Intro to AI and applications

Basic theoretical concepts and practical examples

TALK

Who am I?

Luca Bianchi, PhD

CTO @ Neosperience

AWS Hero, passionate about **serverless** and **machine learning**

github.com/aletheia

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

@bianchiluca

https://speakerdeck.com/aletheia

www.ai4devs.io

the why, how and what of artificial intelligence

the **why**, how and what of artificial intelligence artificial intelligence

Why is it relevant? an example

AI can predict therapy outcome or help MD in diagnosis

An AI-Driven Genomics Company Is Turning to Drugs

Deep Genomics aims to develop drugs by using deep learning to find patterns in genomic and medical data.

by Will Knight May 3, 2017

T he next blockbust
from machine-lea
spreading from A

Deep Genomics, a Canadian
to trace potential genetic cau
that it's getting into drug dev
companies betting that their
new drugs by finding subtle
data.

Comments 243 Shares

Children's Hospital LOS ANGELES
We Treat Kids Better

Share Your Science: Leveraging Deep Learning for Personalized Drug Treatment Recommendations

April 15, 2016

David Ledbetter, data scientist at the Children's Hospital Los Angeles, shares how his team is using TITAN X GPUs and deep learning to help provide better recommendations of drug treatments for children in their pediatric intensive care unit.

To train their models, 13,000 patient snapshots were created from ten years of electronic health records at the hospital to understand the interactions between a patient's vital state, heart rate, blood pressure and the treatments they were given. By understanding the most important relationships in the data, they are then able to generate the probability of survival predictions for the patients moving forward as well as physiology predictions in order to simulate augmented treatments.

Prediction of premature all-cause mortality: A prospective general population cohort study comparing machine-learning and standard epidemiological approaches

Stephen F. Weng, Luis Vaz, Nadeem Qureshi, Joe Kai

Published: March 27, 2019 • https://doi.org/10.1371/journal.pone.0214365

Article | Authors | Metrics | Comments | Media Coverage

Abstract
Introduction
Methods
Results
Discussion
Supporting information
Acknowledgments
References

Reader Comments (0)
Media Coverage (2)
Figures

Abstract

Background

Prognostic modelling using standard methods is well-established, particularly for predicting risk of single diseases. Machine-learning may offer potential to explore outcomes of even greater complexity, such as premature death. This study aimed to develop novel prediction algorithms using machine-learning, in addition to standard survival modelling, to predict premature all-cause mortality.

Methods

A prospective population cohort of 502,628 participants aged 40–69 years were recruited to the UK Biobank from 2006–2010 and followed-up until 2016. Participants were assessed on a range of demographic, biometric, clinical and lifestyle factors. Mortality data by ICD-10 were obtained from linkage to Office of National Statistics. Models were developed using deep learning, random forest and Cox regression. Calibration was assessed by comparing observed to predicted risks; and discrimination by area under the 'receiver operating curve' (AUC).

Findings

14,418 deaths (2.9%) occurred over a total follow-up time of 3,508,454 person-years. A simple age and gender Cox model was the least predictive (AUC 0.689, 95% CI 0.681–0.699). A multivariate Cox regression model significantly improved discrimination by 6.2% (AUC 0.751, 95% CI 0.748–0.767). The application of machine-learning algorithms further improved discrimination by 3.2% using random forest (AUC 0.783, 95% CI 0.776–0.791) and 3.9% using deep learning (AUC 0.790, 95% CI 0.783–0.797). These ML algorithms improved discrimination by 9.4% and 10.1% respectively from a simple age and gender Cox regression model. Random forest and deep learning achieved similar levels of discrimination with no significant difference. Machine-learning algorithms were well-calibrated, while Cox regression models consistently over-predicted risk.

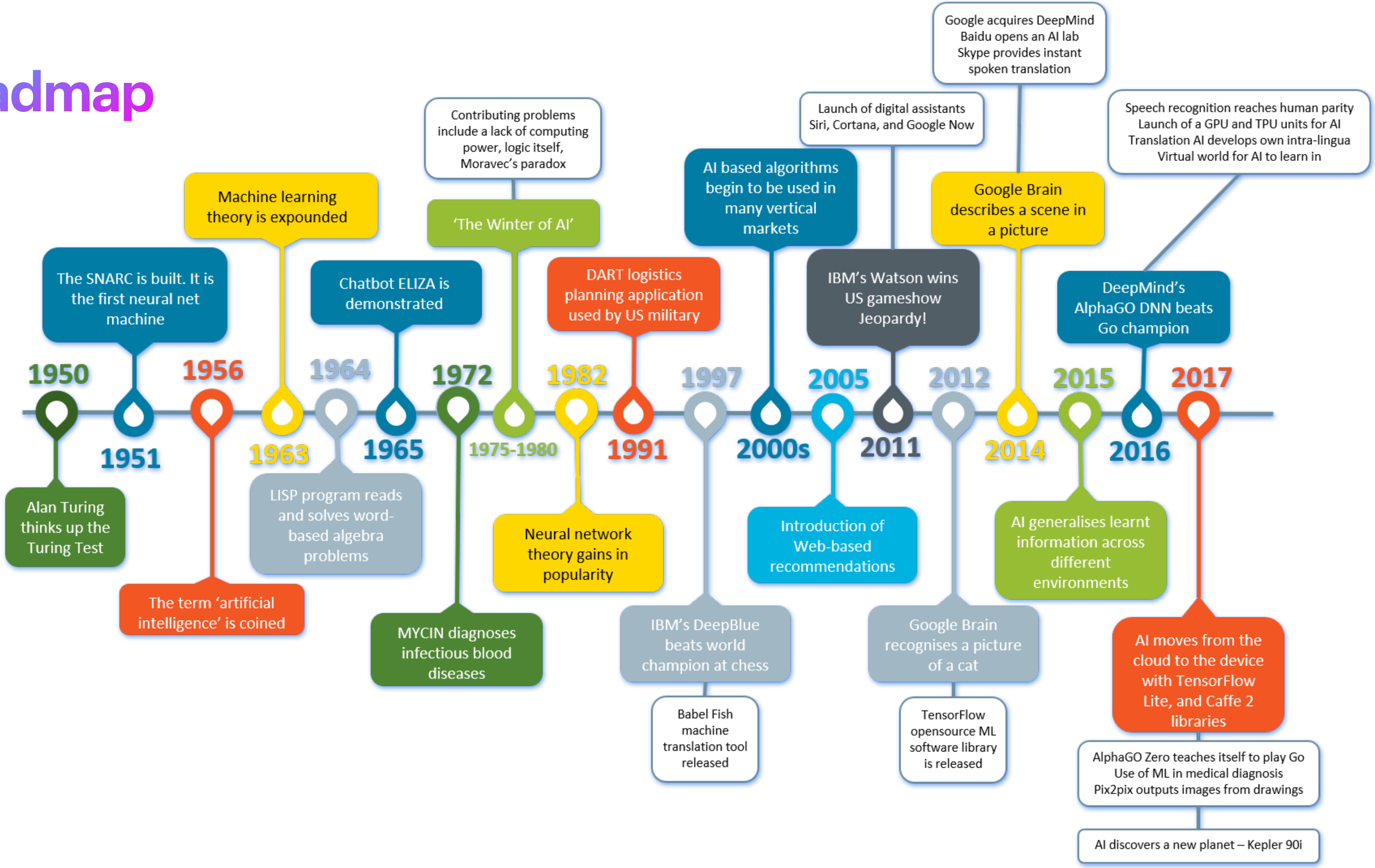
VENTURE CAPITAL DISPATCH

Enlitic Raises $10 Million for A.I. in Medical Imaging

By Timothy Hay
Oct 28, 2015 3:19 pm ET

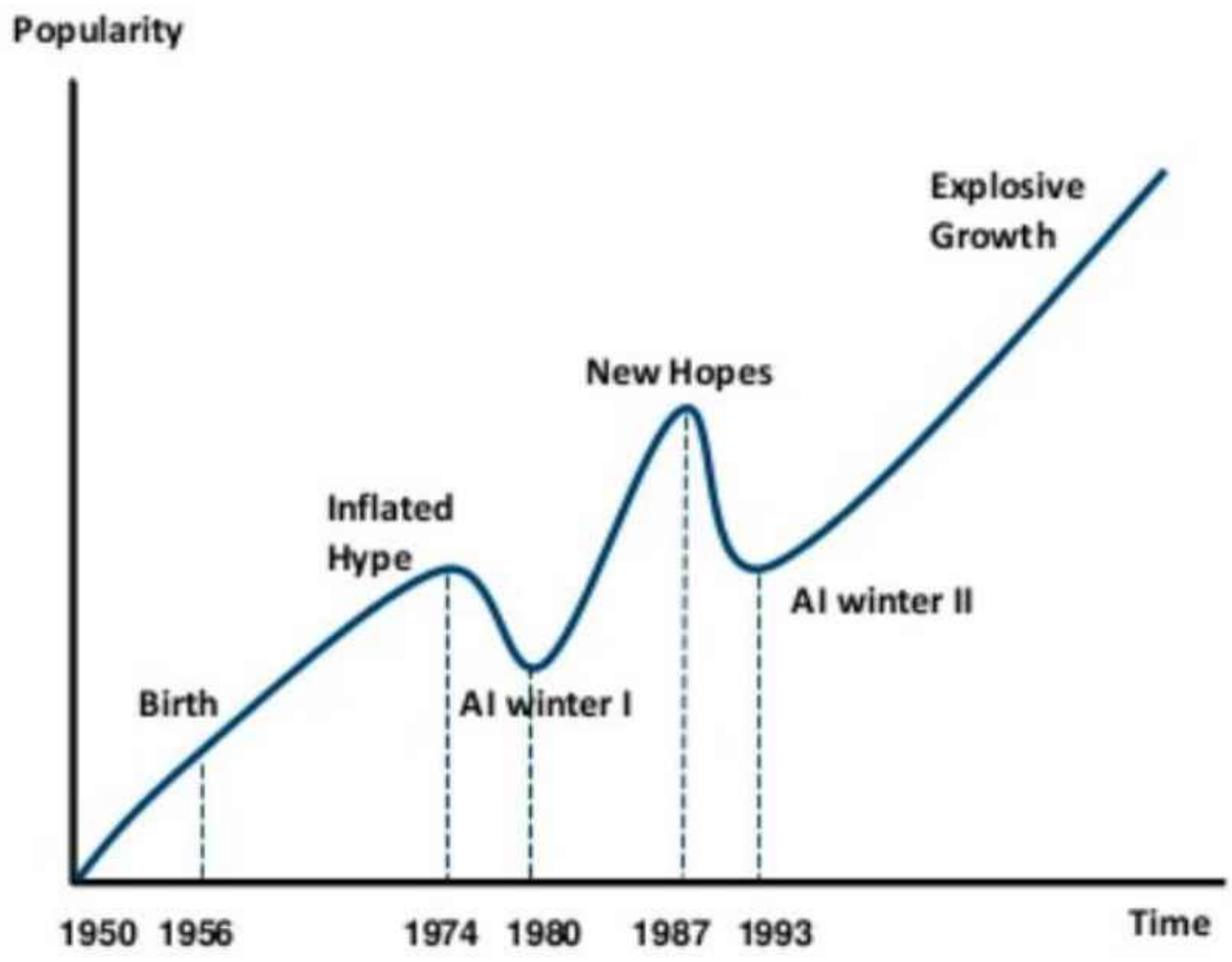
The company, which applies machine learning to X-rays, MRIs and other medical images to offer a new diagnostic product for radiologists, has announced new funding and a partnership that will see Enlitic's technology used in more than 80 imaging centers.

