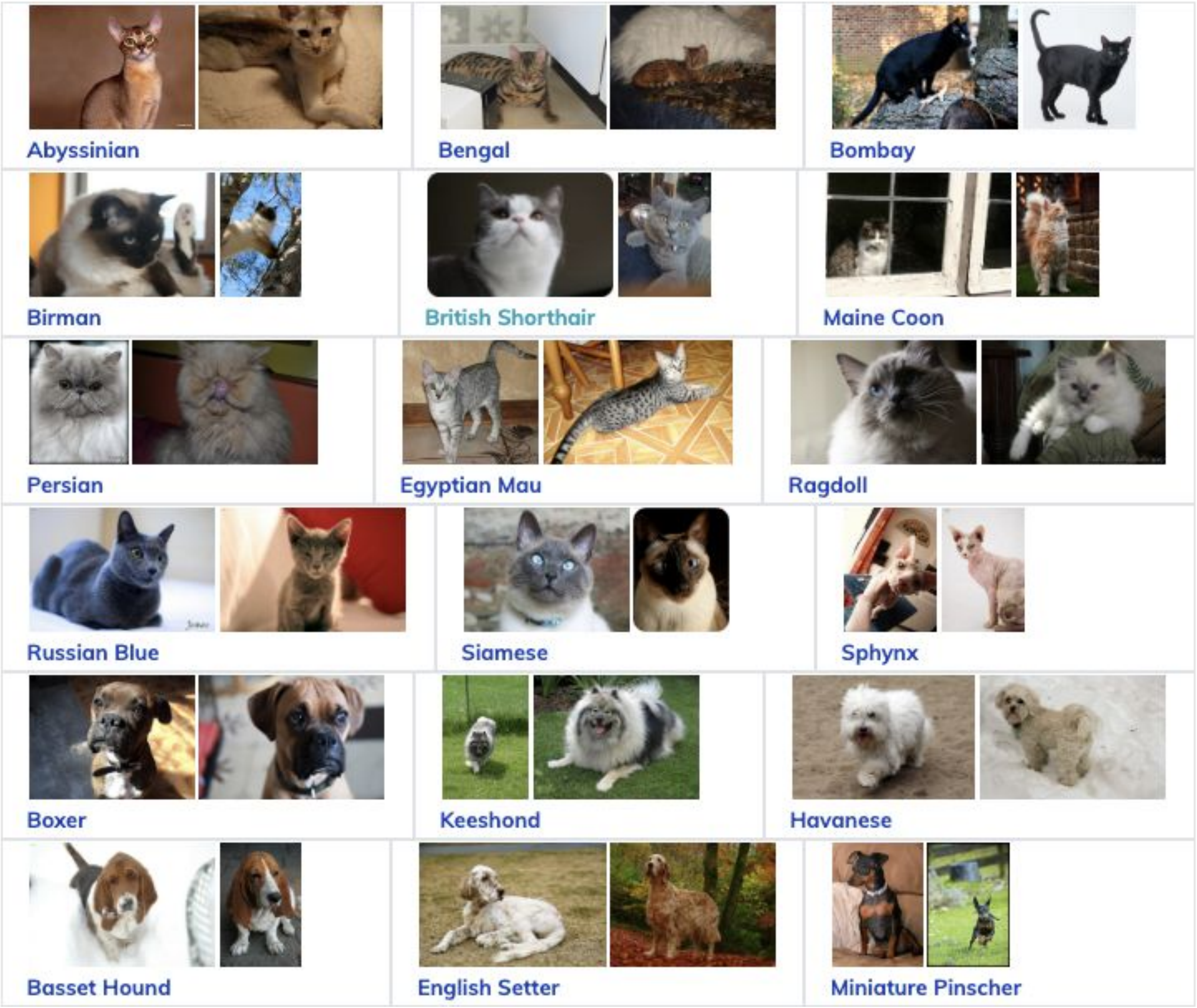
Problem: build a breed detector

We want to detect not only whether an image contains a cat or a dog, but also which breed is the pet pictured.

One of the most difficult tasks in computer vision was, until 2013 image classification: telling the difference between a dog and a cat has been one of the best benchmarks for a CNN.

Since 2016 the computing power of GPUs makes this problem too naive to be used as benchmark, so we moved to detecting the breed of the pet in a picture

http://www.robots.ox.ac.uk/~vgg/publications/2012/parkhi12a/parkhi12a.pdf



Neosperience

Step 1: Data Exploration

Never under estimate your intuition looking at the data. This phase is usually named data exploration and involves extracting some statistical figures.

The first thing we do when we approach a problem is to take a look at the data. We always need to understand very well what the problem is and what the data looks like before we can figure out how to solve it. Taking a look at the data means understanding how the data directories are structured, what the labels are and what some sample images look like.

Labels:

'Abyssinian', 'Bengal', 'Birman', 'Bombay', 'British_Shorthair', 'Egyptian_Mau', 'Maine_Coon', 'Persian', 'Ragdoll', 'Russian_Blue', 'Siamese', 'Sphynx', 'american_bulldog', 'american_pit_bull_terrier', 'basset_hound', 'beagle', 'boxer', 'chihuahua', 'english_cocker_spaniel', 'english_setter', 'german_shorthaired', 'great_pyrenees', 'havanese', 'japanese_chin', 'keeshond', 'leonberger', 'miniature_pinscher', 'newfoundland', 'pomeranian', 'pug', 'saint_bernard', 'samoyed', 'scottish_terrier', 'shiba_inu', 'staffordshire_bull_terrier', 'wheaten_terrier', 'yorkshire_terrier'

Step 2: Data Cleaning

Remove outliers or unwanted data.

In a real-life scenario data has not been prepared into a dataset for your convenience, but needs to be converted, normalized and cleaned. Often datasets contain images that are blurred, too dark or simply wrong.

Finding the right amount of data needed for a classificator

- how different are the classes that you're trying to separate?
- how aggressively can you augment the training data?
- can you use pre-trained weights to initialise the lower layers of your net?
- do you plan to use batch normalisation?
- is dataset balanced or unbalanced?

A thumb rule would be starting with thousands of images, then extending your dataset as soon as more data is required (i.e. error stops going down)

Step 3: Data Augmentation

If your model needs to be able to work with practical images, you need to “augment” the batch set with rotations, skews and different sizes.

All modern frameworks allow for dataset creation with augmentation techniques zooming, flipping and rotating images. This makes your model robust to these transforms: the network learns how to classify a pet also if the image is not perfectly captured or gets distorted for any reason.

More transforms your add, more images and training time you need.

```
??get_transforms
```

```
Signature: get_transforms(do_flip:bool=True, flip_vert:bool=False, max_rotate:float=10.0, max_zoom:float=1.1, max_lighting:float=0.2, max_
warp:float=0.2, p_affine:float=0.75, p_lighting:float=0.75, xtra_tfms:Union[Collection[fastai.vision.image.Transform], NoneType]=None) ->
 Collection[fastai.vision.image.Transform]
```

Step 4: Training

Choose your network architecture, a loss function and an error metric

Many CNN models come already pre-trained into Pytorch or Keras. Using a pre-trained model and specializing the network on our dataset is often called **transfer learning**. Finding a good metric is important to tell whether our model is overfitting a dataset (loss functions goes down, error goes up).

Some metrics are already built in, such as MSE, RMSE. FBeta, etc.

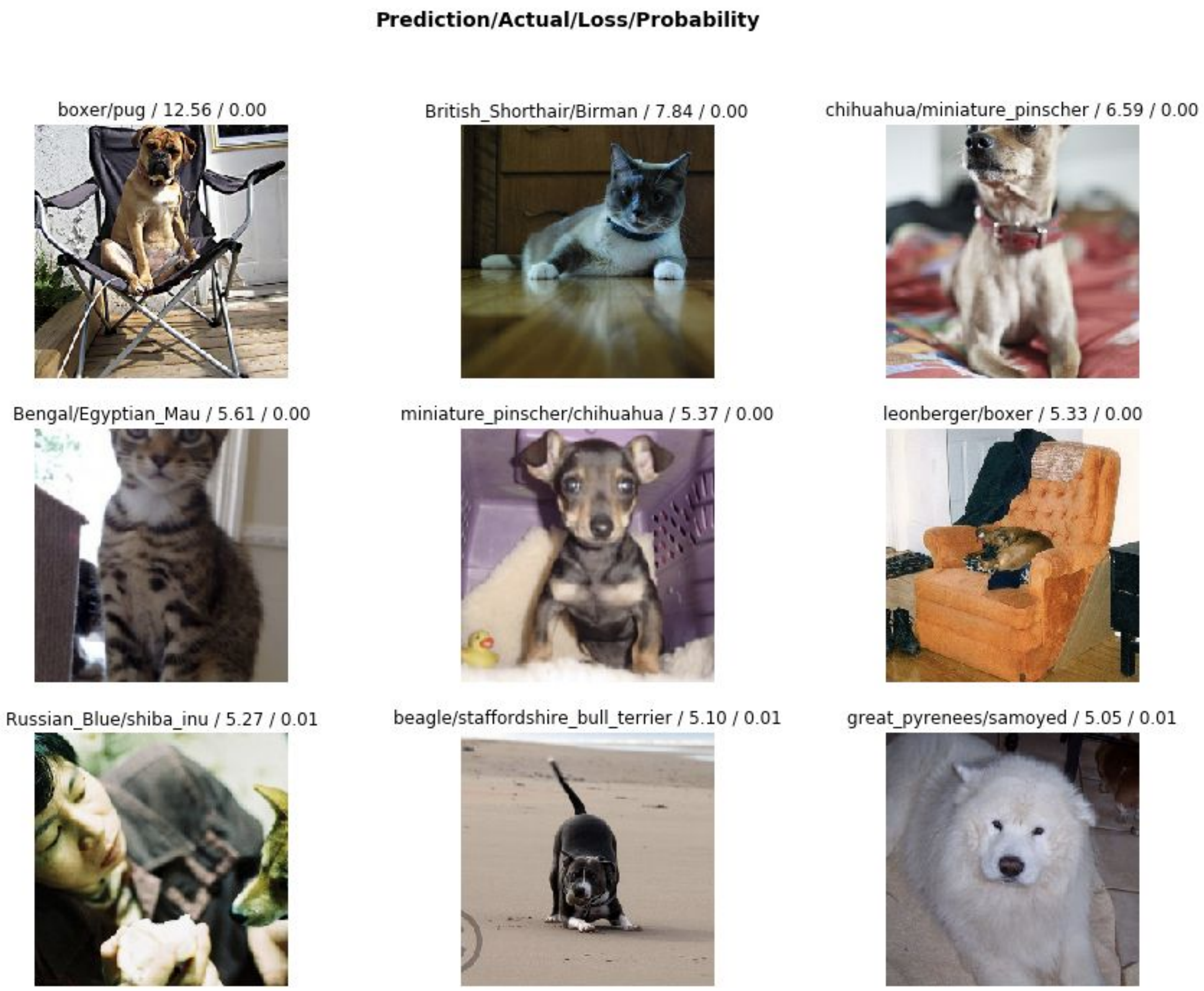
```
learn = cnn_learner(data, models.resnet34, metrics=error_rate)
```

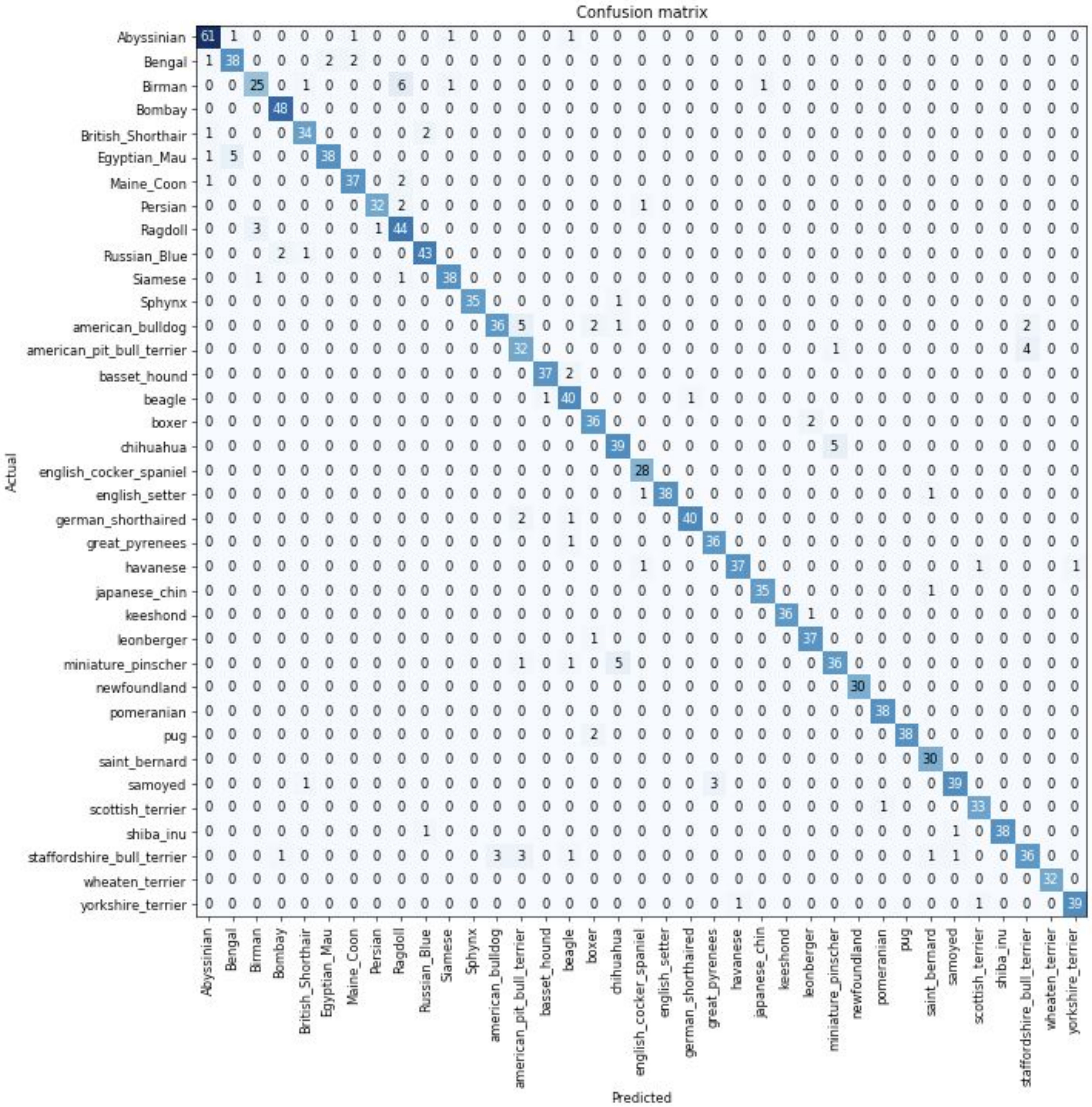
```
learn.fit_one_cycle(epocs)
```

epoch	train_loss	valid_loss	error_rate	time
0	1.427408	0.359479	0.117050	01:03
1	0.615426	0.274810	0.086604	00:57
2	0.390386	0.255367	0.085927	00:57
3	0.282586	0.237271	0.073748	00:58

Step 5: Evaluation

Evaluate results. Improve. Rinse. Repeat.





Neosperience

Breeds can be accurately recognized!

course.fast.ai

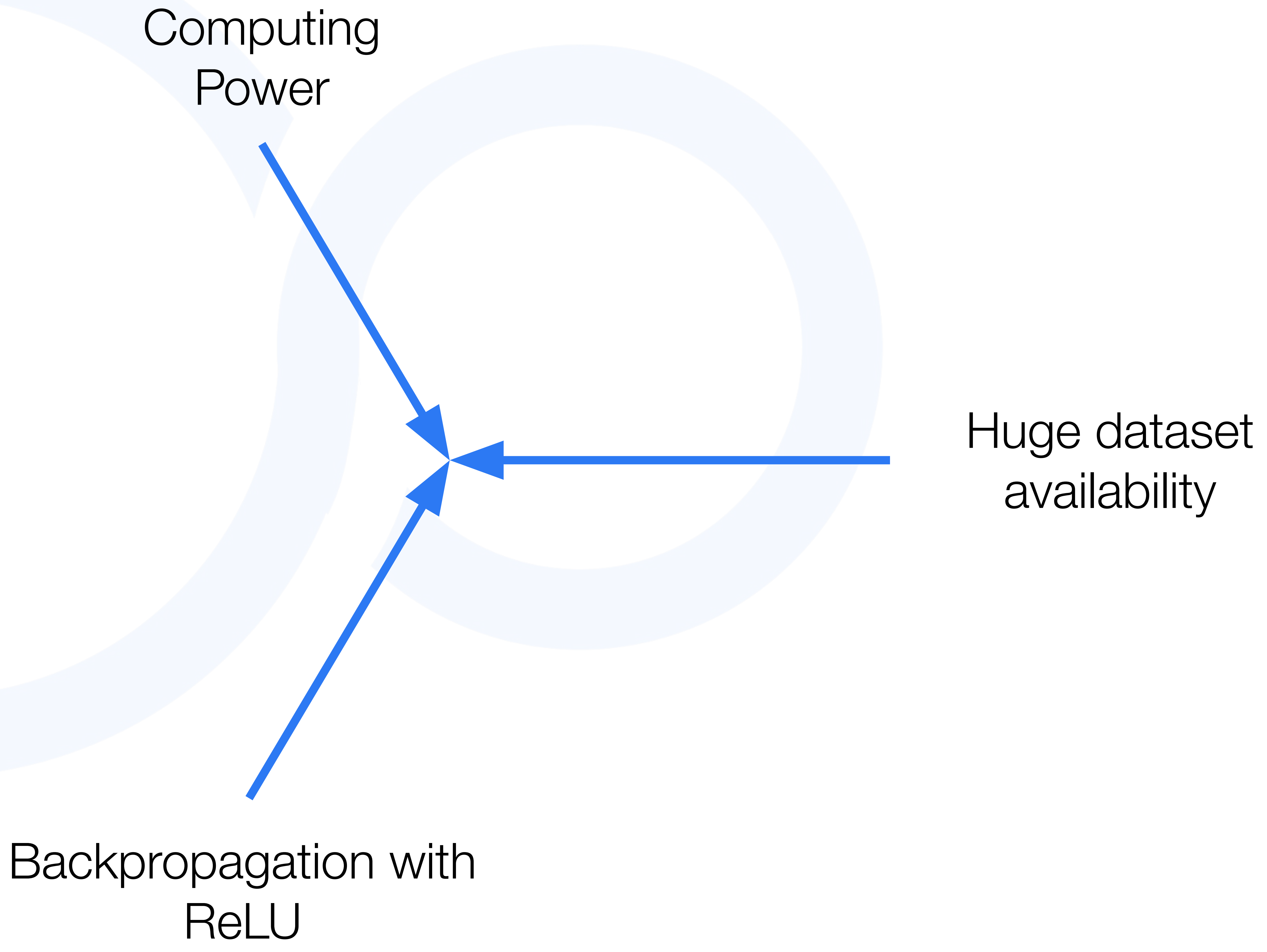
Part 2

Industrial Applications

Neosperience

Why now?

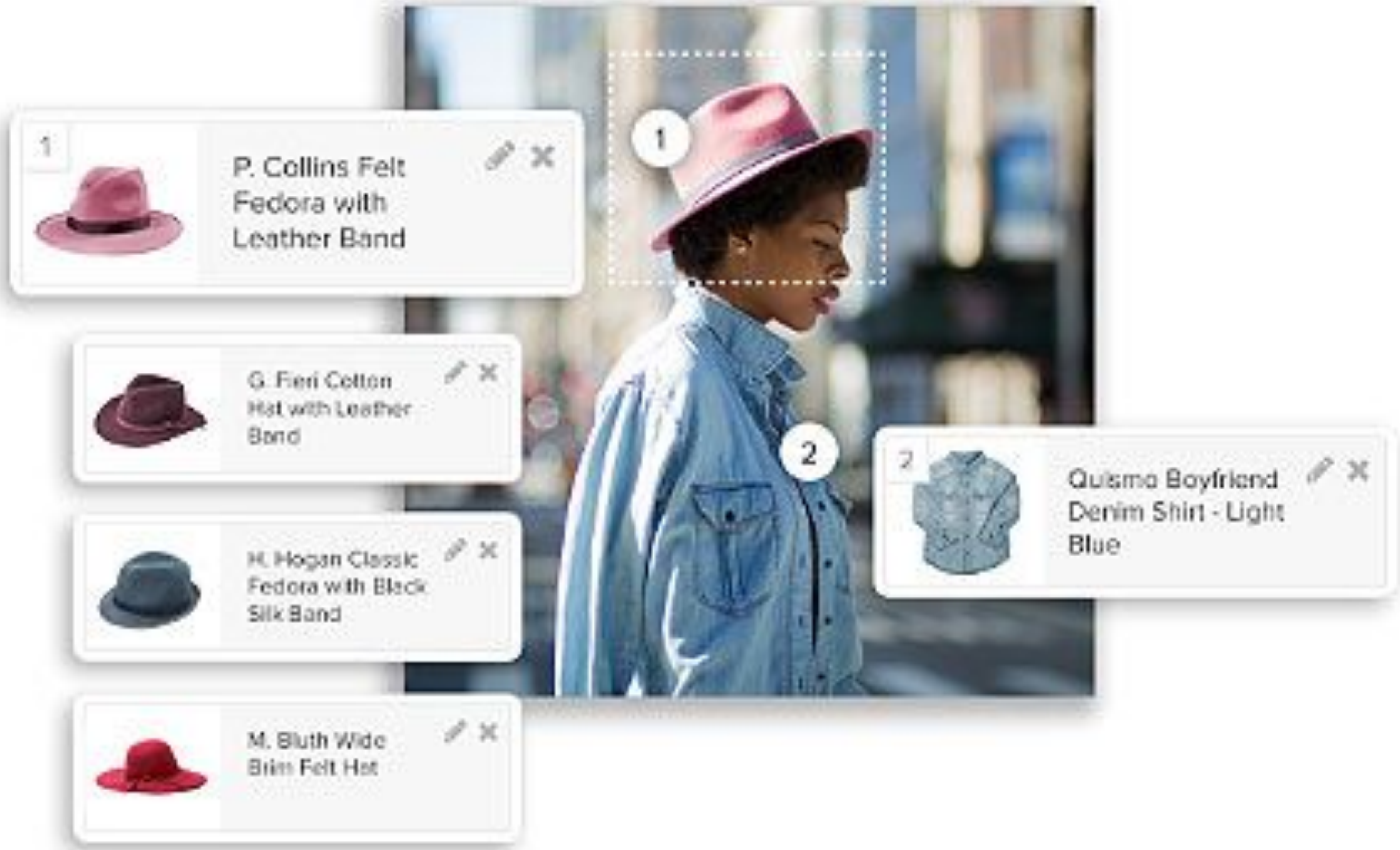
Even if algorithms have been known for more than three decades, today we’re at the nexus of converging opportunities



Product auto-tagging and visual search

Use plain ResNet or VGG with transfer learning to find products within images coming from catalogs or customer pictures.

- Automatically tag products
- Cut down on workload to categorize products
- Show related products
- Find cheaper version of high end products
- Find complimentary products
- Find products usage on social media



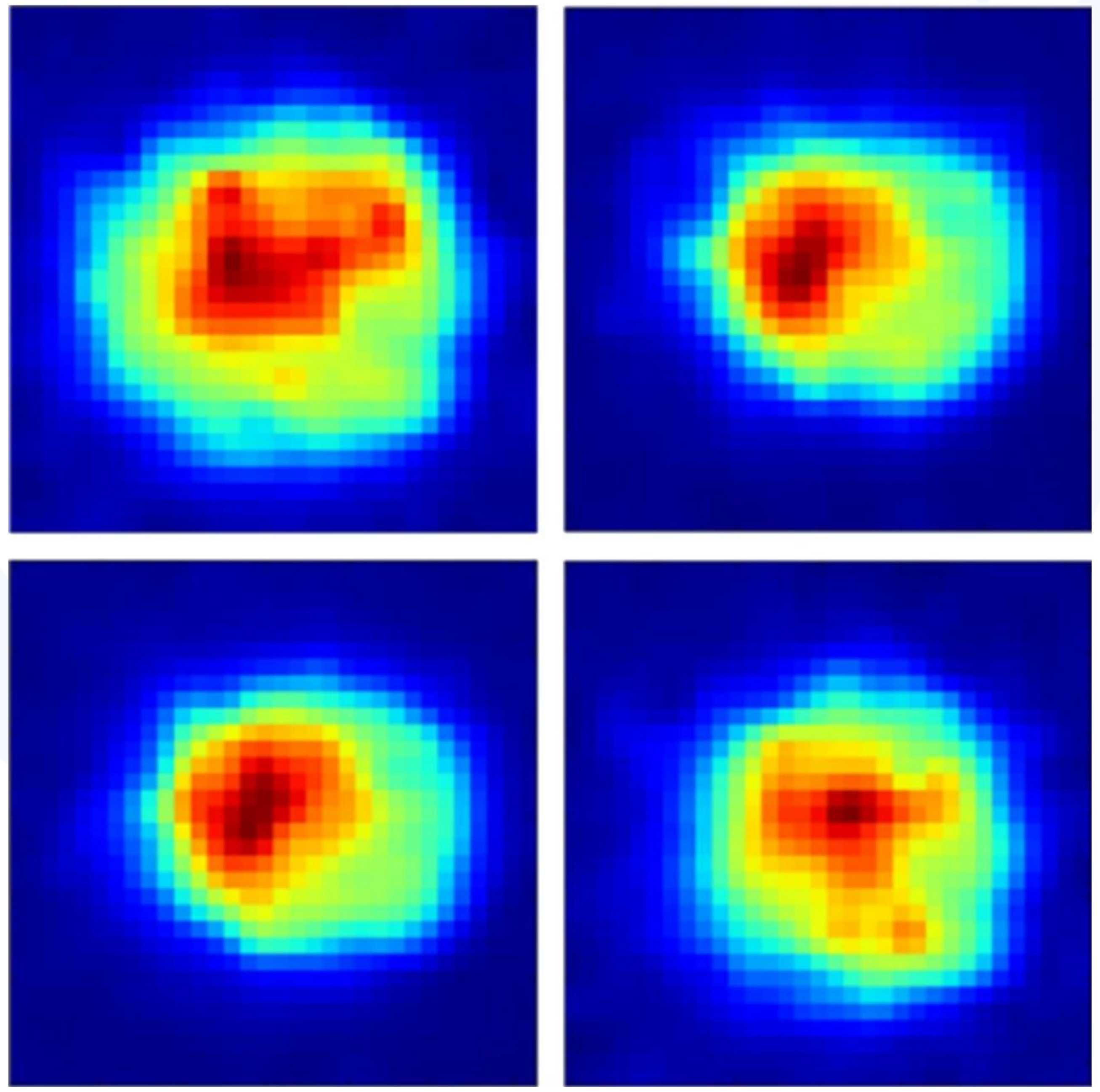
https://www.kaggle.com/paramaggarwal/fashion-product-images-dataset

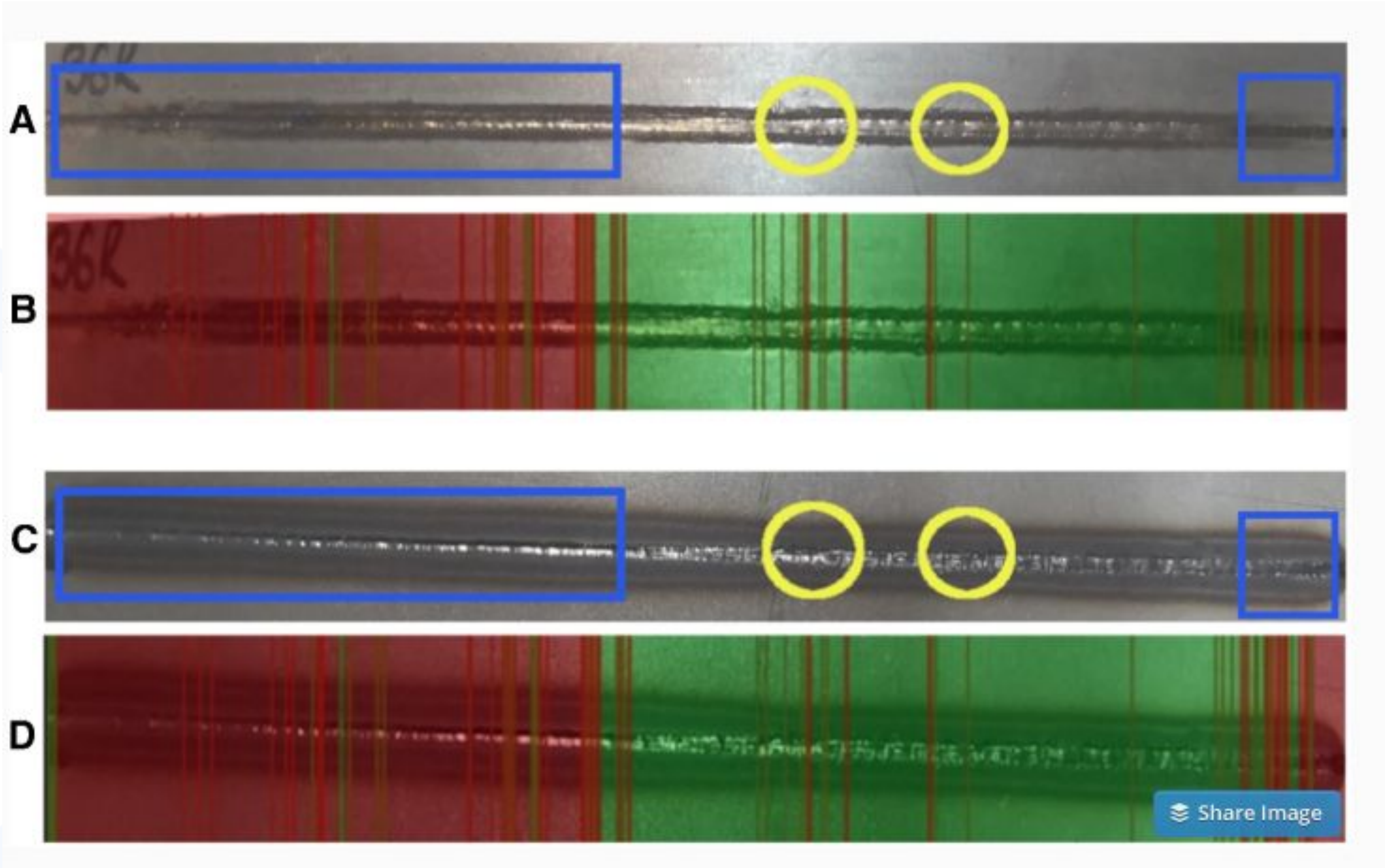
Neosperience

Quality assurance

Detect items not compliant with accepted sizes/shapes/colors.

CNNs approaches are capable of analysing MWIR thermal images to extract parameters of laser processes and quality indicators.





Real-time defect detection on a laser weld bead. a and c show two side views of the weld bead where the blue rectangles mark a defective section in the first and final segments due to undercuts and the yellow ellipses mark a region where some points have excessive porosity