AI roadmap



1950
Alan Turing thinks up the Turing Test
1951
The SNARC is built. It is the first neural net machine
1956
The term 'artificial intelligence' is coined
1963
Machine learning theory is expounded
1964
LISP program reads and solves word-based algebra problems
1965
Chatbot ELIZA is demonstrated
1972
MYCIN diagnoses infectious blood diseases
1975-1980
'The Winter of AI'
Contributing problems include a lack of computing power, logic itself, Moravec's paradox
1982
Neural network theory gains in popularity
1991
DART logistics planning application used by US military
1997
IBM's DeepBlue beats world champion at chess
Babel Fish machine translation tool released
2000s
AI based algorithms begin to be used in many vertical markets
2005
Introduction of Web-based recommendations
2011
IBM's Watson wins US gameshow Jeopardy!
Launch of digital assistants Siri, Cortana, and Google Now
2012
Google Brain recognises a picture of a cat
TensorFlow opensource ML software library is released
2014
Google Brain describes a scene in a picture
Google acquires DeepMind
Baidu opens an AI lab
Skype provides instant spoken translation
2015
AI generalises learnt information across different environments
2016
DeepMind's AlphaGO DNN beats Go champion
Speech recognition reaches human parity
Launch of a GPU and TPU units for AI
Translation AI develops own intra-lingua
Virtual world for AI to learn in
2017
AI moves from the cloud to the device with TensorFlow Lite, and Caffe 2 libraries
AlphaGO Zero teaches itself to play Go
Use of ML in medical diagnosis
Pix2pix outputs images from drawings
AI discovers a new planet – Kepler 90i

Beware of “winters”

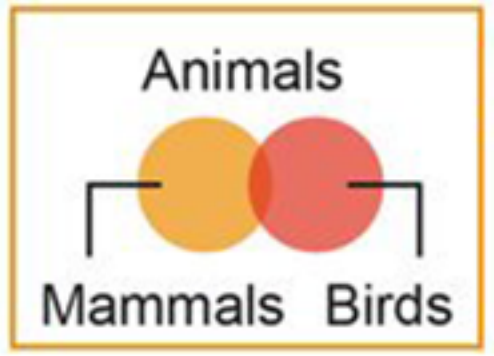


Tribes of Artificial Intelligence

for decades individual “tribes” of artificial intelligence researchers have vied one another for dominance. Is the time now for tribes to collaborate? They may be forced to, as collaboration and algorithm blending are the only ways to reach true AGI.

What are the five Tribes?

Symbolists



Use symbols, rules, and logic to represent knowledge and draw logical inference

Favored algorithm

Rules and decision trees

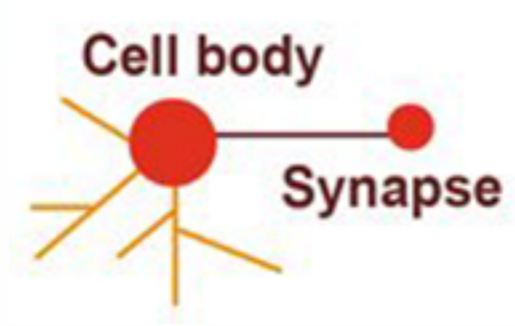
Bayesians

Assess the likelihood of occurrence for probabilistic inference

Favored algorithm

Naive Bayes or Markov

Connectionists



Recognise and generalise patterns dynamically with matrices of probabilistic weighted neurons

Favored algorithm

Neural Networks

Evolutionaries

Generate variations and then assess the fitness of each for a given purpose

Favored algorithm

Genetic Programs

Analogizers

Optimize a function in light of constraints (“going as high as you can while staying on the road”)

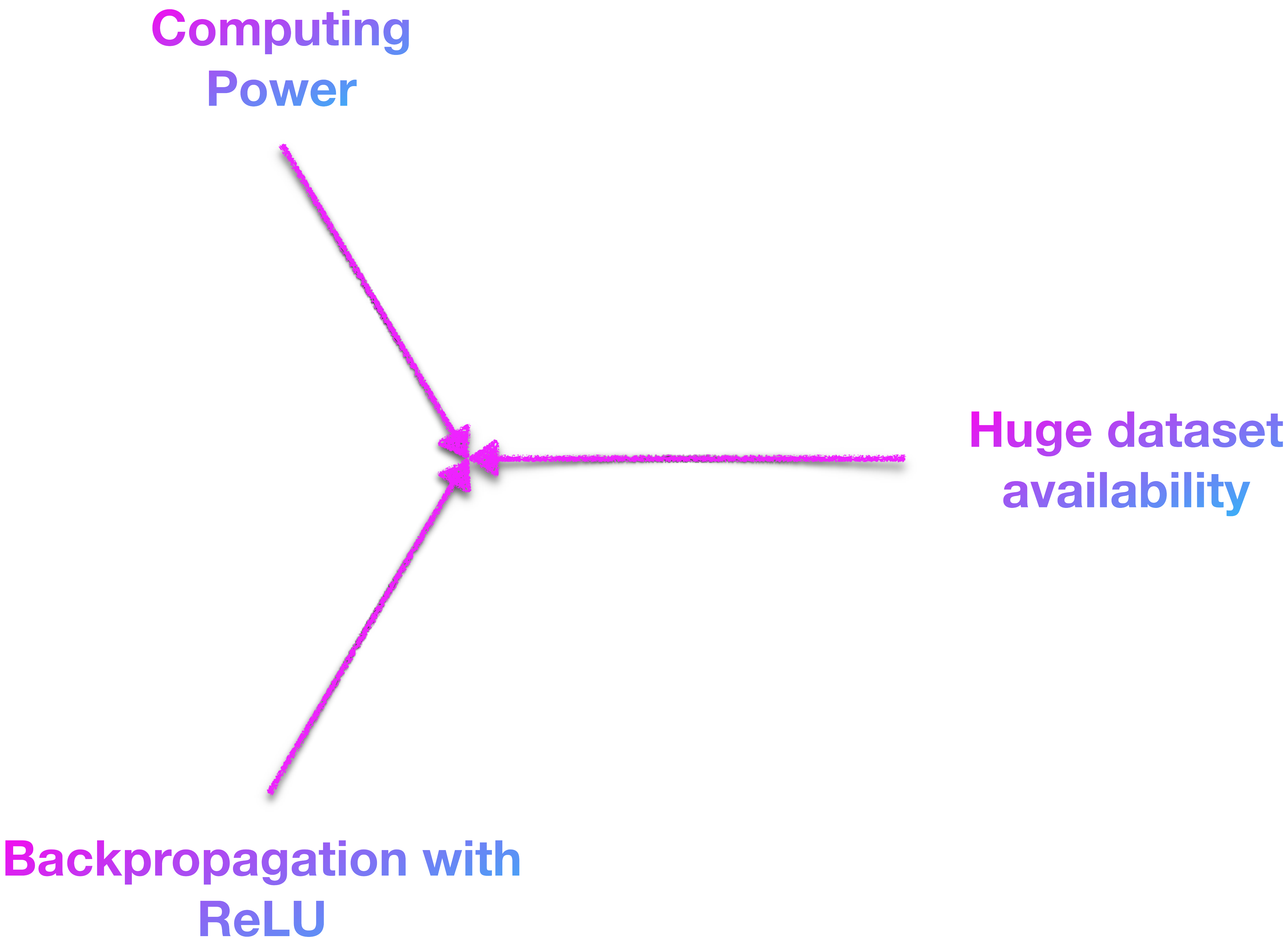
Favored algorithm

Support vectors

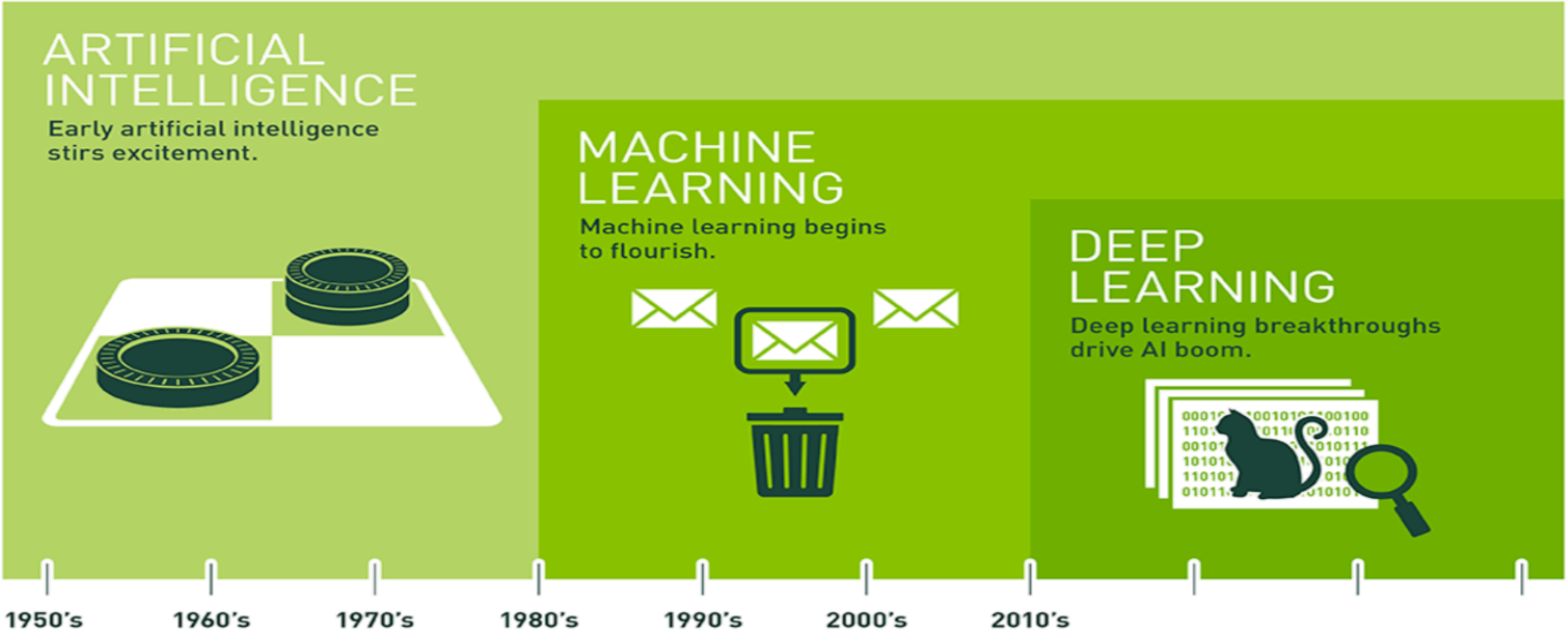
Neosperience

Why now?