Neosperience

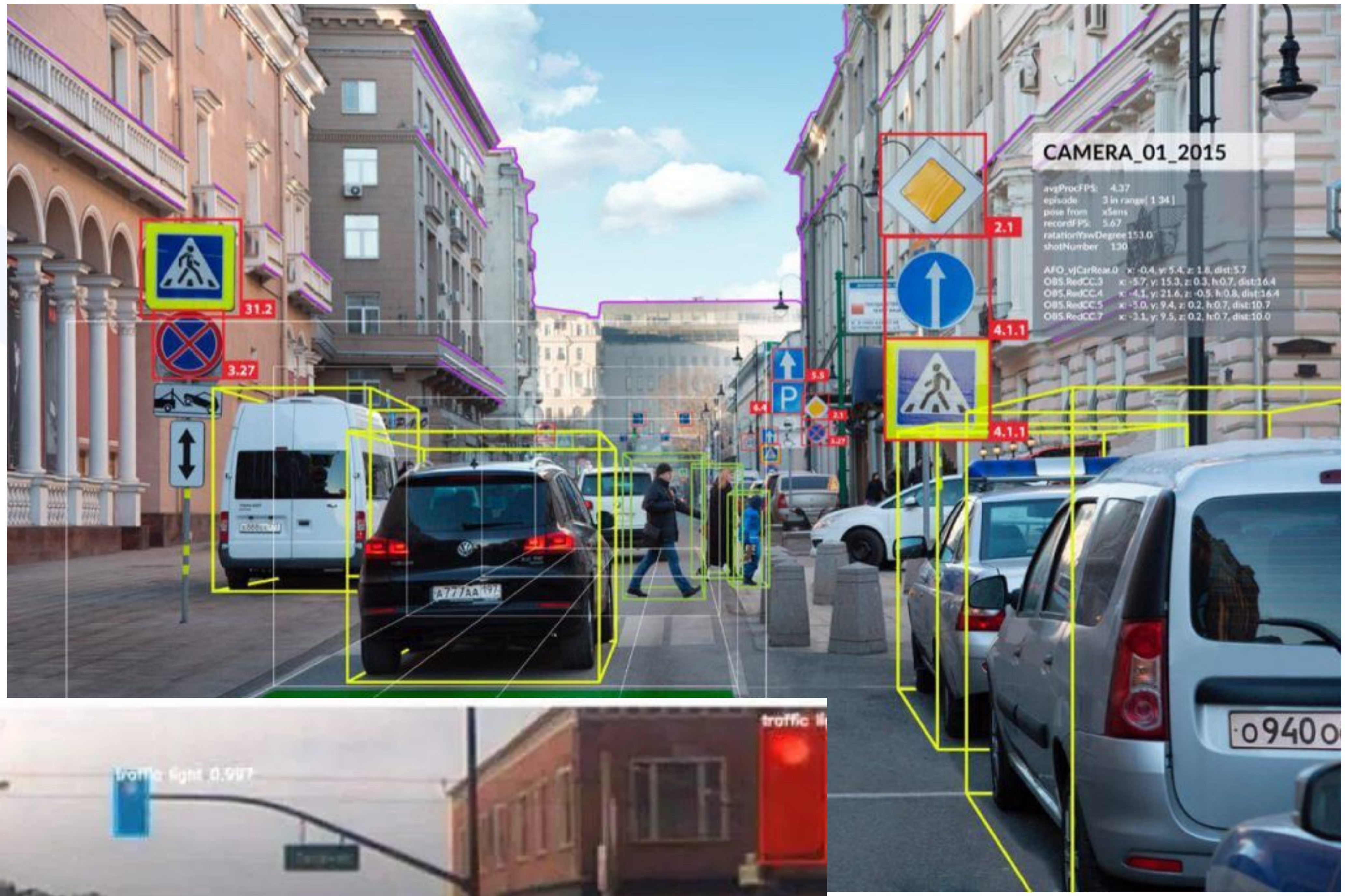
Self Driving cars

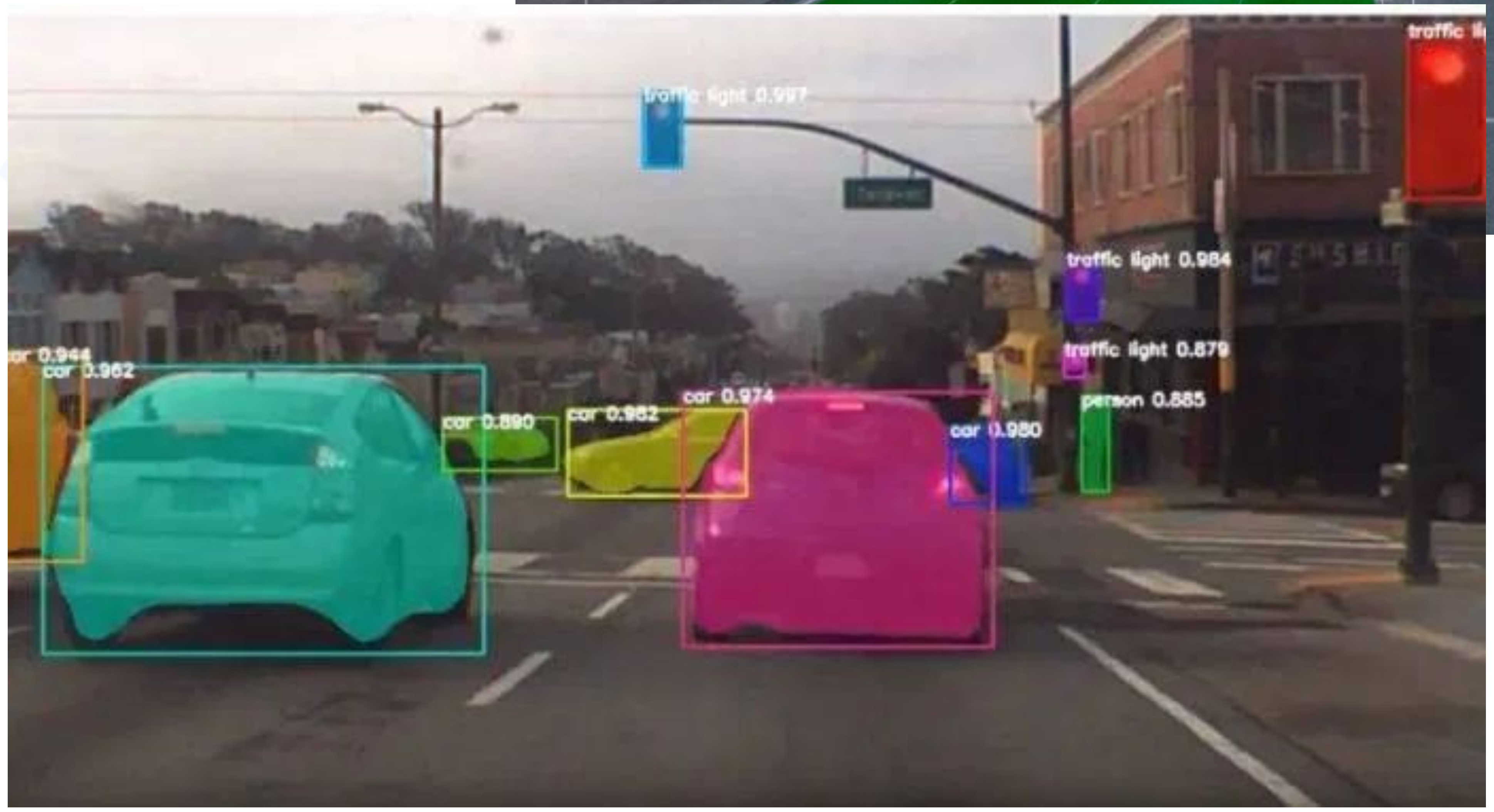
Deep usage in security: detect accesses to restrict areas, detect people unhealthy behavior or

Uses a model ensenble to leverage segmentation properties of CNNs.

CNNs to identify and segment, other ML models to track cars and respond to inputs

Lyft and Uber are experimenting self driving cars for public transportation in big cities such as Las Vegas.





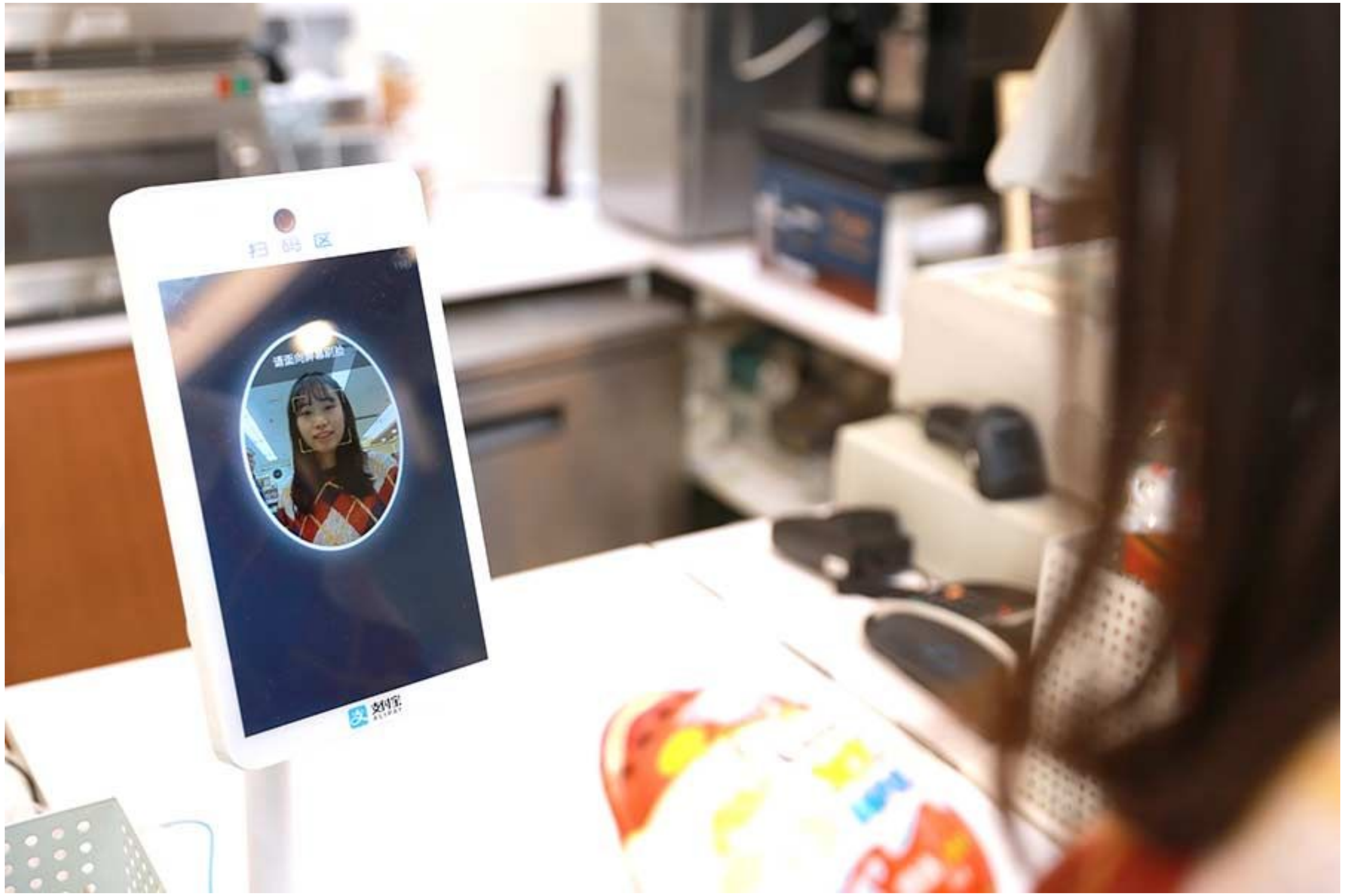
Payments using FaceID

Use customer face as key to unlock credit card informations in a third party store

Facebook Pay is experimenting payments with face recognition.

AliPay just updated its proprietary algorithm for face recognition to unlock payments in store and personalized advertising.

Libraries such as DLIB offer face embeddings extraction and recognition with an accuracy over 90%



DLIB a face recognition library

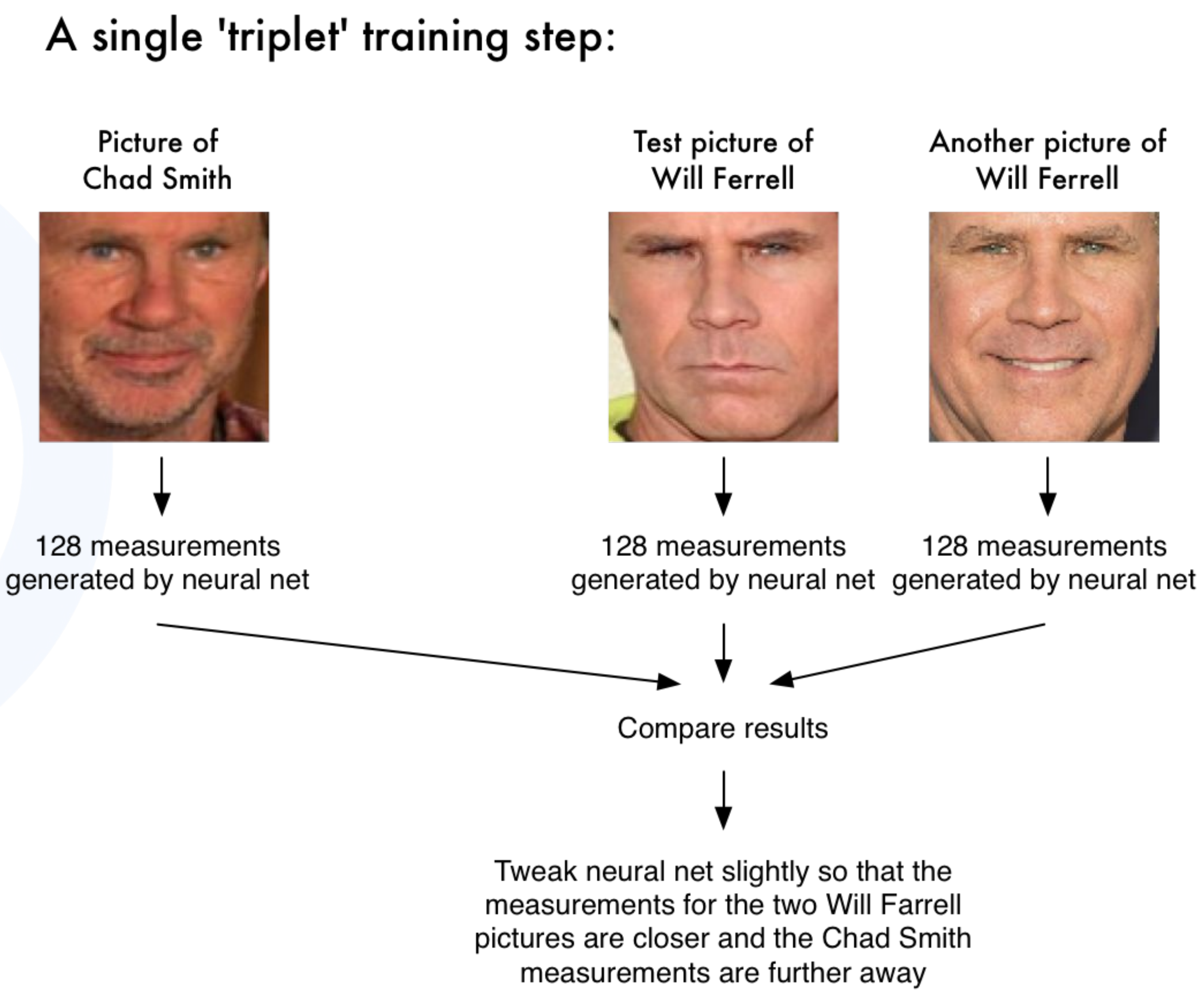
Multi-stage feature extraction and face rekognition. A CNN trained with triplet loss function

Sometimes we have to train a network not to recognize a given object, but to tell whether an image is or *is not* a given person of interest.

A common technique is to define a particular loss function named **Triplet Loss**.

DLIB network extracts landmarks from a face (named measurements), then trains a network wit a known image and two unknown different images.

This process makes the network able to understand differences between pictures of any face.



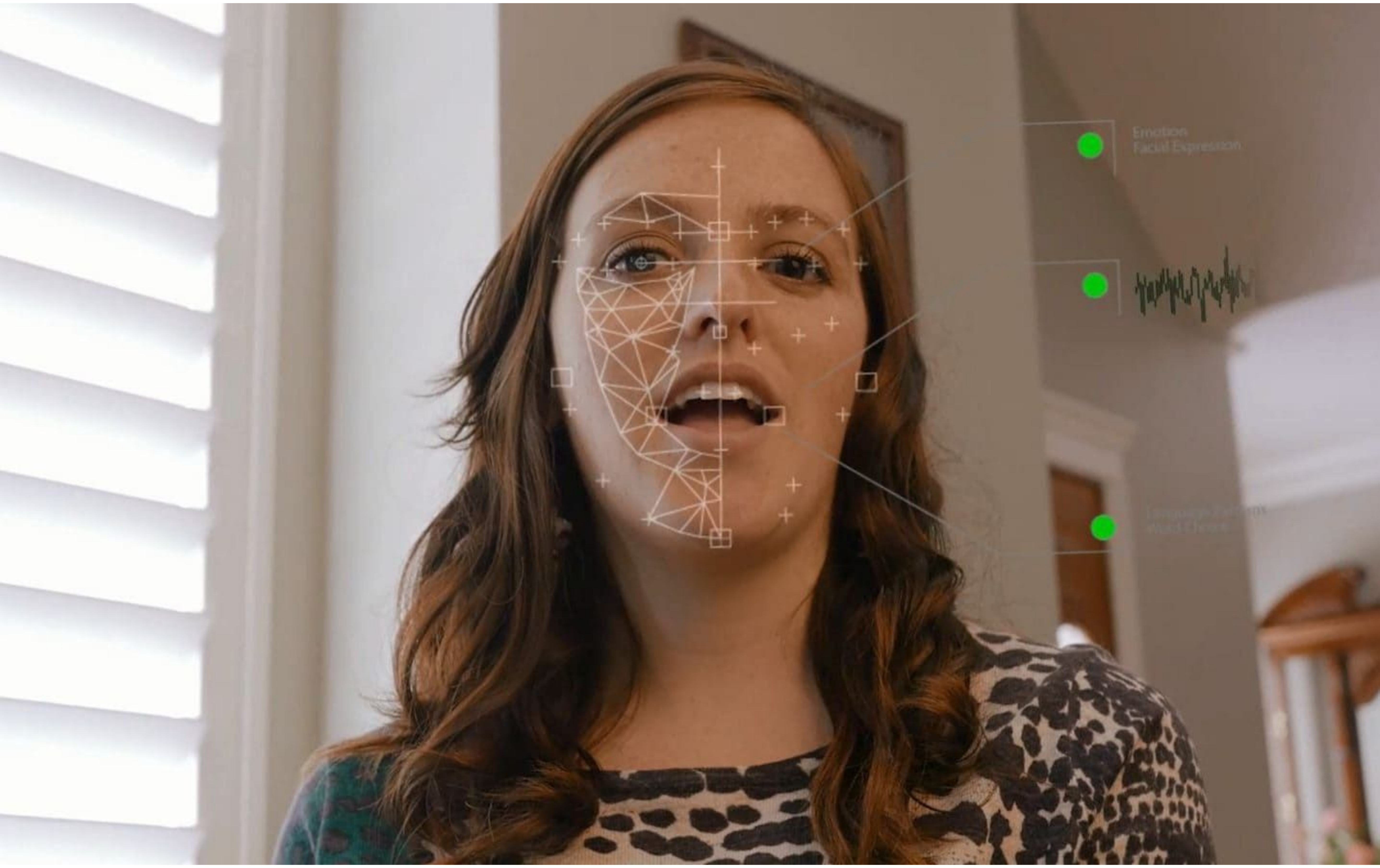
Neosperience

CNNs used in recruiting

AI used for first time in job interviews in UK to find best applicants

Unilever is among companies using AI technology to analyse the language, tone and facial expressions of candidates when they are asked a set of identical job questions which they film on their mobile phone or laptop.

The algorithms select the best applicants by assessing their performances.