Nowadays we're approaching a **nexus** of many forces



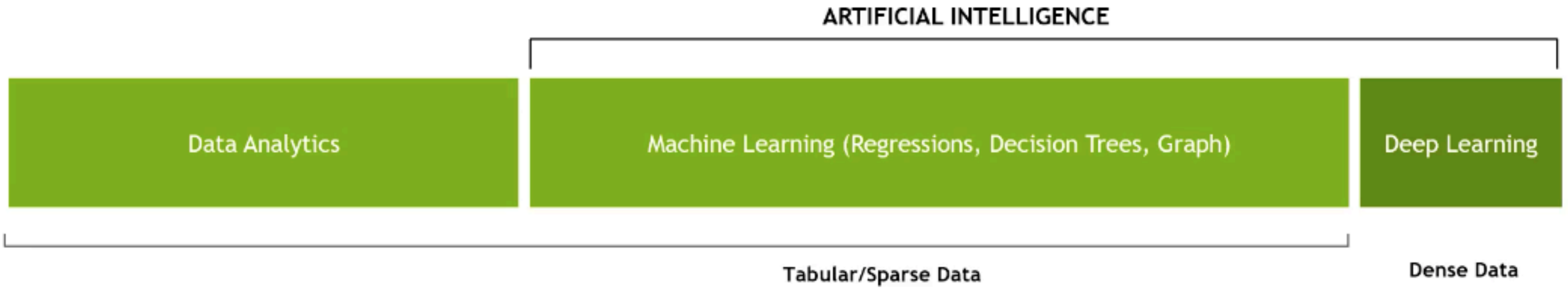
Deep Learning Roadmap



Accelerated Data Science

Beyond Deep Learning

Structured data doesn't need deep learning, but it could be "just" a machine learning or a big data problem



2.2 exabytes (2.2B GB) of data created daily - McKinsey

$274B annual revenue by 2022 for big data and business analytics - IDC

What problem are you solving?

Unstructured data type, deep learning task, and business domain

INPUTS	BUSINESS QUESTIONS	AI / DL TASK	EXAMPLE OUTPUTS		
			HEALTHCARE	RETAIL	MANUFACTURING
Text Data Images Audio Video	Is "it" **present** or not?	Detection	Cancer Detection	Targeted Ads	Defect Detection
	What **type** of thing is "it"?	Classification	Transcription / Image Classification	Basket Analysis	Material Sorting
	To what **extent** is "it" present?	Segmentation	Tumor Size & Shape Analysis	360° Customer Views	Autonomous Navigation
	What is the likely **outcome**?	Prediction	Survivability Prediction	Sentiment & Behavior Recognition	Predictive Maintenance
	What will likely satisfy the objective?	Recommendations	Therapy Recommendation	Recommendation Engine	Supply Chain Optimization

machine learning

the why, **how** and what of artificial intelligence

Machine Learning

An operational definition

“A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E”

The importance of Experience

- Machine Learning (ML) algorithms have data as input, ‘cause data represents the Experience.
  This is a focal point of Machine Learning: large amount of data is needed to achieve good performances.
- The Machine Learning equivalent of program in ML world is called ML model and improves over time as soon as more data is provided, with a process called training.
- Data must be prepared (or filtered) to be suitable for training process. Generally input data must be collapsed into a n-dimensional array with every item representing a sample.
- ML performances are measured in probabilistic terms, with metrics called accuracy or precision.

Machine Learning — Taxonomy

Types of Machine Learning

Machine learning tasks are typically classified into three broad categories, depending on the nature of the learning "signal" or "feedback" available to a learning system.

Input-based taxonomy

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

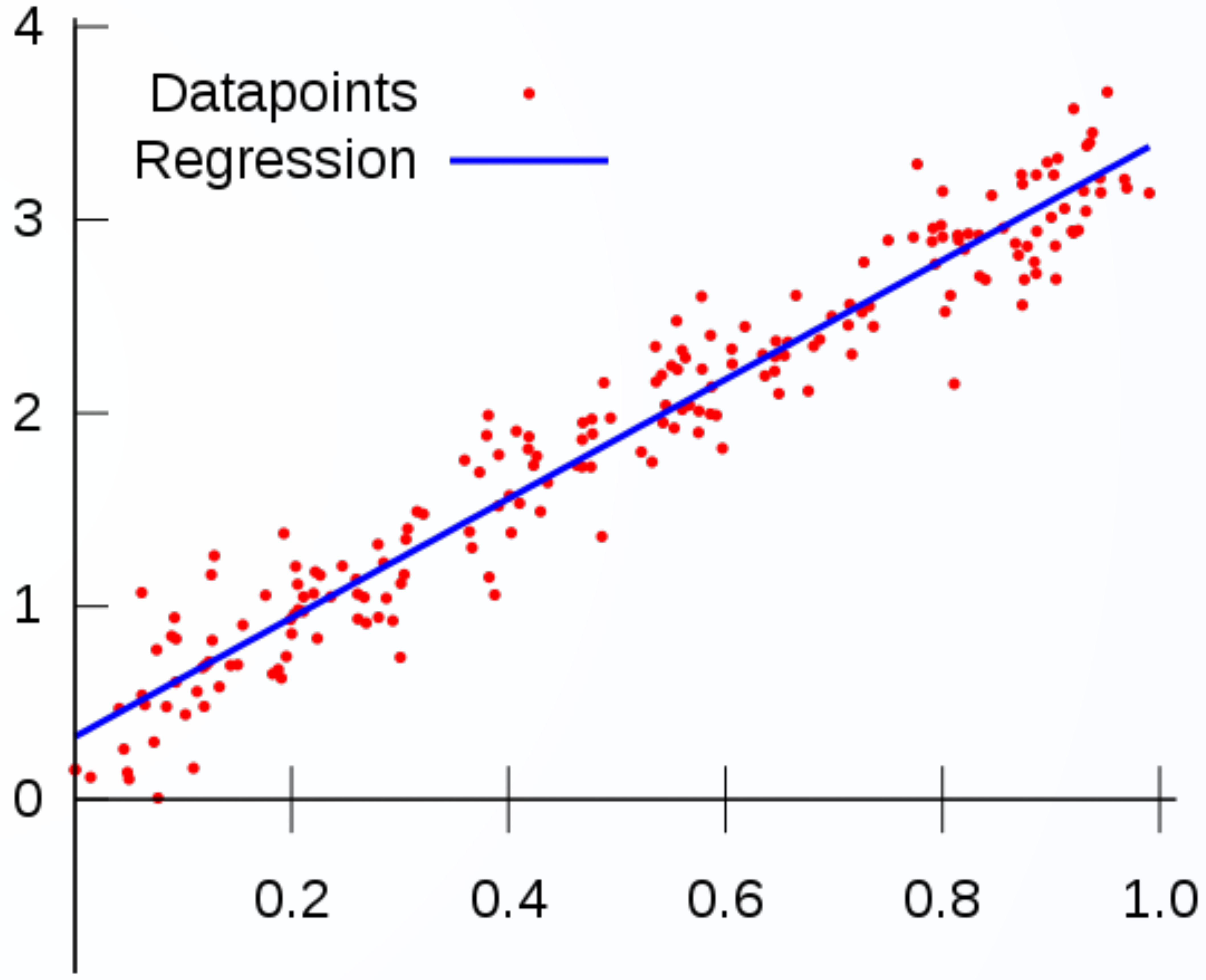
Output-based taxonomy

- Regression
- Classification
- Clustering
- Density estimation
- Dimensionality reduction

Regression

Regression analysis helps one understand how the typical value of the dependent variable (or 'criterion variable') changes when any one of the independent variables is varied, while the other independent variables are held fixed.

Is a statistical method of data analysis. The most common algorithm least square method that provides an estimation of regression parameters. When dataset is not trivial estimation is achieved through is gradient descent.



Regression — use cases

Statistical regression is used to make predictions about data, filling the gaps

Regression, even in the most simple form of Linear Regression is a good tool to learn from data and make predictions based on data trend.

Common scenarios

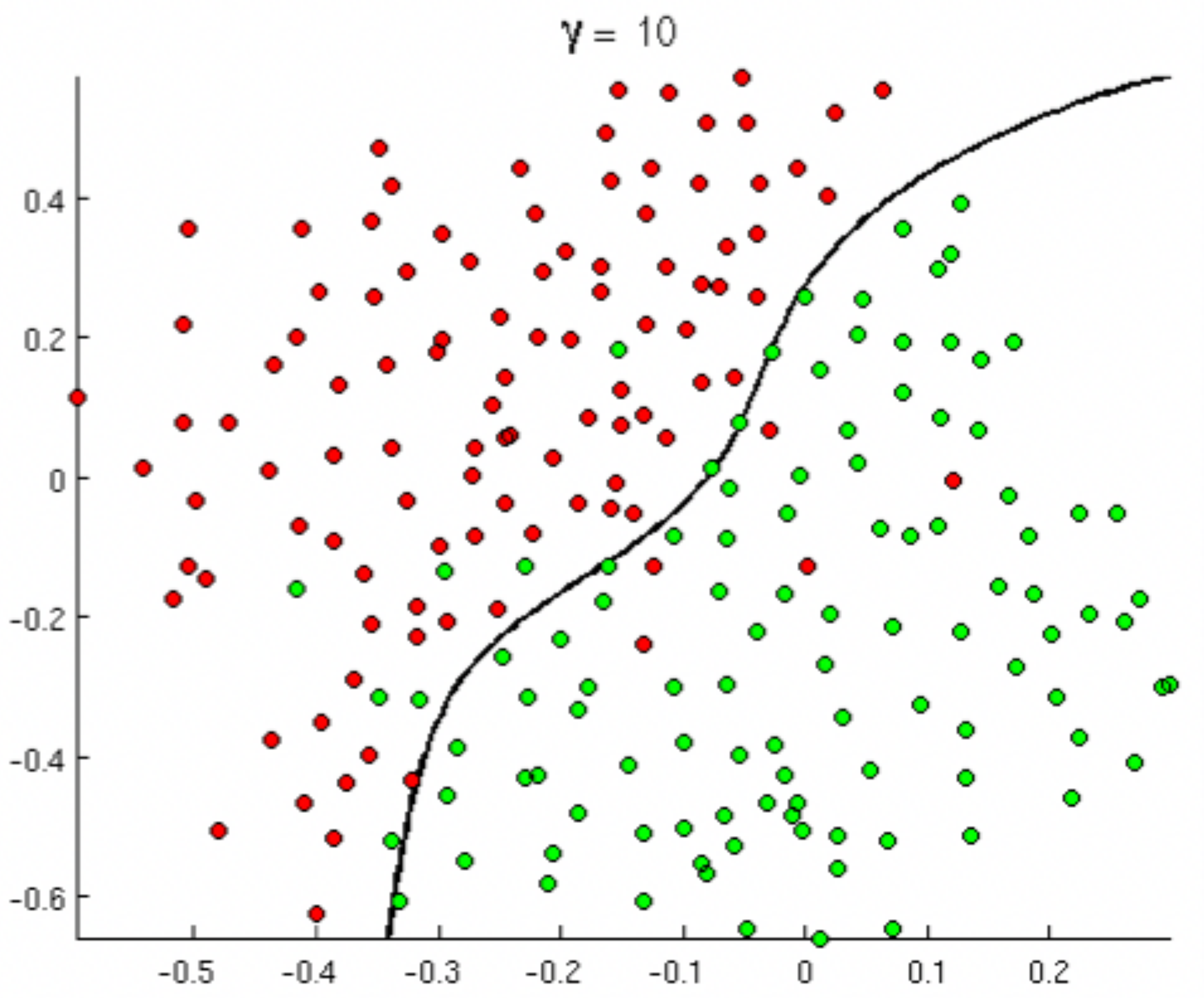
- Stock price value
- Product Price Estimation
- Age estimation
- Customer satisfaction rate defining variables such as response-time, resolution-ration we can forecast satisfaction level or churn
- Customer Conversion rate estimation (based on click data, origin, timestamp, ...)