CNNs in education

China is the current biggest investor on Computer Vision applications, with focus on schools and performance monitoring

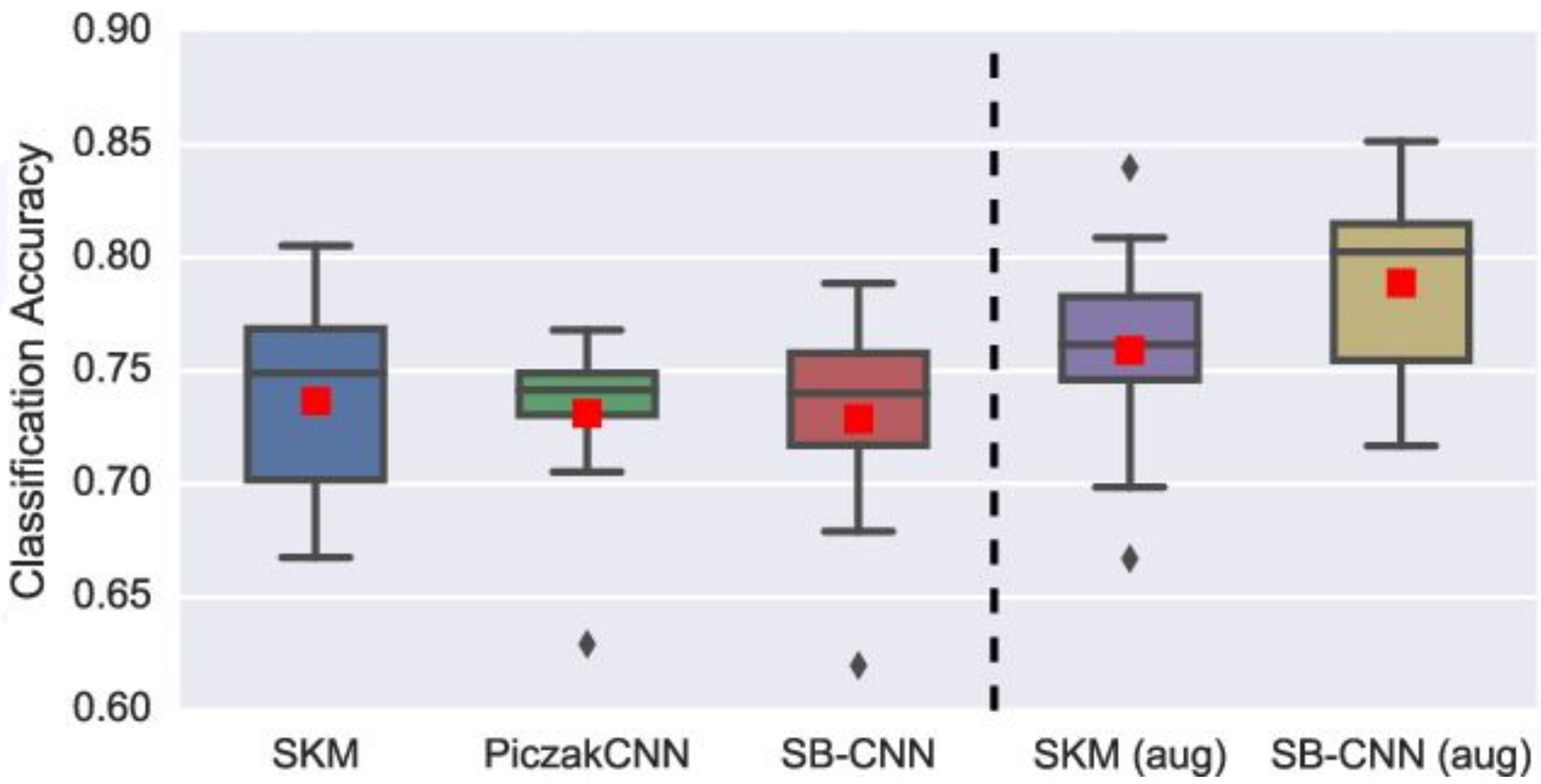
CNNs are used by China schools to monitor students attention and posture, thus avoiding injuries or being too distracted

https://youtu.be/JMLsHI8aV0g?t=52

Environmental Sound Classification

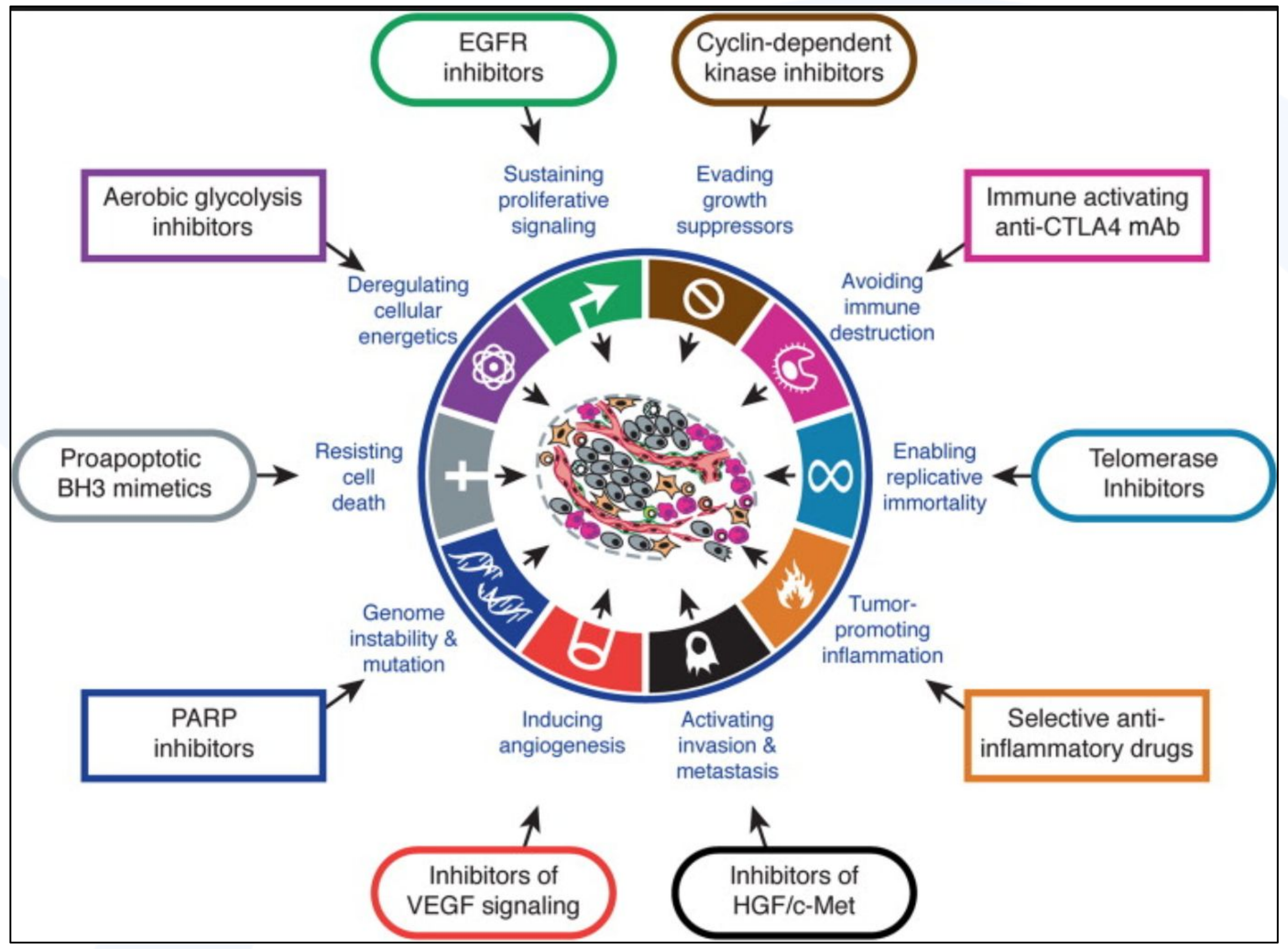
Use CNNs to classify different sounds in an open environment

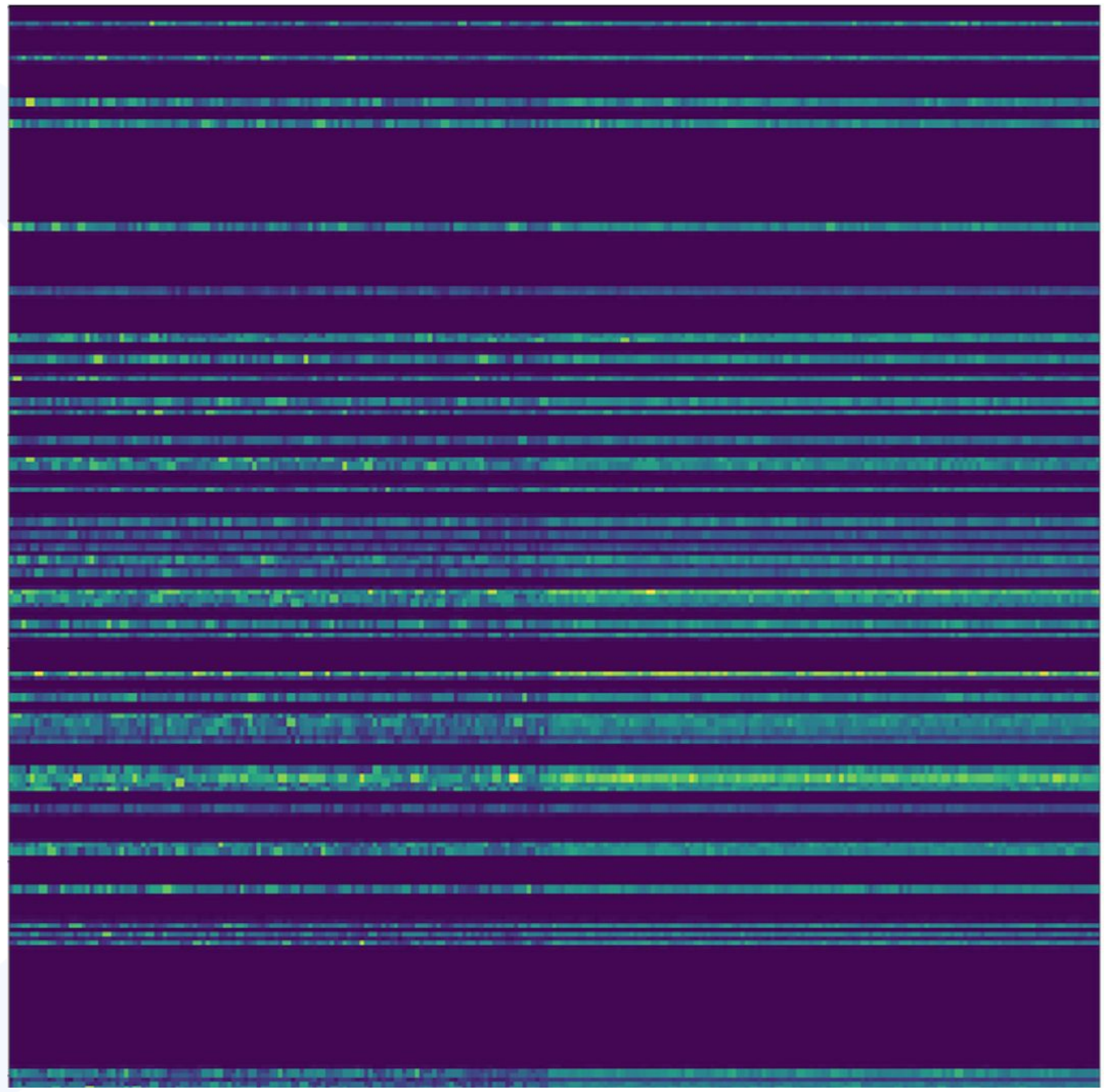
Represent sound frequencies as images, then classify different types of spectrum to better classify sounds in an environment

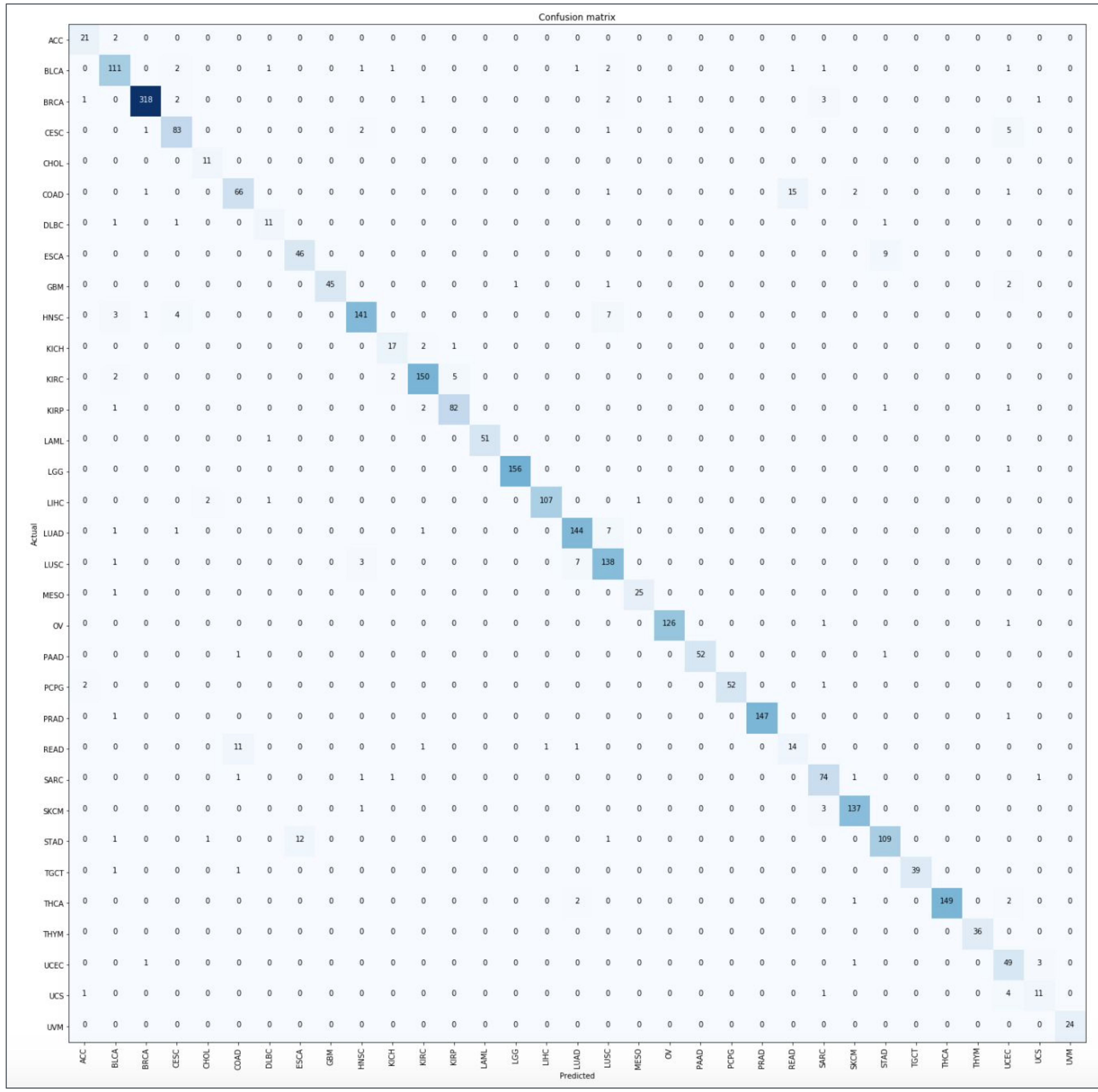


Neural Networks Applications in real life problems

Cancer Type Classification using CNN and Fast.AI







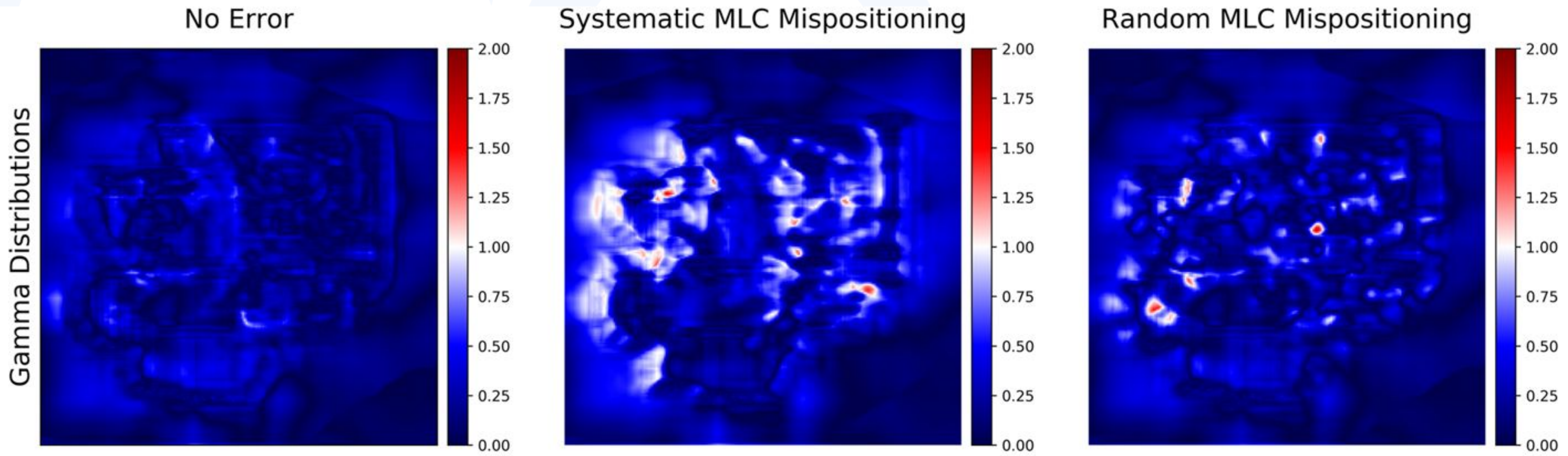
https://towardsdatascience.com/the-mystery-of-the-origin-cancer-type-classification-using-fast-ai-libray-212eaf8d3f4e

Neosperience

Quality assurance in radiotherapy

Deep learning for patient-specific quality assurance: Identifying errors in radiotherapy delivery by radiomic analysis of gamma images with convolutional neural networks

CNNs can be used to detect operational errors when exposing patients to radiotherapy and provide a better upfront correction of medical errors.