Classification

Classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known.

Most used algorithms for classification are:

- Logit Regression
- Decision Trees
- Random Forest



Classification — use cases

Classification is used to detect the binary outcome of a variable

Classification is often used to classify people into pre-defined clusters (good-payer/bad-payer, in/out target, etc.)

Common scenarios

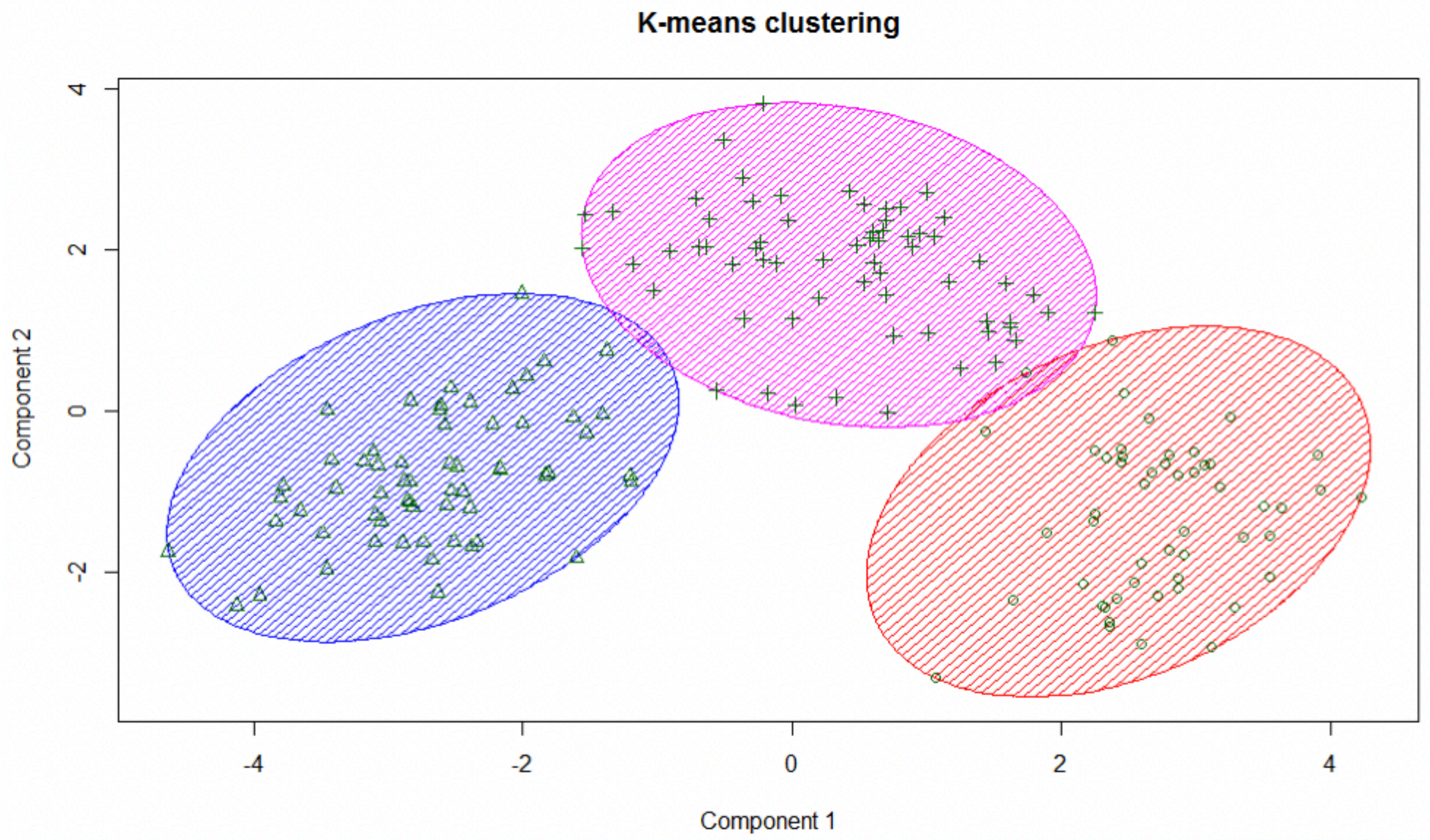
- Credit scoring
- Human Activity Recognition
- Spam/Not Spam classification
- Customer conversion prediction
- Customer churn prediction
- Customer personas classification

Clustering

is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters).

The difference between algorithms is due to the **similarity function** that is used:

- Centroid based clusters
- Density based cluster
- Random Forest



Clustering — use cases

Clustering is used to segment data

Clustering labels each sample with a name representing its belonging cluster. Labelling can be exclusive or multiple.

Clusters are dynamic structures: they adapt to new sample coming into the model as soon as thy label them.

Common scenarios

- Similar interests recognition
- Shape detection
- Similarity analysis
- Customer base segmentation

...and deep learning?

“deep learning is a great phrase,
*it seems so **deep**”*

understanding your problem

Deep Learning

An operational definition

“A class of machine learning techniques that exploit many layers of non-linear information processing for supervised or unsupervised feature extraction and transforma- tion, and for pattern analysis and classification.”

How “deep” is your deep learning?

- Deep Learning (DL) is based on non-linear structures that process information. The “deep” in name comes from the contrast with “traditional” ML algorithms that usually use only one layer. What is a layer?
- A cost-function receiving data as input and outputting its function weights.
- More complex is the data you want to learn from, more layers are usually needed to learn from. The number of layers is called **depth** of the DL algorithm.

Neural Networks (NN)

An operational definition

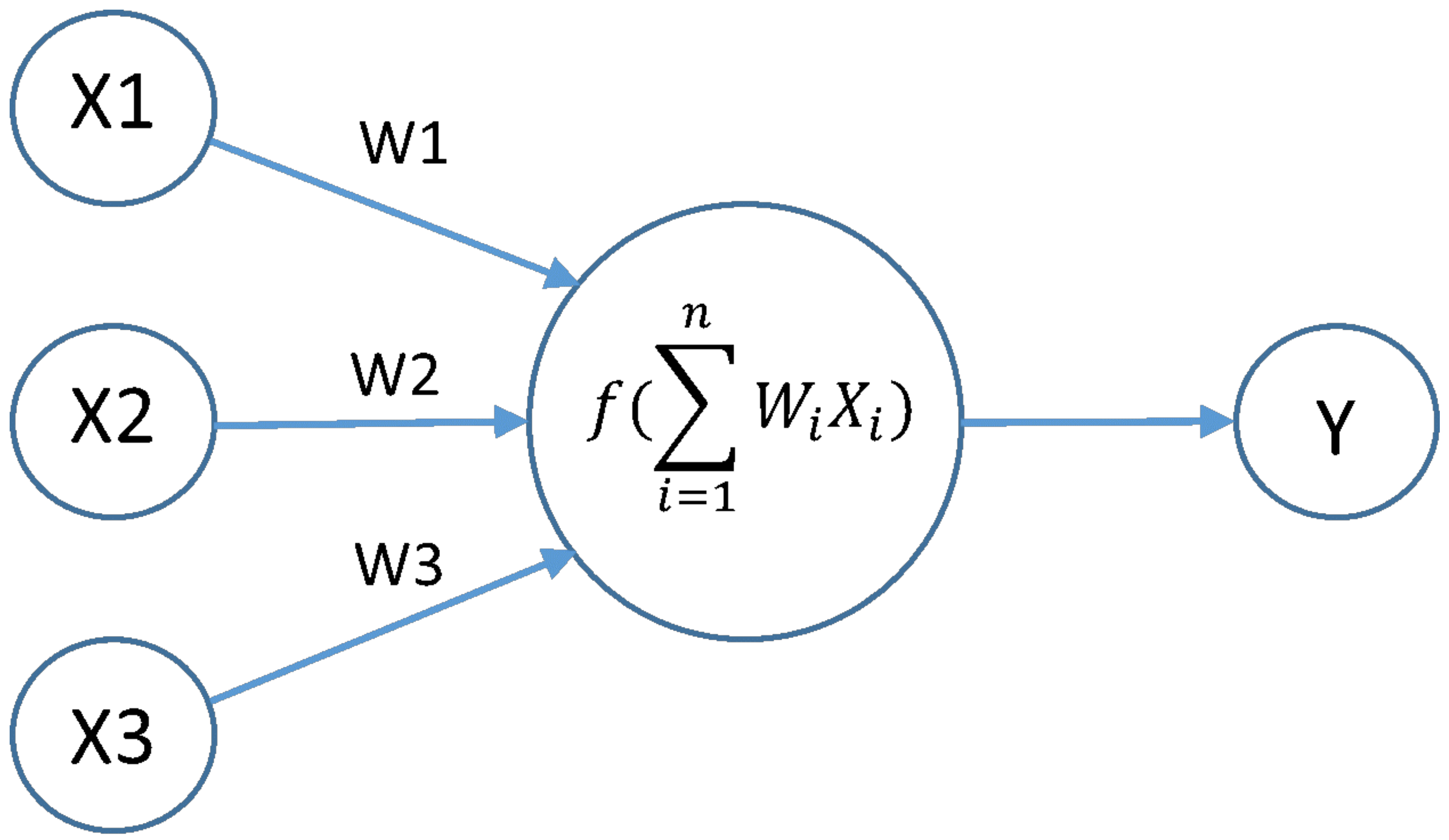
“computing systems inspired by the biological neural networks that constitute animal brains. Such systems learn (progressively improve performance) to do tasks by considering examples, generally without task-specific programming”

A NN is based on a collection of connected units called artificial neurons, (analogous to axons in a biological brain). Each connection (synapse) between neurons can transmit a signal to another neuron. The receiving (postsynaptic) neuron can process the signal(s) and then signal downstream neurons connected to it. Neurons may have state, generally represented by real numbers, typically between 0 and 1. Neurons and synapses may also have a weight that varies as learning proceeds, which can increase or decrease the strength of the signal that it sends downstream. Further, they may have a threshold such that only if the aggregate signal is below (or above) that level is the downstream signal sent.

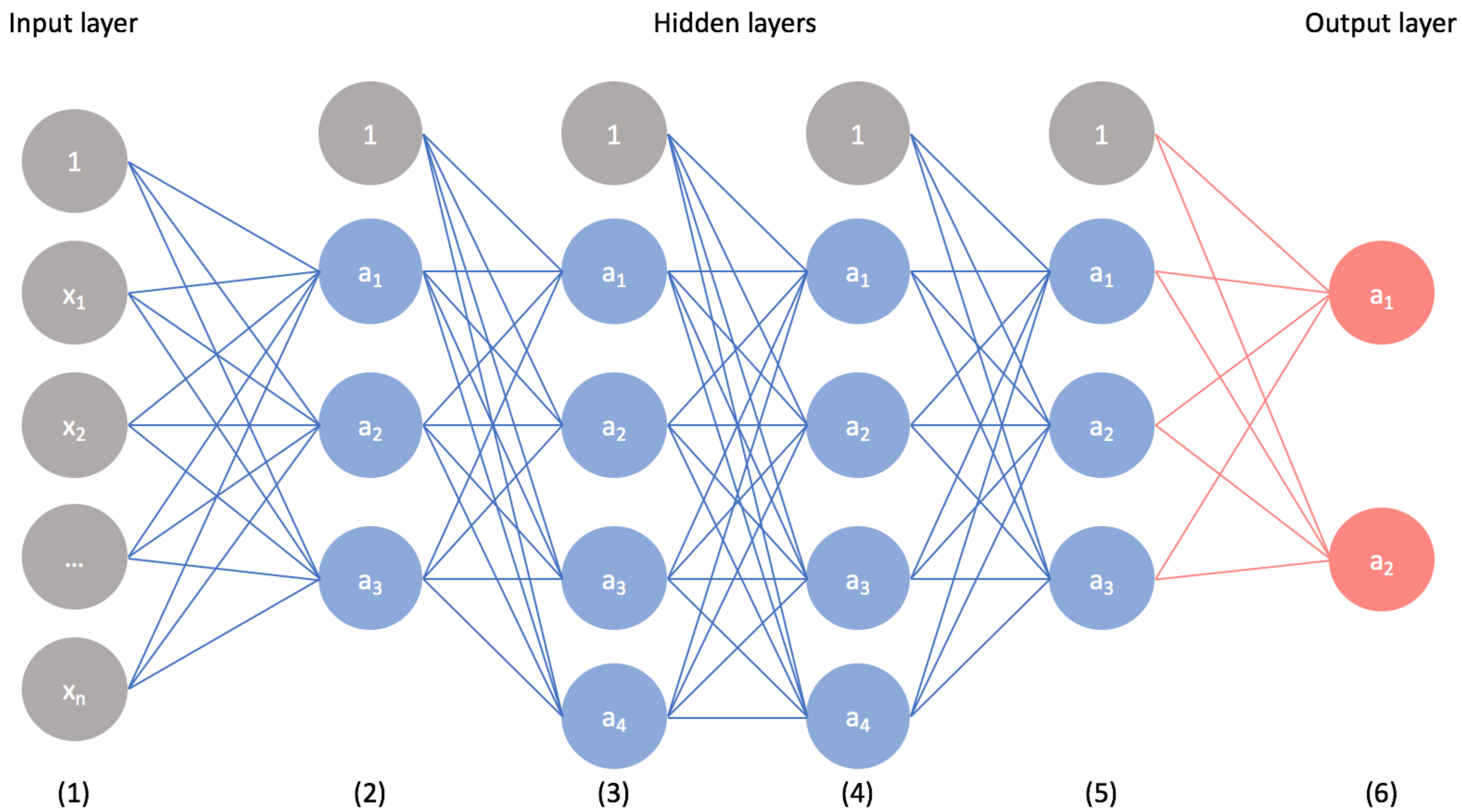
Typically, neurons are organized in layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first (input), to the last (output) layer, possibly after traversing the layers multiple times.