Deep dive on
Computer Vision Applications

powered by Neosperience

Neosperience

Neosperience
Image Memorability

Image Memorability — A business perspective

What is a memorability score?

Memorability is a measure of how much an image sticks into the memory of an average customer respect to average baseline images

A memorability score is a number representing memorability of an image, compared to the average capability of a human to remember an image which is 0.72

Images with a score higher than 0.72 have high memorability and are suitable for campaigns

Images with a score lower than 0.72 underperform and should be avoided because are not remembered

Neosperience

Image Memorability — A business perspective

A memorable image is a good image?

High memorability score is a good starting point, but using it to select an image could be too naive

More relevant than memorability itself is understanding which feature makes an image memorable

Assigning a score to each pixel of the image regarding its contribution to the resulting score

In this case memorability analysis outperforms humans because it is able not only to tell the score

but also to understand what makes this score

Neosperience

Image Memorability — A technical perspective

How to detect scores and heat maps?

Build an experiment to measure memorability (ground truth)

Deep Learning comes into help with CNNs

A CNN learns from experiment dataset how to estimate a memorability score

From a given inference, finding layer activations (through back propagation)

Convolutions and back propagation are compute intensive tasks that require GPUs even with inference

GPU inference is achieved through DeepLearning AMIs and on-premise instances

We needed an architecture to support inference through GPU in production in a scalable and cost effective way

Neosperience

Neosperience

https://image.neosperience.com

Alisea Visual Clean

Neosperience

Alisea — Transfer learning example

PROBLEM: Classify images of air duct/pipes as ‘dirty’ or ‘clean’

Step 1: Exploratory analysis

Dataset composed of hundreds of images of different air pipes, taken with different cameras, in different sizes.

Balanced dataset: 50% labelled ‘dirty’, 50% labelled ‘clean’.

RGB color channel.

Which images size to use? Which color channels?

Step 2: Data Cleaning

Choose which images are appropriate for your training dataset. Remove photos that would add 'noise'. In our case MANUALLY!

Considered image size:

- 128x128x3
- 256x256x3
- 320x320x3
- 480x480x3

Color channels:

- RGB, HSV

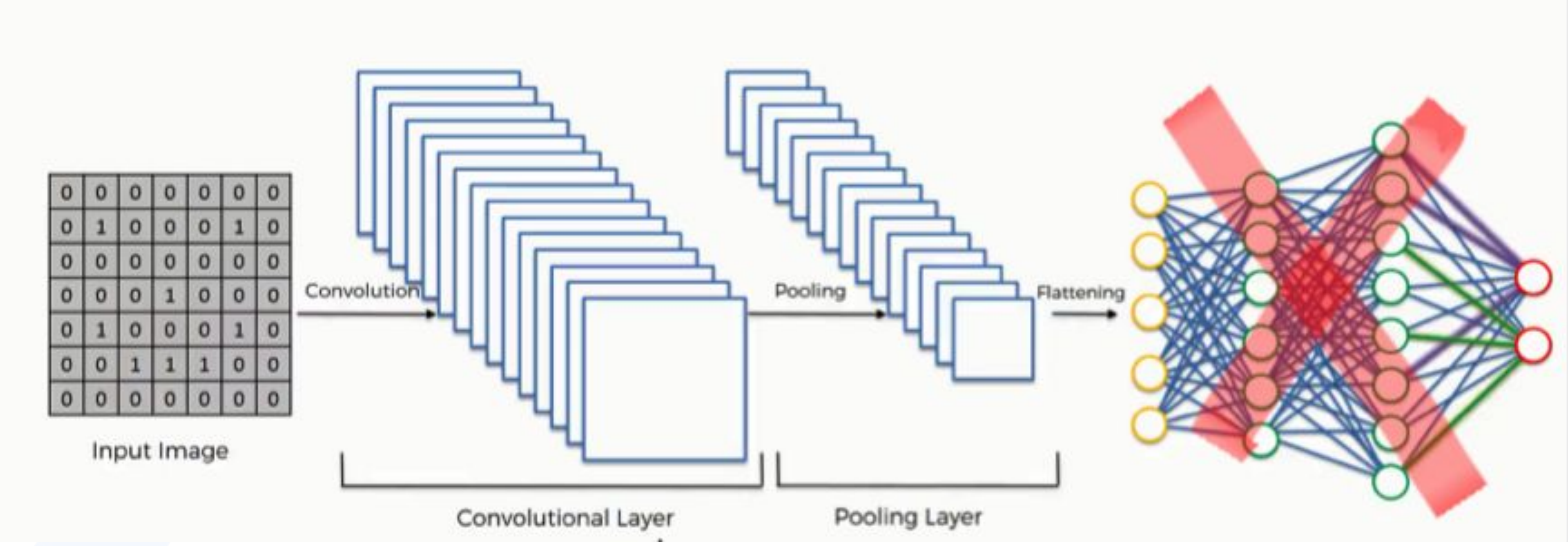
Not appropriate images for our dataset

Step 3 & 4: Data augmentation and training

Data augmentation to increase image size.

Keras and other libraries allow you to import already trained CNNs, downloading both pretrained weights and model architecture. Based on your need you can choose to keep the model as it is or:

- remove the fully connected (FC) layers at the end and add new layers that you need: Ex. final FC layer with more output classes.
- Keep all the weights or train them all over again



Considered CNN architectures:

- ResNet34, ResNet50, ResNeXt50

Trained several models using different image sizes to notice if there was a difference in our results.

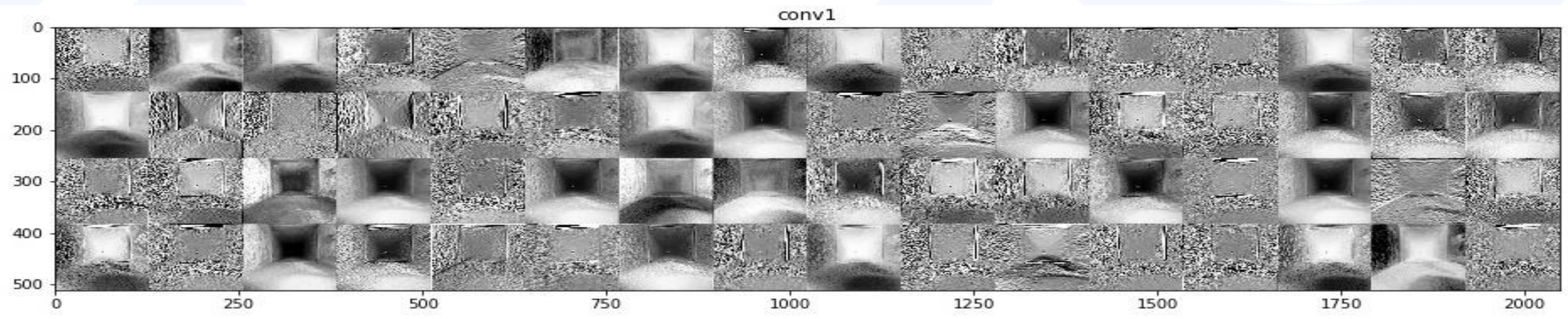
Best models in our case: ResNet50 and ResNeXt50

Best size: 256x256x3, bigger images need more computing power and longer training time

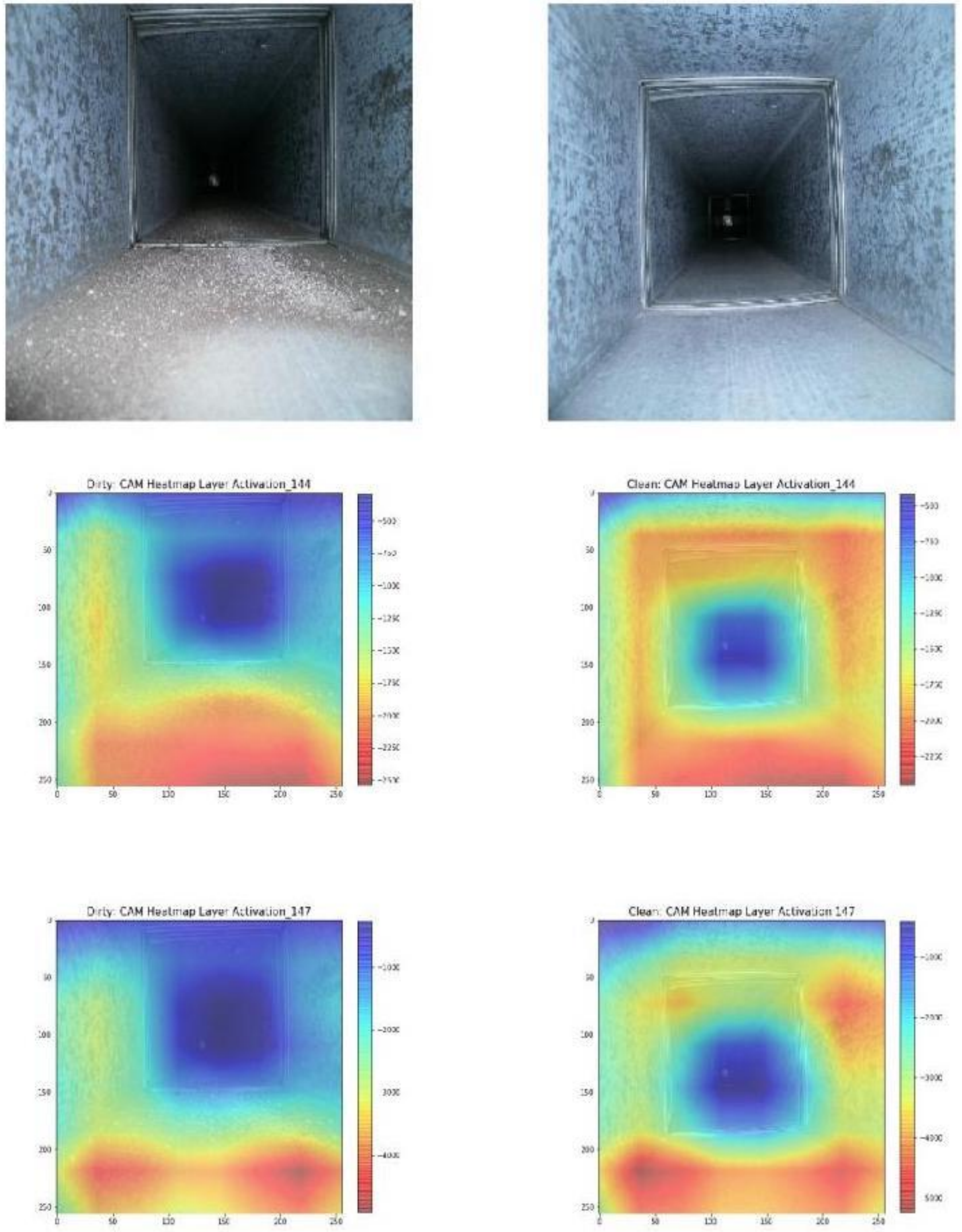
Best color channel: RGB

Final score: ~92% accuracy

What does the model see?



Feature Map of first Conv Layer



Attention Heatmap

Neosperience

Neosperience
People Analytics

Neosperience

Why count people in store?

People number is a KPI used to estimate ROI

- Understanding the number of people is considered a good way to estimate the average return of a given store
- The daily income of a store divided by the overall number of people detected gives a ROI
- Understanding high traffic stores can led to strategic decisions
- Low traffic or lower ROI can be closed or moved

People counting is broken

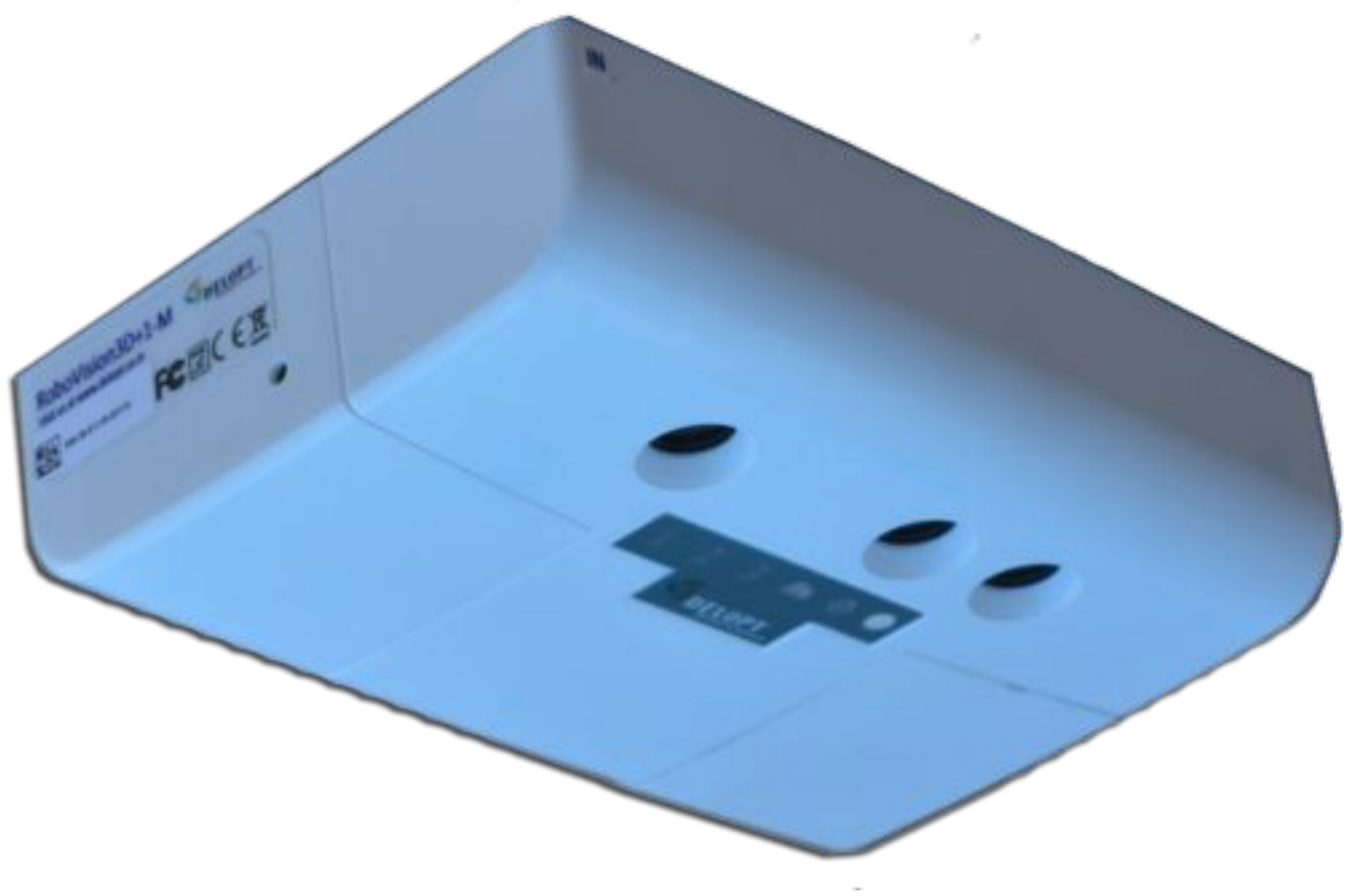
Commonly used devices are flawed and lack accuracy

IR sensors are cheap but inaccurate

- no unique counting
- no information about what happens in store
- false positives

laser / thermal counters are expensive and do not provide relevant information except counting

- unique counting
- no information about what happens in store
- lesser false positives