Anatomy of a Neural Network

A Perceptron

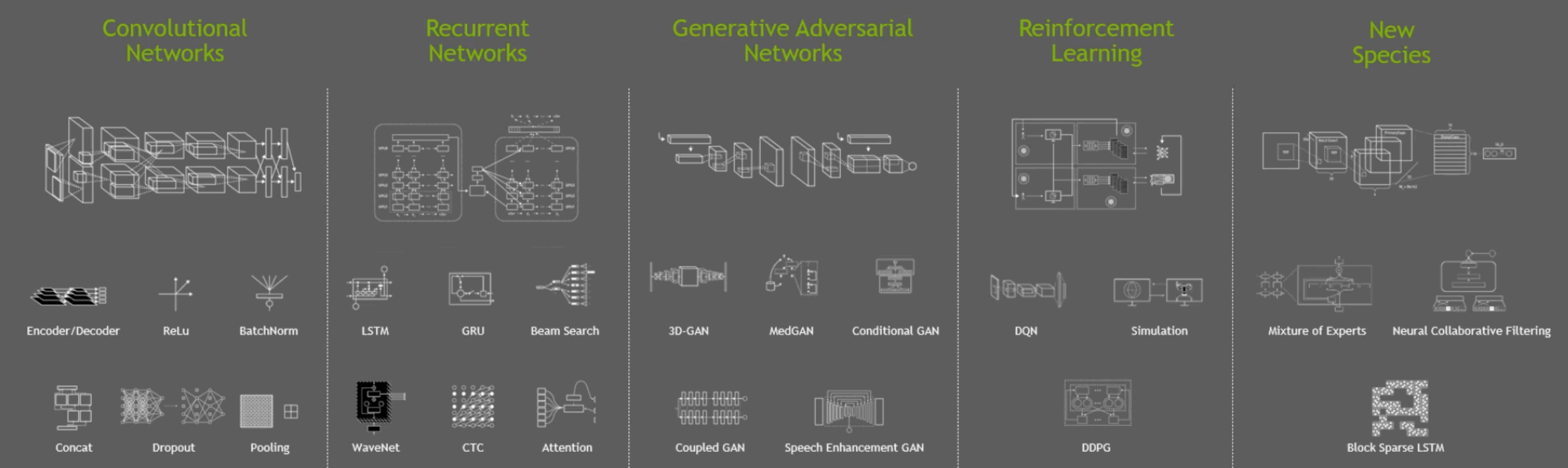


A network of Perceptron



Deep Learning Species

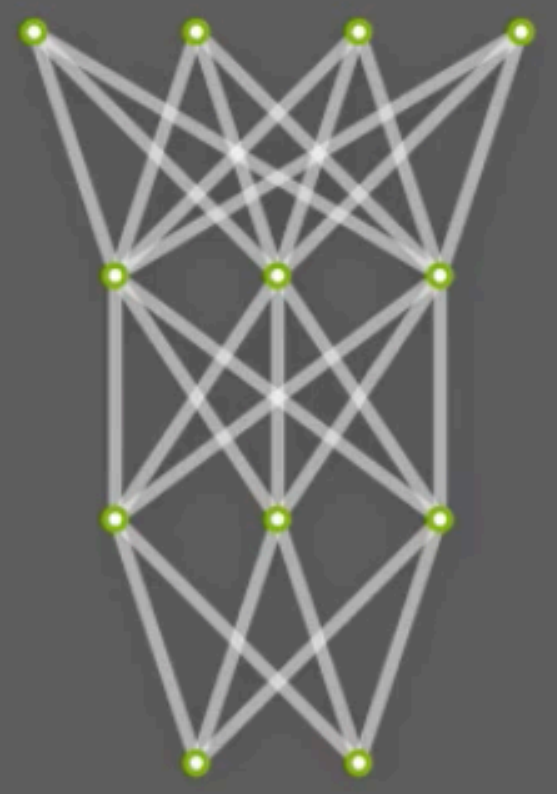
A Cambrian Explosion



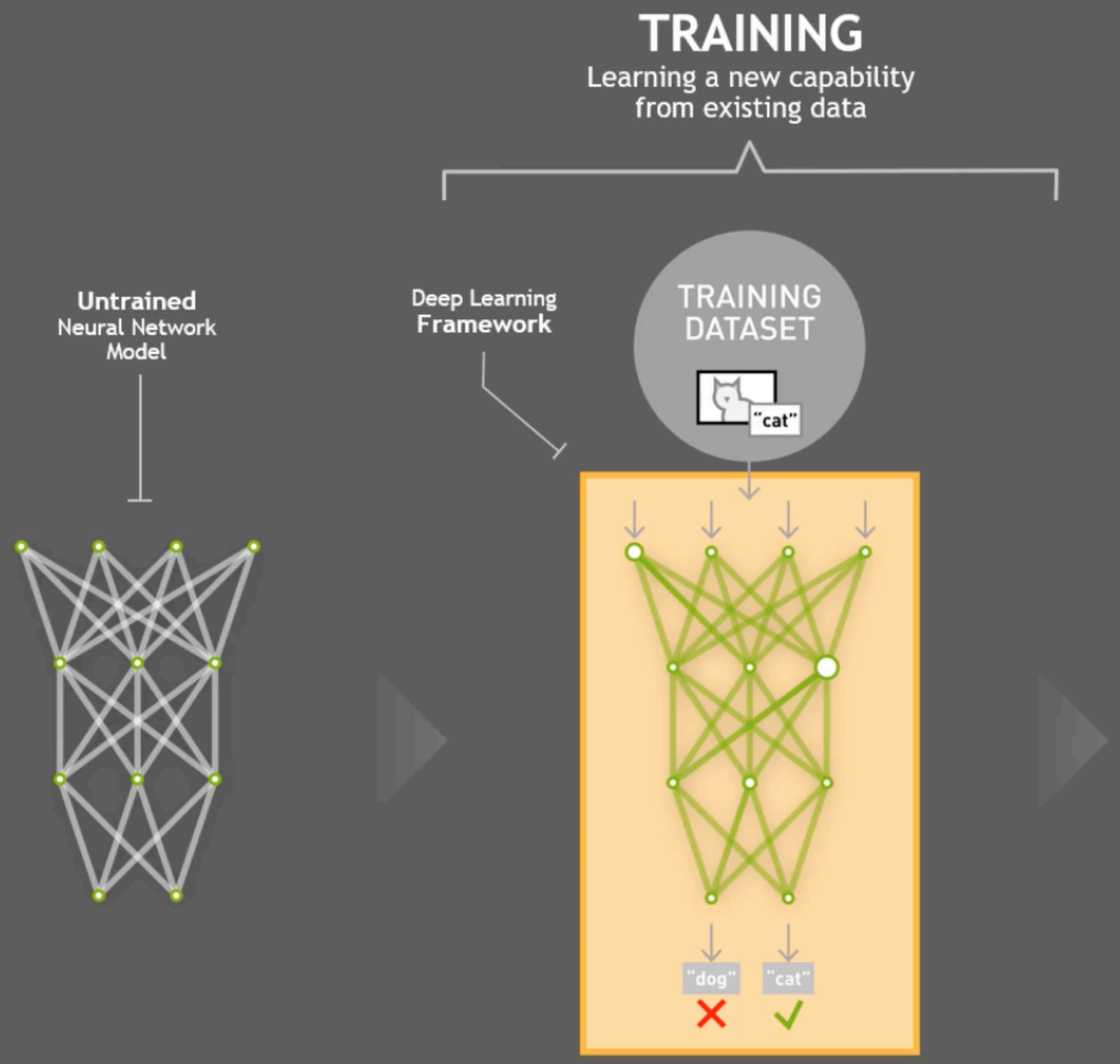
DEEP LEARNING APPLICATION DEVELOPMENT

DEEP LEARNING APPLICATION DEVELOPMENT

Untrained
Neural Network
Model



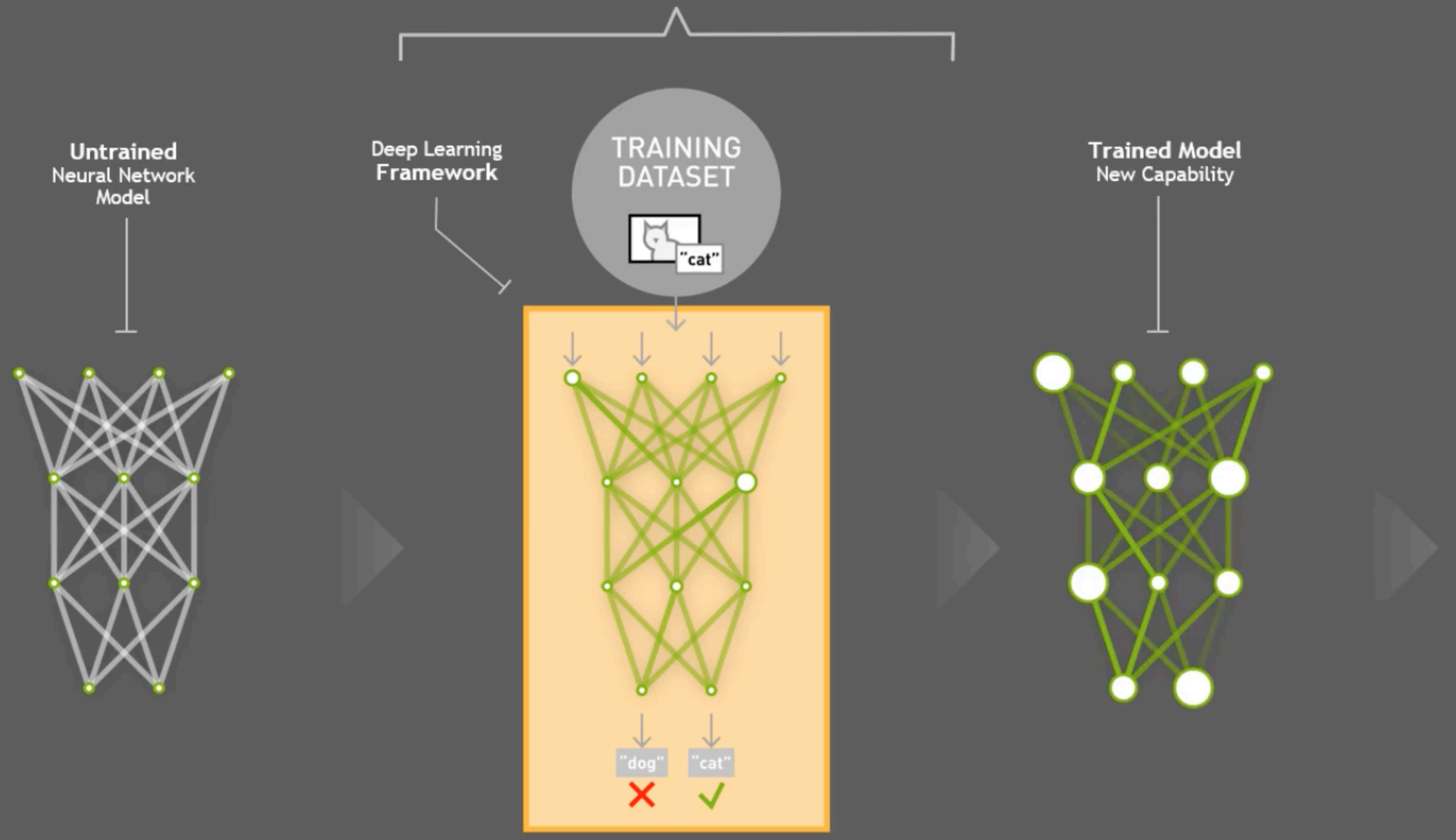
DEEP LEARNING APPLICATION DEVELOPMENT



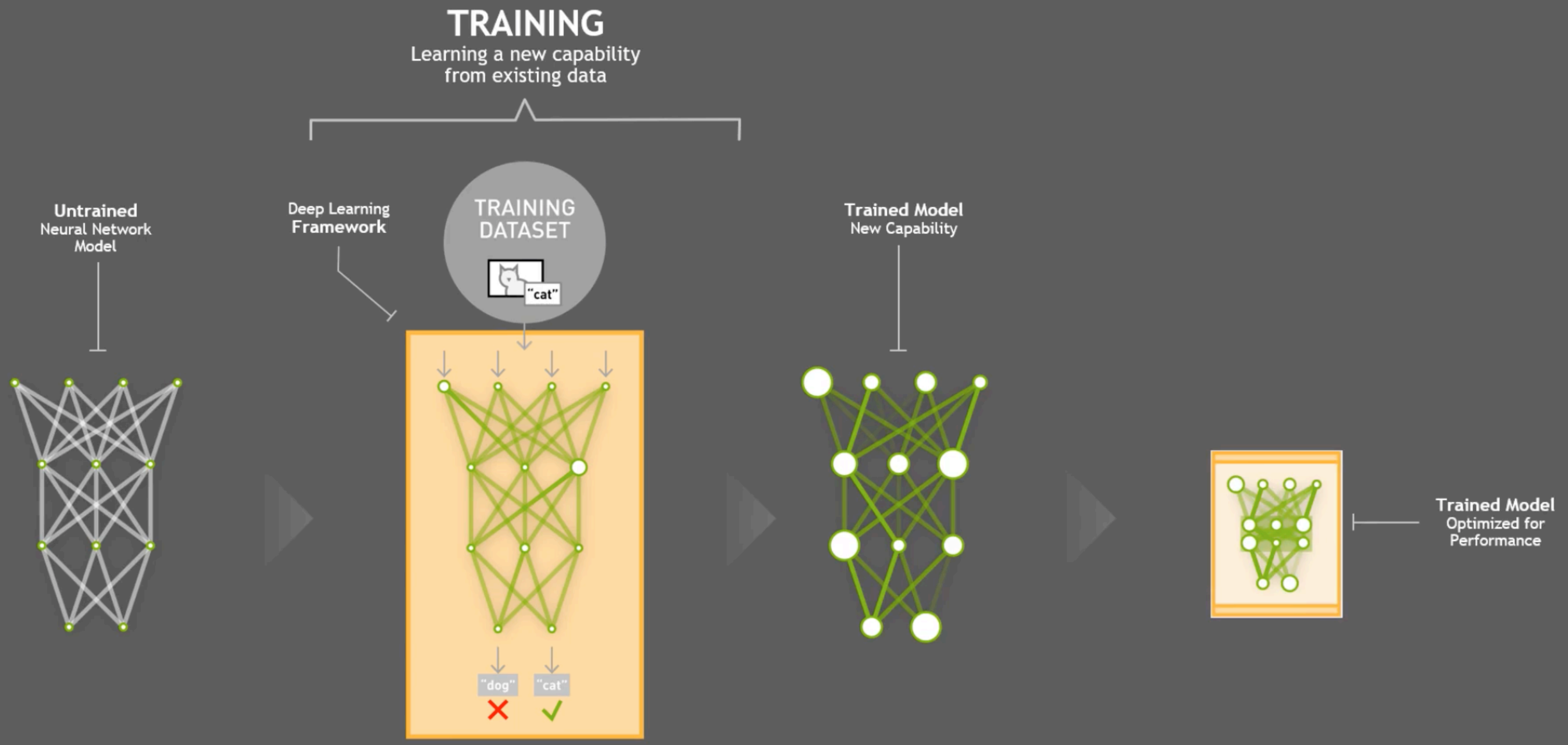
DEEP LEARNING APPLICATION DEVELOPMENT

TRAINING

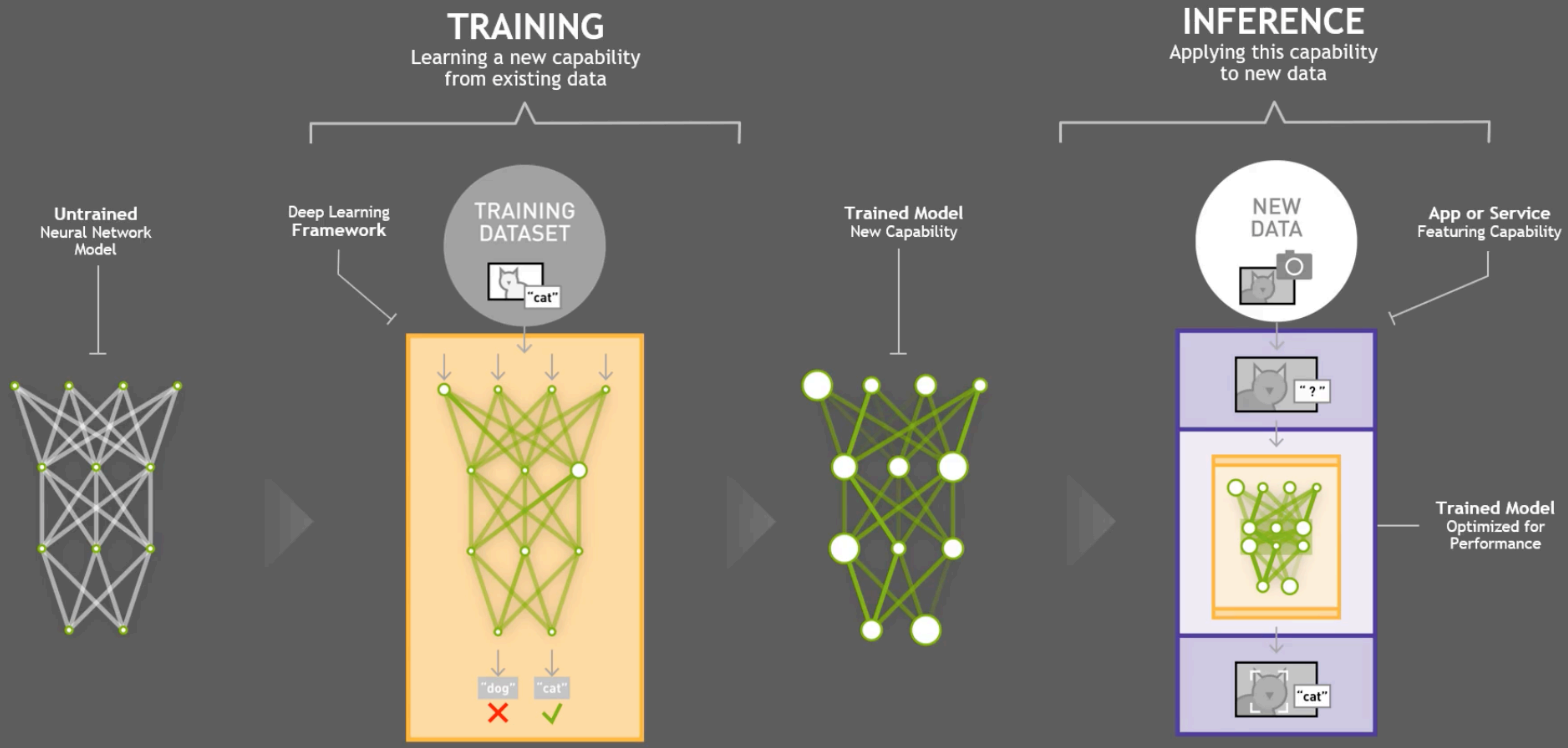
Learning a new capability
from existing data