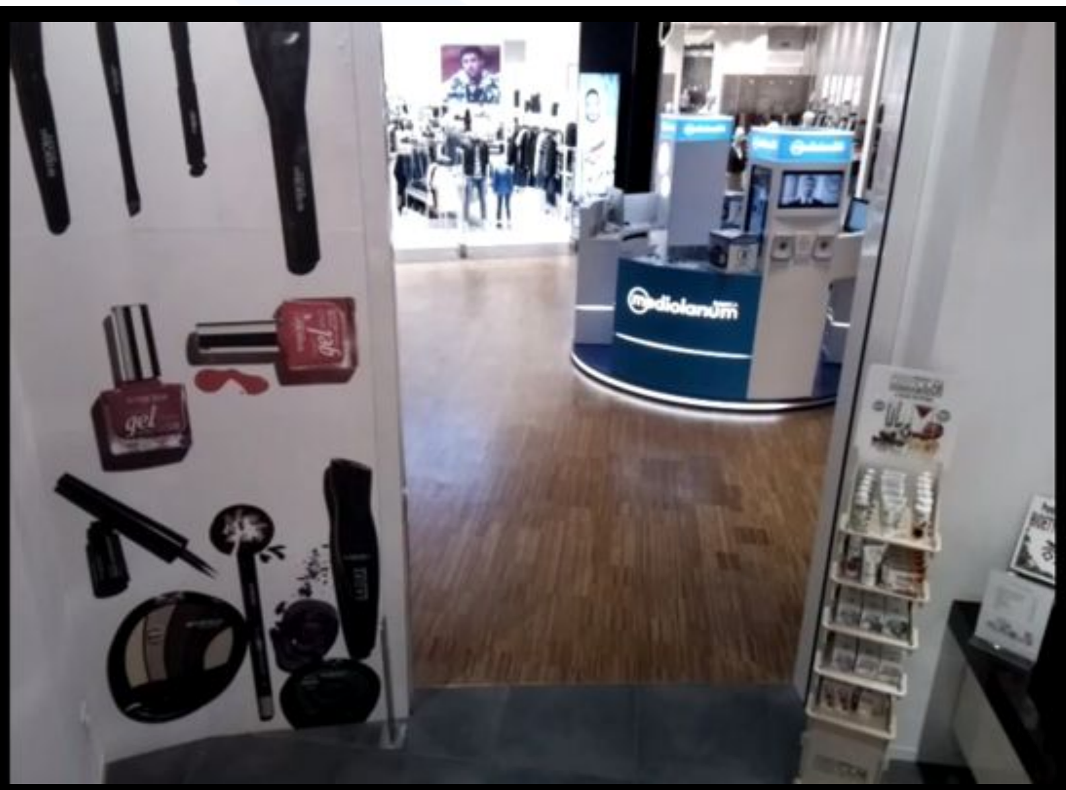


Introducing Neosperience Store Analytics

Detect relevant insights about your customers in stores using cameras

Neosperience Store Analytics is the SaaS solution to extract meaningful informations about people visiting stores in an accurate and reliable way

- Uses both standard cameras and dedicated hardware with a cost effective profile
- Dedicated Hardware is projected to optimise costs, heat management and reliability
- Stream acquisition is achieved in cloud
- Allows for multiple people counting, detects unique visits
- Enables advanced insights extraction



Results: people heatmaps, trajectories, insight

Mapping people presence within a given area of interest

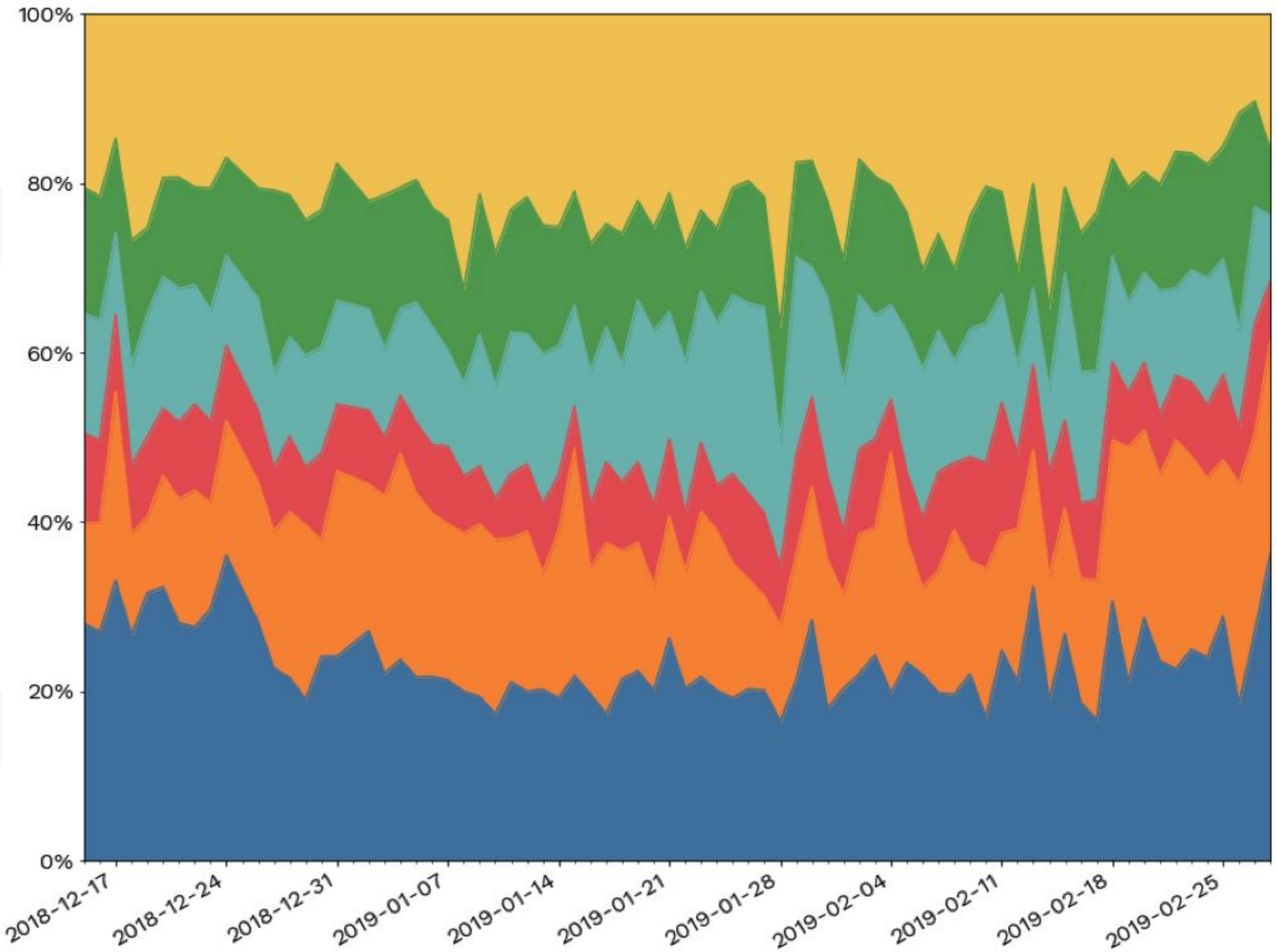
Being able to recognise people and track their movements in front of a camera leds to interesting results not only related to people counting

- Store managers can obtain a clear view of the preferred areas inside a store
- And event the overall amount of people that do not enter the store
- Store Analytics over delivered about store understanding, delivering a different but more meaningful metric

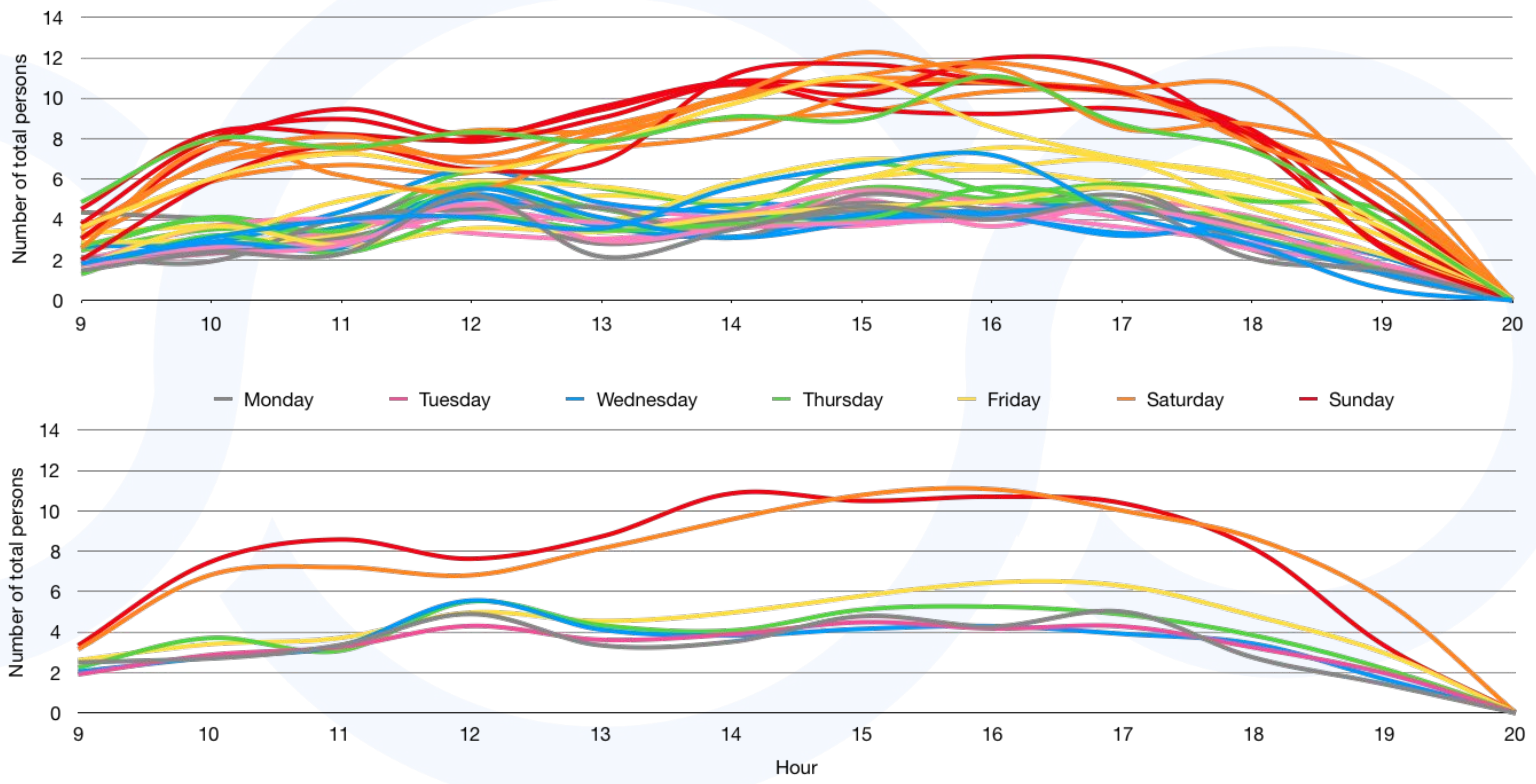
Neosperience

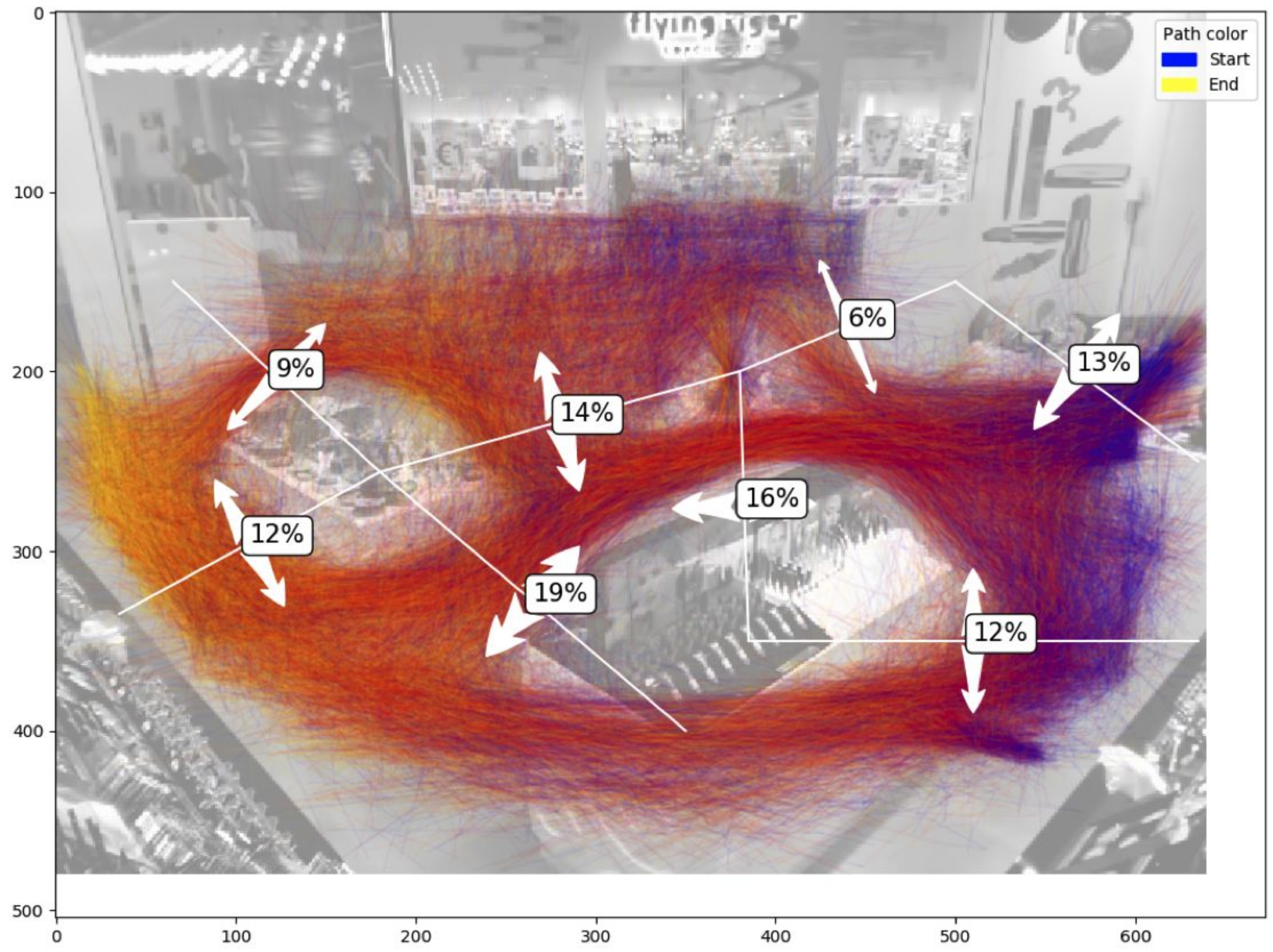
Results


1
36.0%
2
15.9%
3
8.8%
4
10.7%
5
11.5%
6
17.0%


100%
80%
60%
40%
20%
0%
2018-12-17
2018-12-24
2018-12-31
2019-01-07
2019-01-14
2019-01-21
2019-01-28
2019-02-04
2019-02-11
2019-02-18
2019-02-25

Results





Neosperience

Neosperience Psychographics

Detecting Psychographic traits

Understand customer behavior based on their shared content on social networks

Locus of Control
Belief to have control over the outcome of events in our lives, as opposed to external forces beyond our control.

Sensation Seeking
Search for experiences and feelings that are varied, novel, and intense, and the readiness to take risks for the sake of such experiences.

Agreeableness
General concern for social harmony. Agreeable people are considerate, kind, generous, trusting and trustworthy, helpful, and willing to compromise their interests with others.

Need for Uniqueness
Pursuit of differentness relative to others that can be achieved through the acquisition, utilization, and disposition of consumer goods.

Need to Belong
Human emotional need to belong and be an important part of a social group greater than ourselves.

Need for Cognitive Closure
Motivation to find a firm answer to an ambiguous situation, enhanced by the perceived benefits of obtaining closure. Closed-mindedness, intolerance of ambiguity, preference for order, and predictability.

Openness to Experience
General appreciation for art, emotion, adventure, unusual ideas, imagination, curiosity, and variety. Tendency to be more creative, unconventional, and aware of our inner feelings.

Conscientiousness
Preference for planned rather than spontaneous behavior, the tendency to display self-discipline, act dutifully, and strive for achievement against measures or outside expectations.

Impulsiveness
Tendency to act on a whim, displaying behavior characterized by little or no forethought, reflection, or consideration of the consequences.

Extraversion
Pronounced engagement with the external world. Extroverts enjoy interacting with people and tend to be enthusiastic, they possess high group visibility, like to talk, and assert themselves.

Need for Cognition
Extent to which we are inclined towards effortful cognitive activities.

Need for Affect
Motivation to approach or avoid emotion-inducing situations.

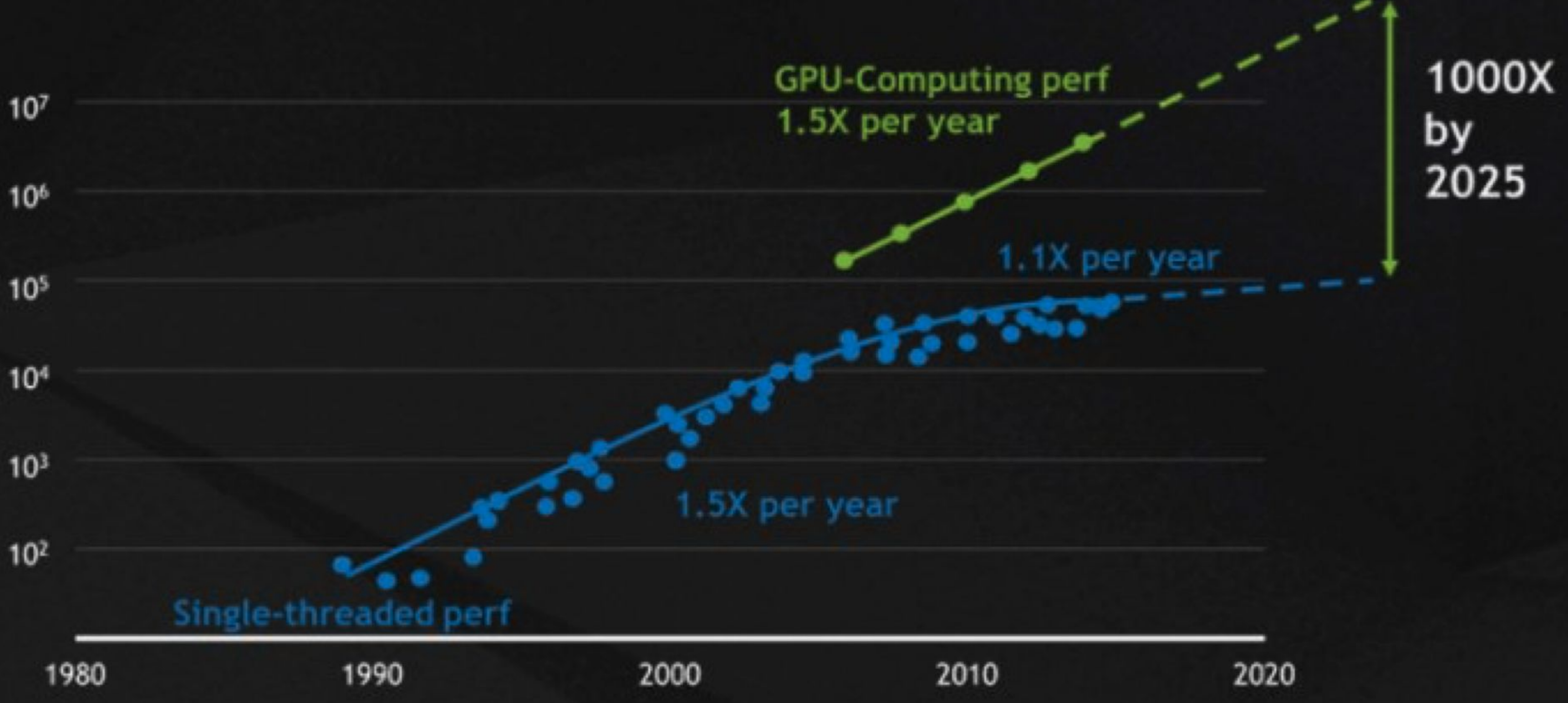
Narcissism
Self-centredness, craving for admiration, and grandiose view of one's talents.

Self Efficacy
Belief in ours ability to succeed in specific situations or accomplish a task.

Emotional Stability
The tendency to be calm, even-tempered, and less likely to feel tense or rattled.

Hardware for Computer Vision

Neosperience

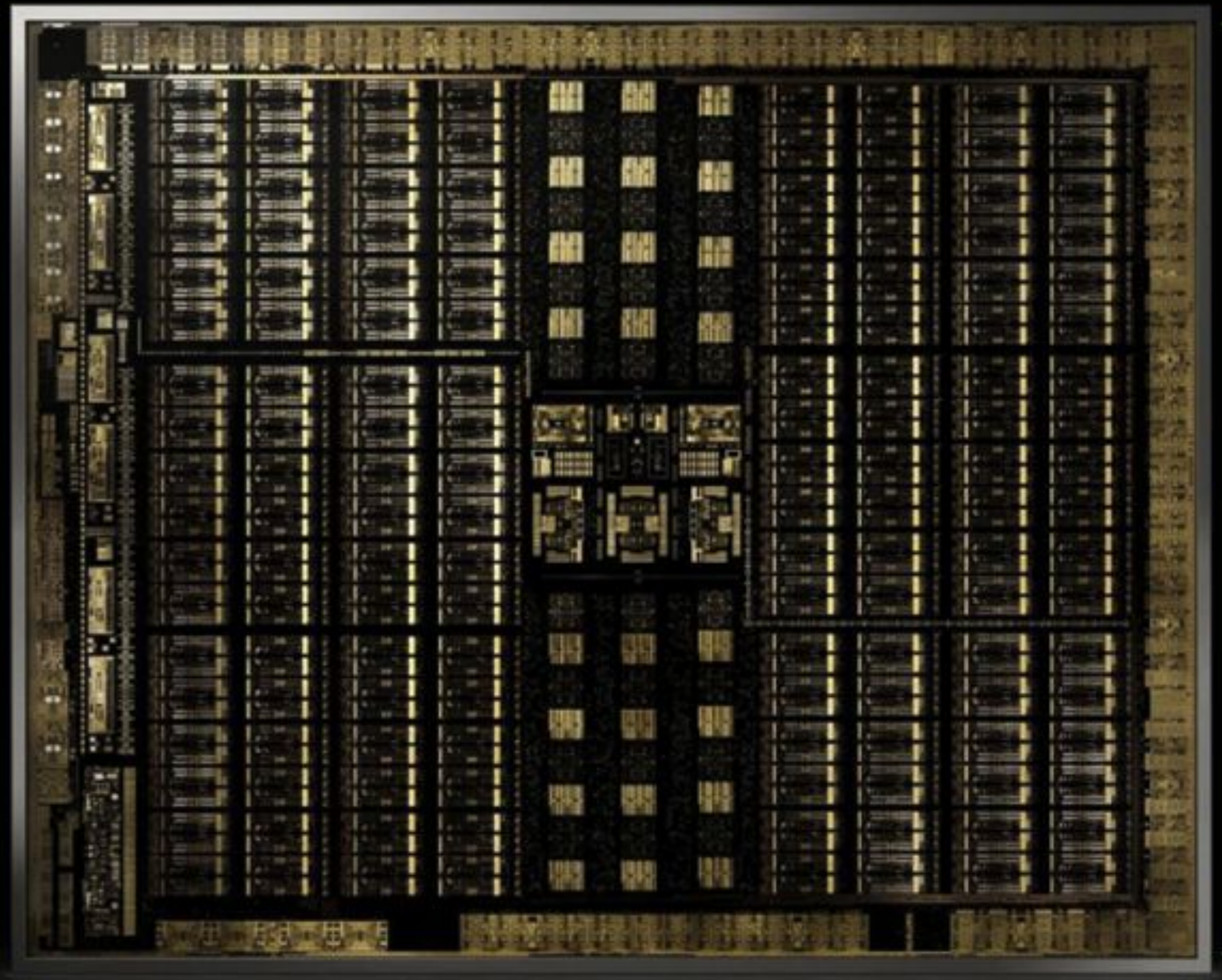


40 Years of Microprocessor Trend Data


RTX 2080Ti
GEFORCE RTX

TURING BUILT FOR RTX

Greatest Leap Since 2006 CUDA GPU



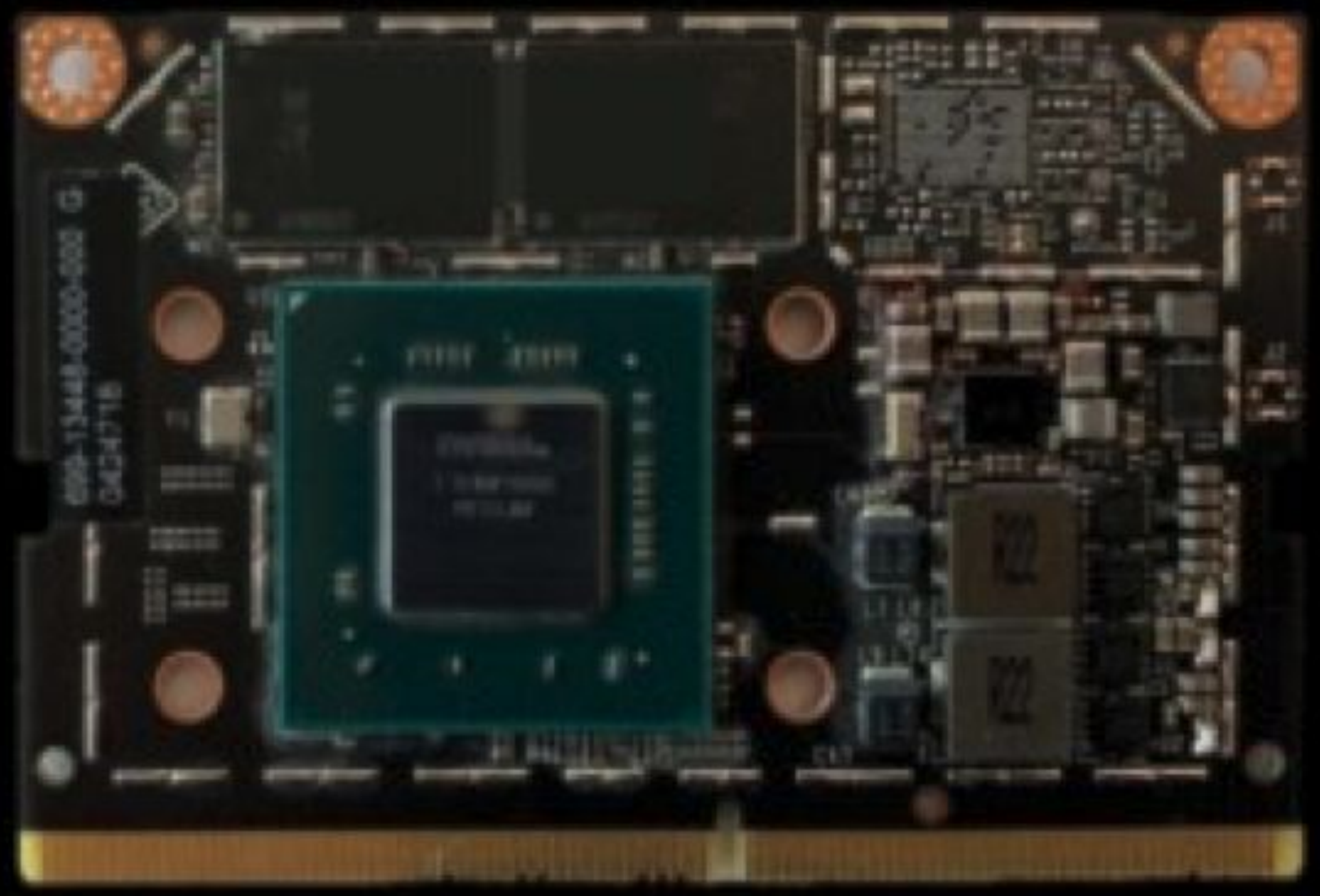
Turing SM
14 TFLOPS + 14 TIPS
Concurrent FP & INT Execution
Variable Rate Shading

Tensor Core
114 TFLOPS FP16
228 TOPS INT8
455 TOPS INT4

NGX

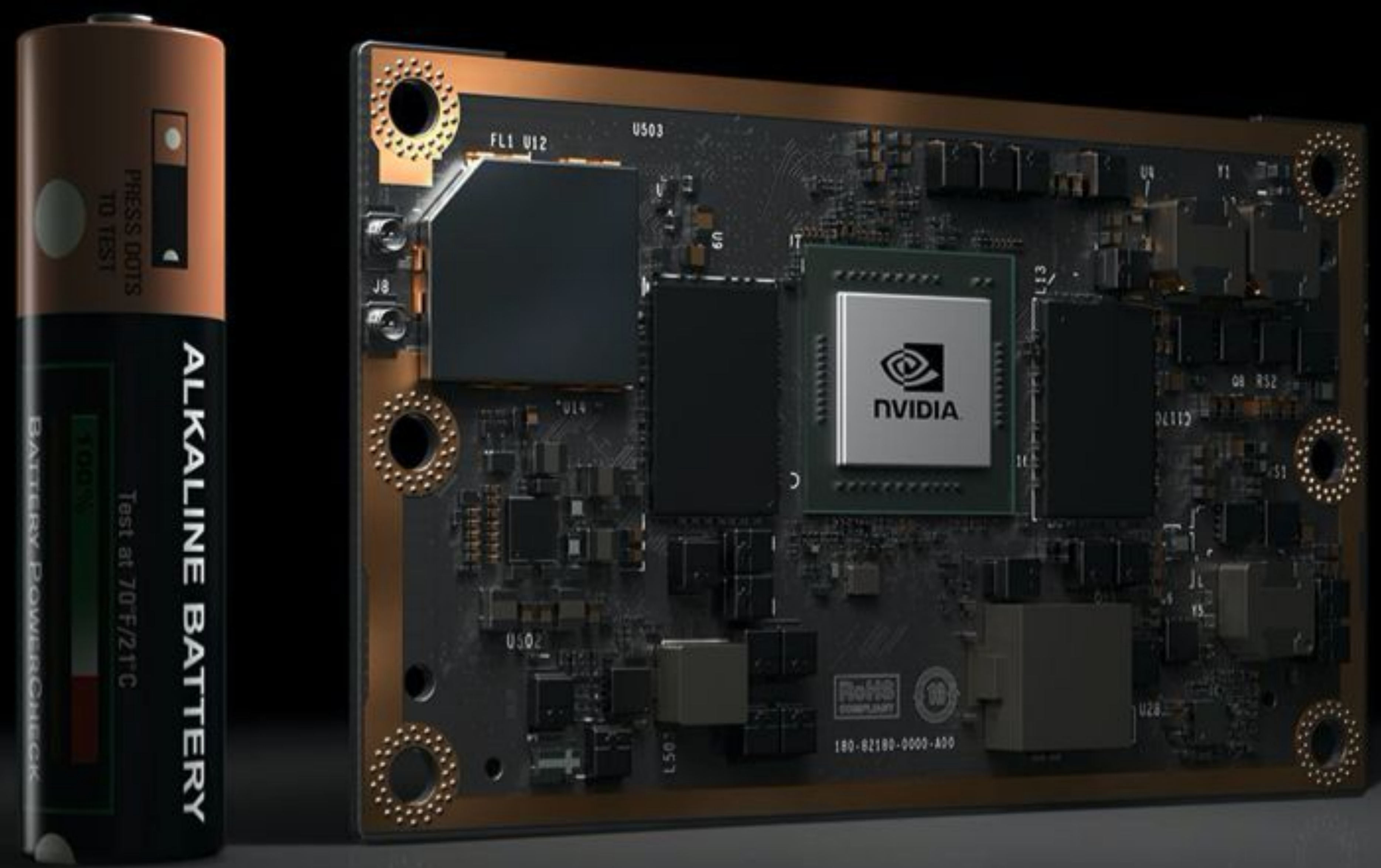
RT Core
10 Giga Rays/sec
Ray Triangle Intersection
BVH Traversal

JETSON NANO SPECIFICATIONS



GPU	128 Core Maxwell 472 GFLOPs (FP16)
CPU	4 core ARM A57 @ 1.43 GHz
Memory	4 GB 64 bit LPDDR4 25.6 GB/s
Storage	16 GB eMMC
Video Encode	4K @ 30 | 4x 1080p @ 30 | 8x 720p @ 30 (H.264/H.265)
Video Decode	4K @ 60 | 2x 4K @ 30 | 8x 1080p @ 30 | 16x 720p @ 30 | (H.264/H.265)
Camera	12 (3x4 or 4x2) MIPI CSI-2 DPHY 1.1 lanes (1.5 Gbps)
Display	HDMI 2.0 or DP1.2 | eDP 1.4 | DSI (1 x2) 2 simultaneous
UPHY	1 x1/2/4 PCIE 1 USB 3.0
SDIO/SPI/SysIOs/GPIOs/I2C	1x SDIO / 2x SPI / 5x SysIO / 13x GPIOs / 6x I2C

NVIDIA JETSON TX2



EMBEDDED AI SUPERCOMPUTER

Advanced AI at the edge

JetPack SDK

< 7.5 watts full module

Up to 2X performance or 2X energy efficiency

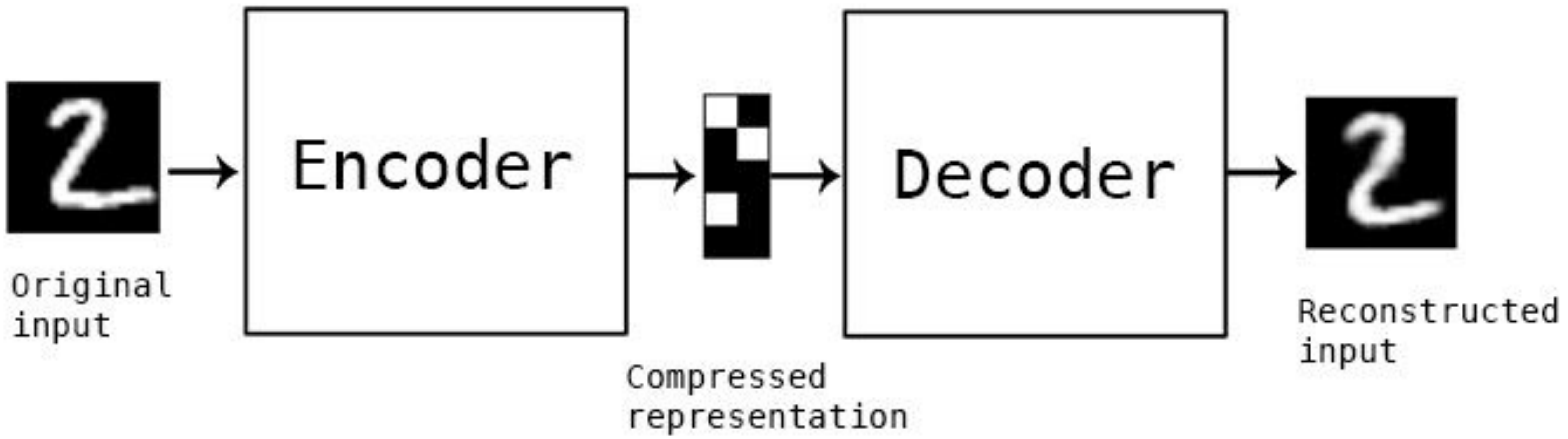
Max-Q operating mode (< 7.5 watts) delivers up to 2x energy efficiency vs. Jetson TX1 maximum performance
Max- P operating mode (< 15 watts) delivers up to 2x performance vs. Jetson TX1 maximum performance

What’s after CNNs?