DEEP LEARNING APPLICATION DEVELOPMENT



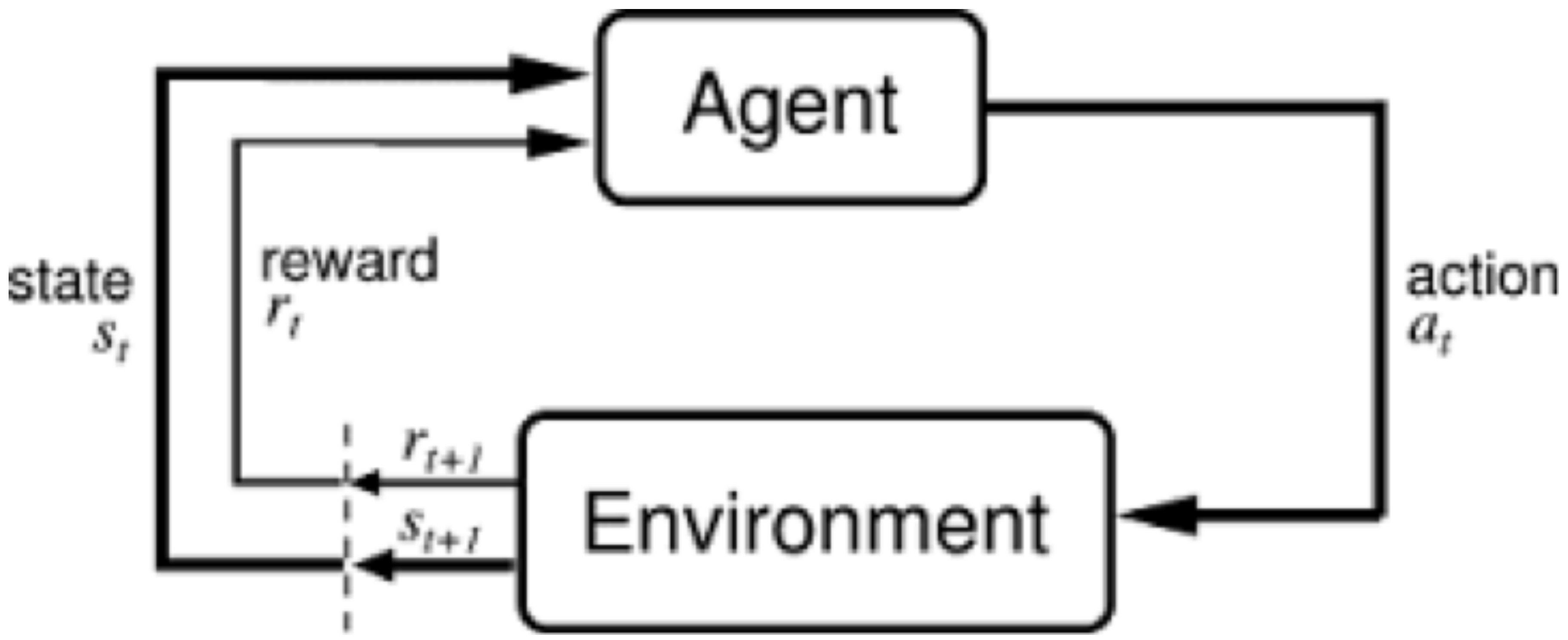
DEEP LEARNING APPLICATION DEVELOPMENT

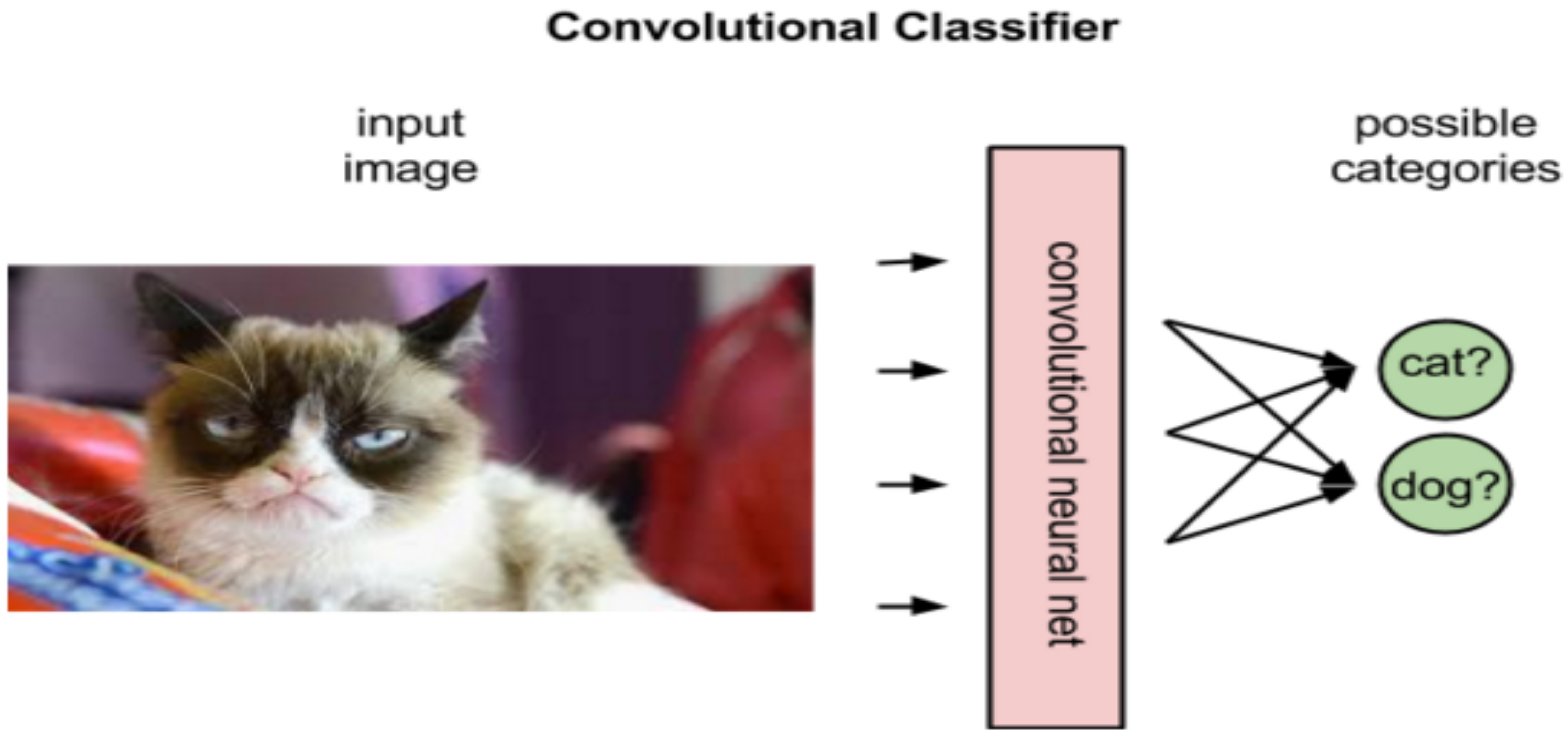


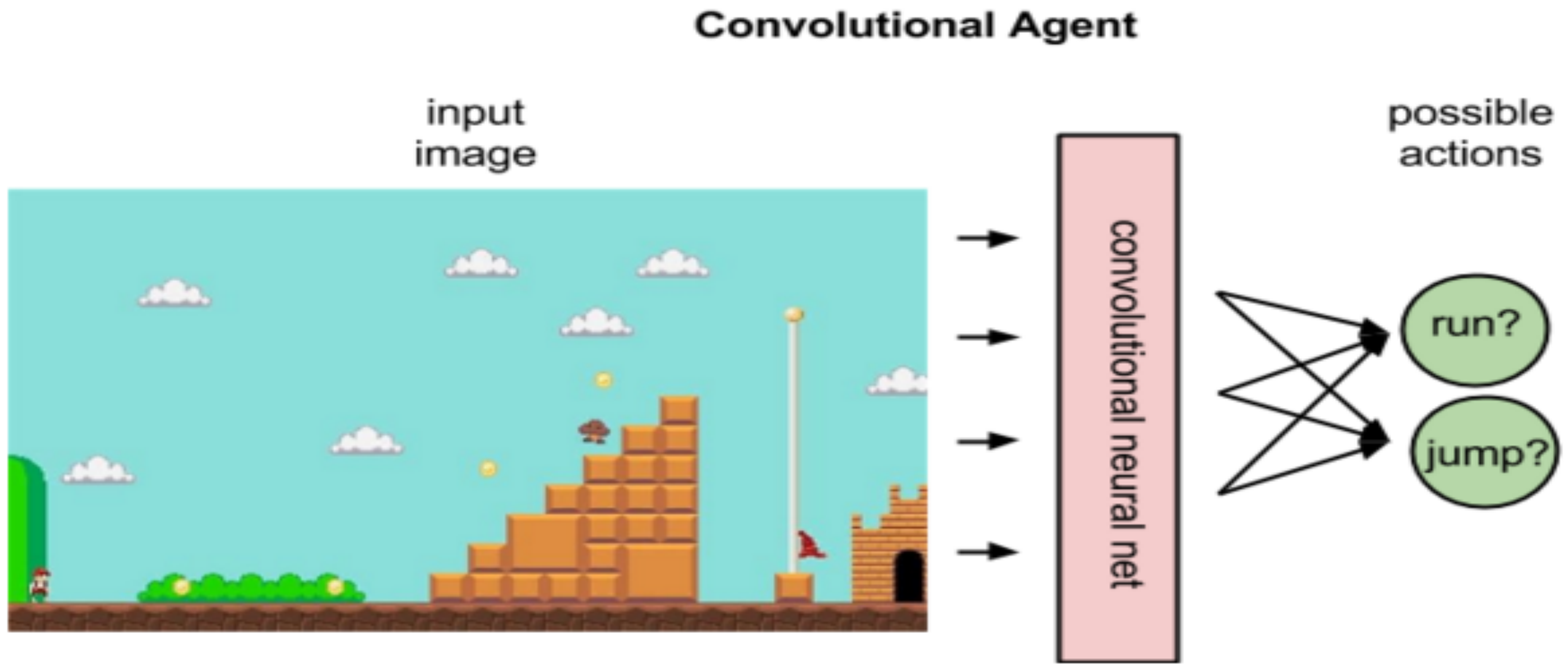
Reinforcement Learning

Train models in autonomous feedback-guided loops. It is used to implement environment exploring and reward driven agents that learn by doing.