(an overview)

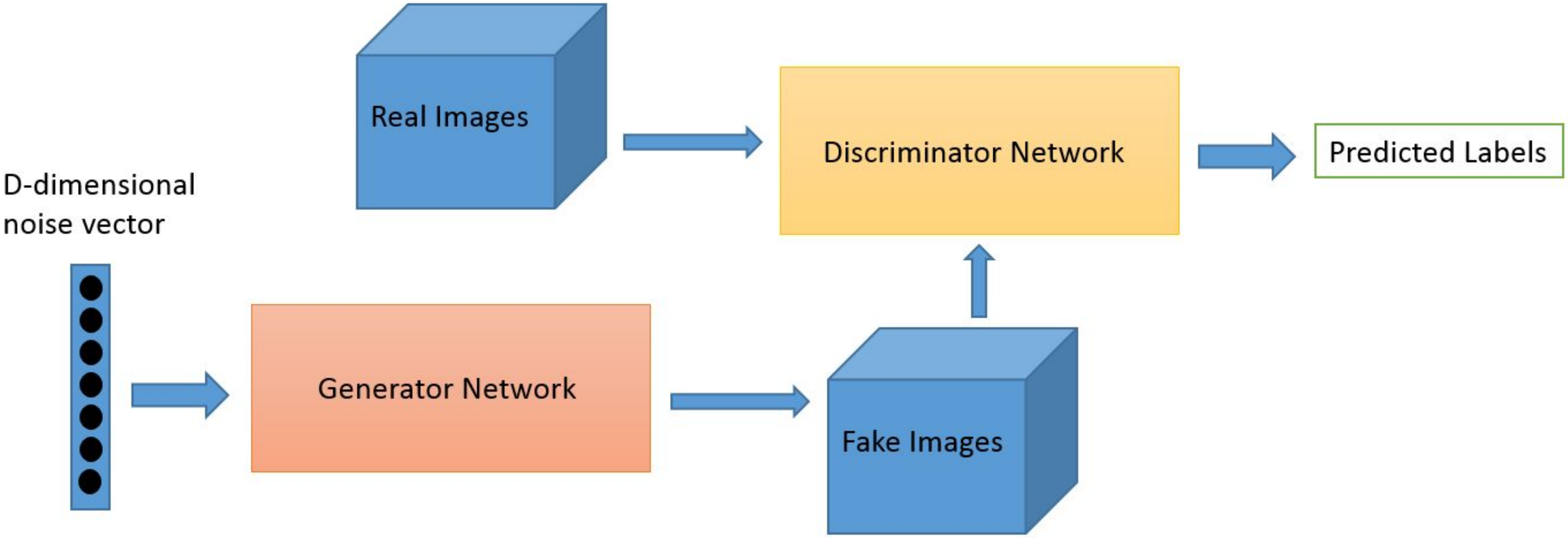
Autoencoders

Used mainly for data compression, a network is trained to represent an image with less information than the original and reconstruct back the input minimizing the loss.



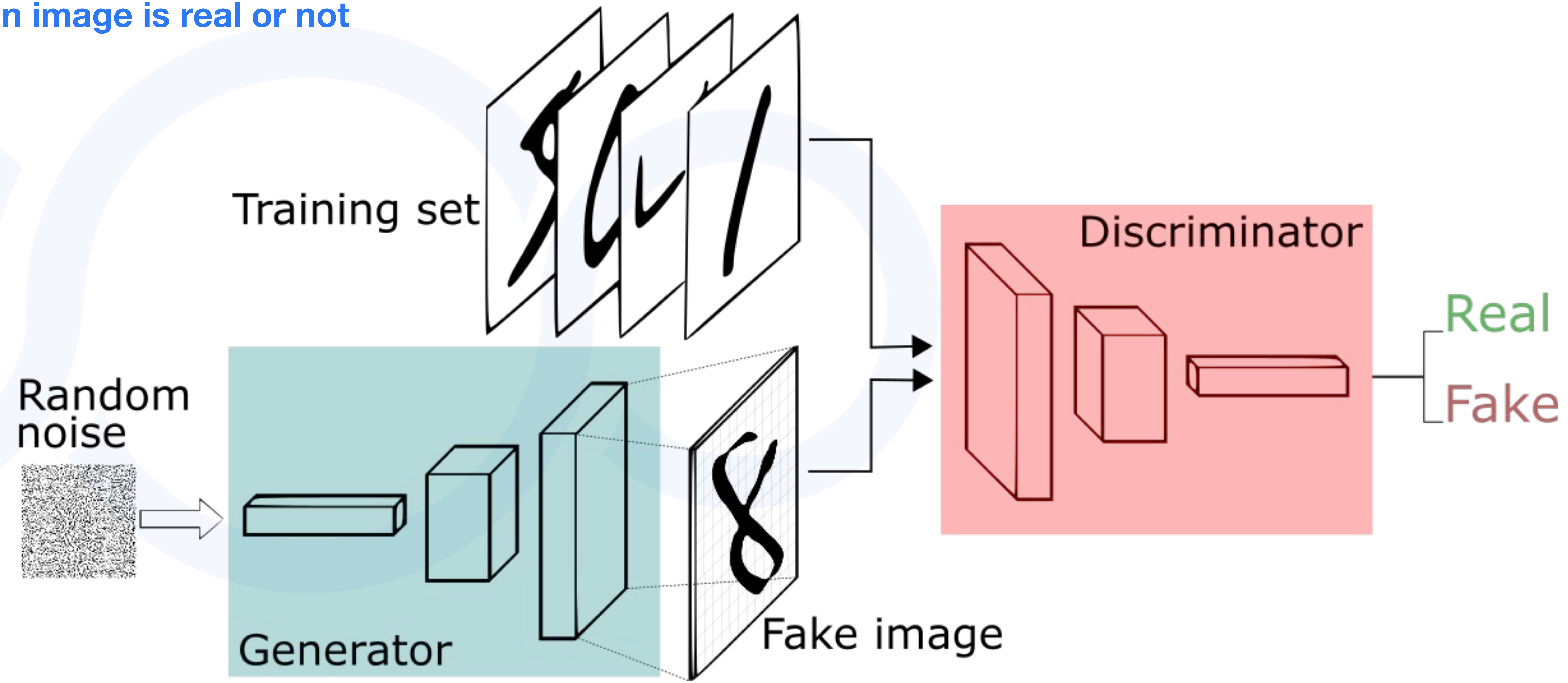
Generative Adversal Networks (GANs)

Generative networks train both a generator network to fake data from noise and detect whether an image is real or not



Generative Adversal Networks (GANs)

Generative networks train both a generator network to fake data from noise and detect whether an image is real or not

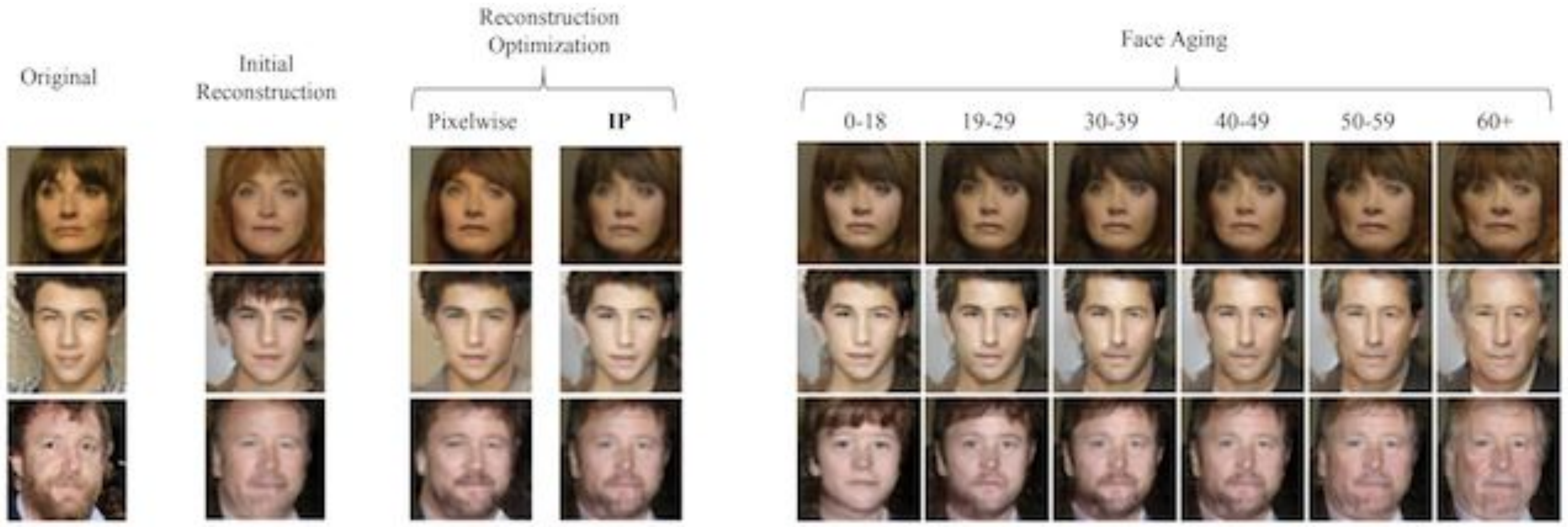


Neosperience

GAN to generate artificial images

GAN can be used to simulate face aging of people in a natural and consistent way.

https://ieeexplore.ieee.org/document/8296650



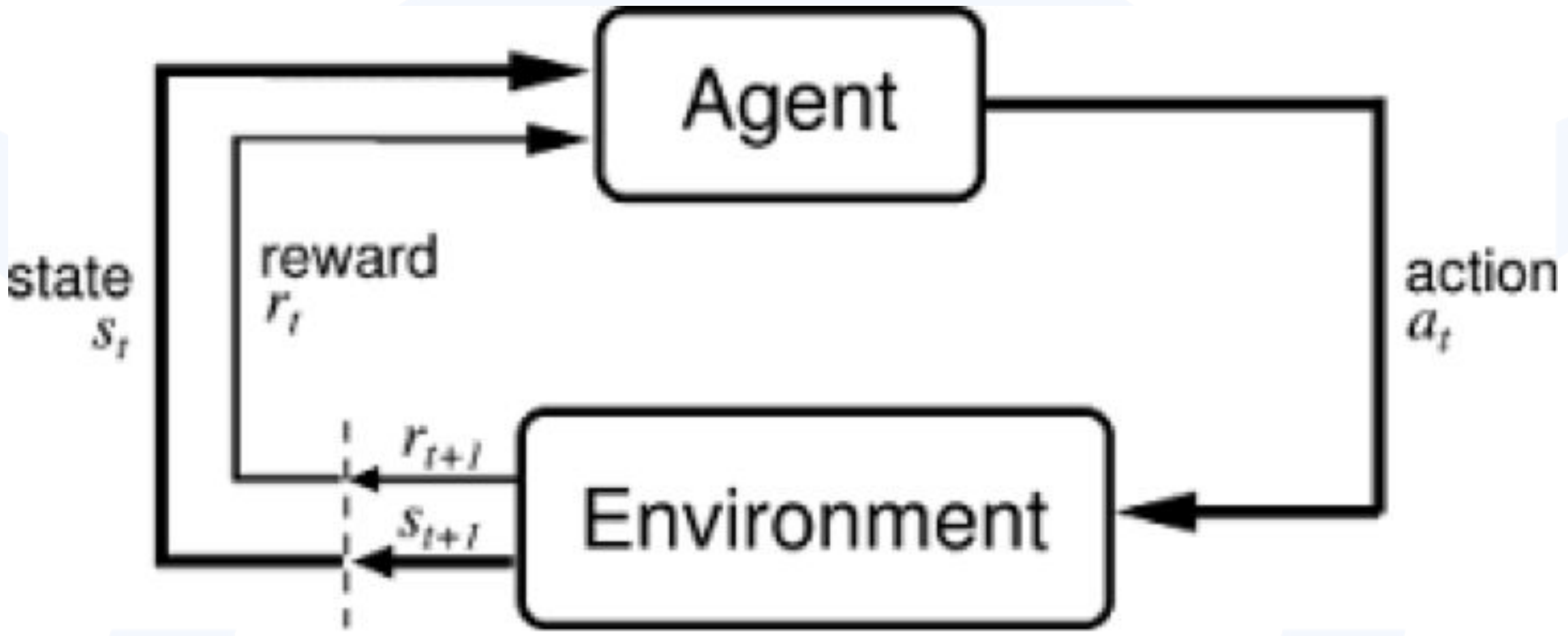
GANs to generate photographs of clothing as may be seen in a catalog or online store, based on photographs of models wearing the clothing.

https://arxiv.org/abs/1603.07442

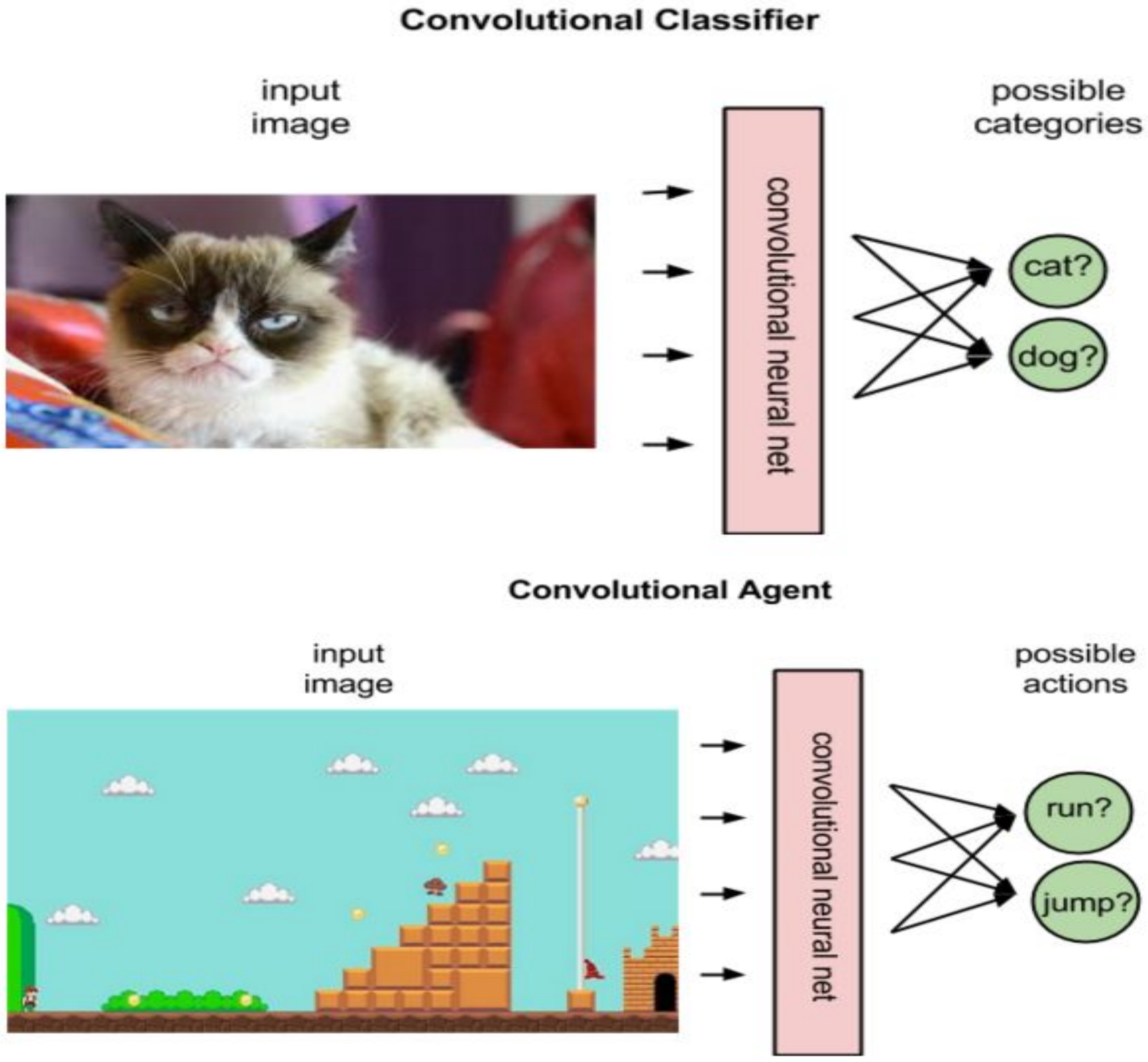
Neosperience

Reinforcement Learning

Used to train models in autonomous feedback-guided loops. It is used to implement variations of autonomous driving agents.



Reinforcement Learning has a wide range of applications from classification with a small dataset, to playing video games, firewall/system parameters tuning, personalizing reccomendations, automatic bidding.



Neosperience

Amazon DeepRacer

Deploy a Deep Learning model on a Deep Racer, then win a race in a global competition

Developers of all skill levels can get hands on with machine learning through a cloud based 3D racing simulator, fully autonomous 1/18th scale race car driven by reinforcement learning, and global racing league.

AWS DeepRacer is an autonomous 1/18th scale race car designed to test RL models by racing on a physical track. Using cameras to view the track and a reinforcement model to control throttle and steering, the car shows how a model trained in a simulated environment can be transferred to the real-world.

We're hiring!

Neosperience

github.com/aletheia

https://it.linkedin.com/in/lucabianchipavia

@bianchiluca

https://speakerdeck.com/aletheia

Thank you.

http://bit.ly/unipv-deep-cv

WE ARE HIRING

Neosperience

www.neosperience.com | blog.neosperience.com | info@neosperience.com