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.



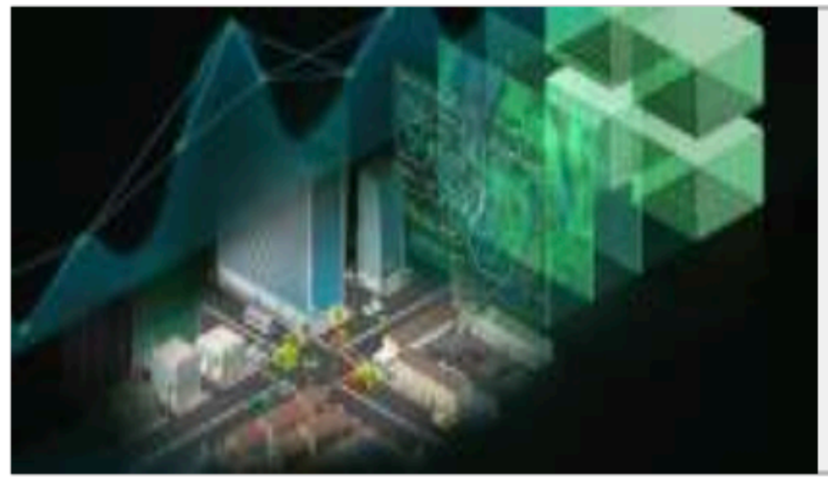




the why, how and **what** of
artificial intelligence

AI applications

Use case in every industry

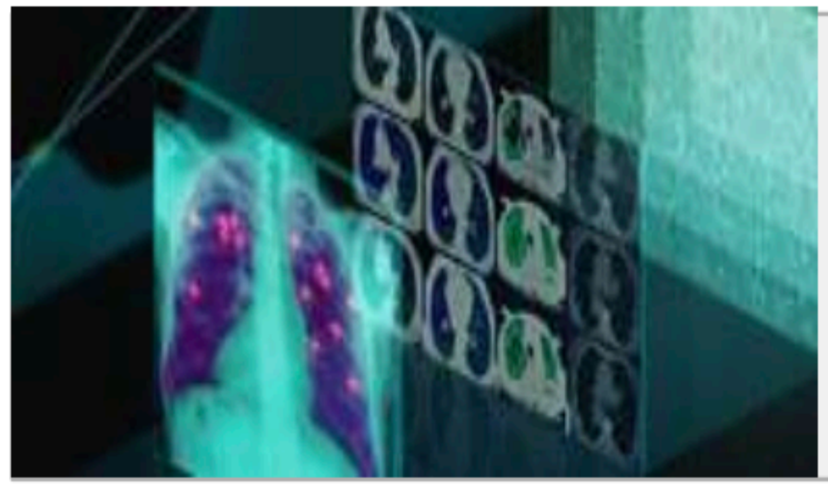


CONSUMER INTERNET

Ad Personalization
Click Through Rate Optimization
Churn Reduction

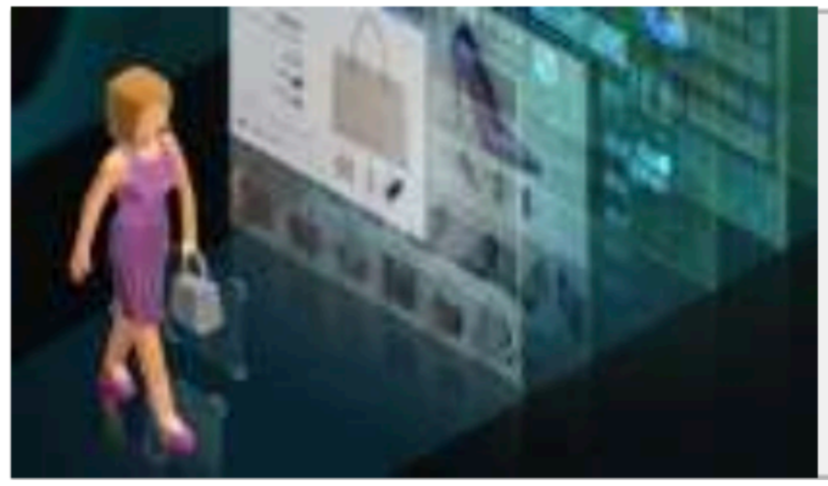
FINANCIAL SERVICES

Claim Fraud
Customer Service Chatbots/Routing
Risk Evaluation



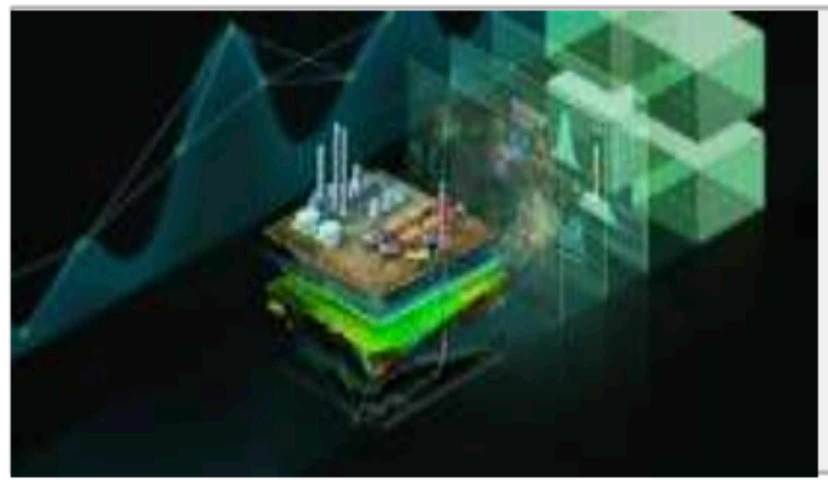
HEALTHCARE

Improve Clinical Care
Drive Operational Efficiency
Speed Up Drug Discovery



RETAIL

Supply Chain & Inventory Management
Price Management / Markdown Optimization
Promotion Prioritization And Ad Targeting



OIL & GAS

Sensor Data Tag Mapping
Anomaly Detection
Robust Fault Prediction

MANUFACTURING

Remaining Useful Life Estimation
Failure Prediction
Demand Forecasting

TELECOM

Detect Network/Security Anomalies
Forecasting Network Performance
Network Resource Optimization (SON)

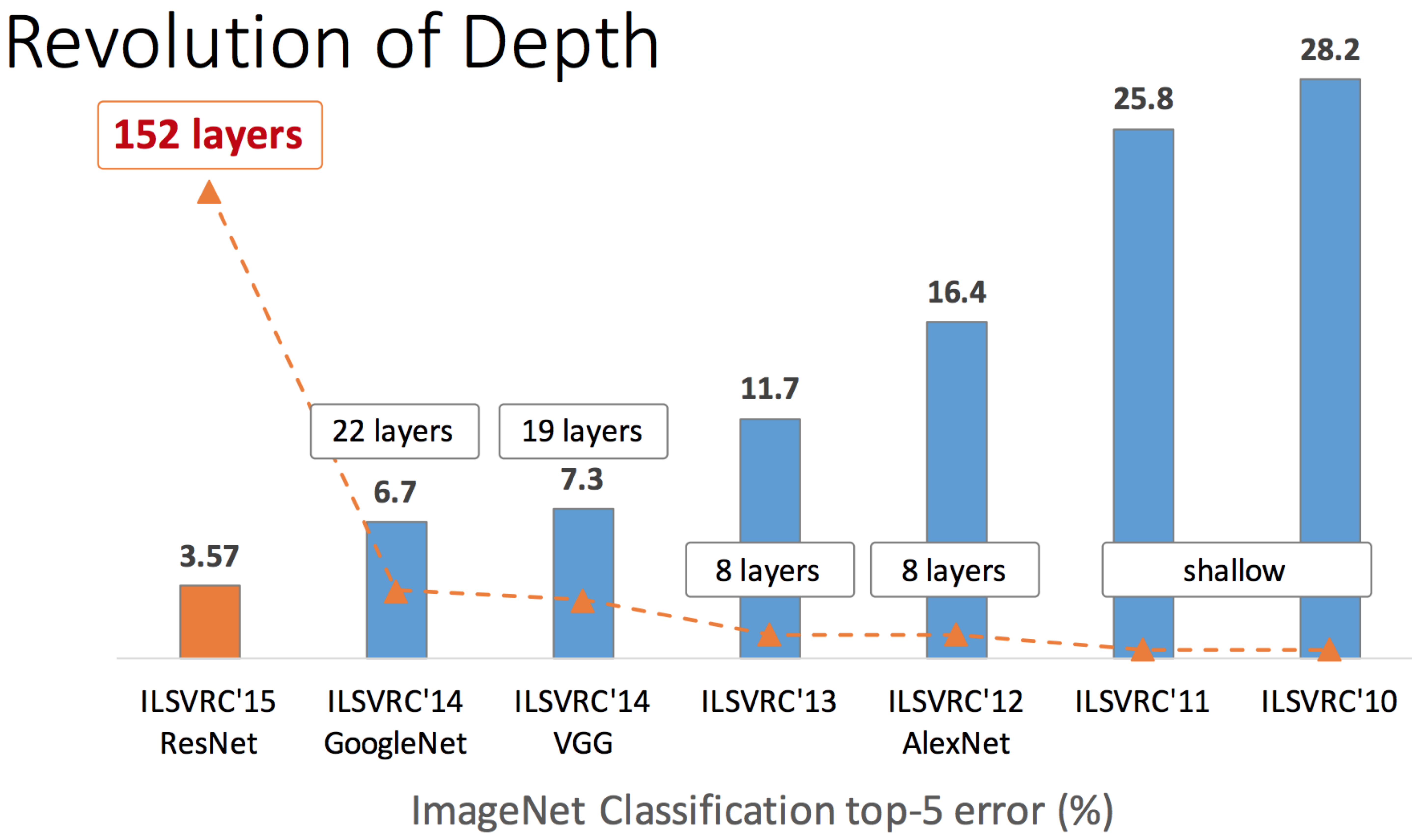
AUTOMOTIVE

Personalization & Intelligent Customer Interactions
Connected Vehicle Predictive Maintenance
Forecasting, Demand, & Capacity Planning

Convolutional Neural Networks

Convolutional Neural Network (CNN)

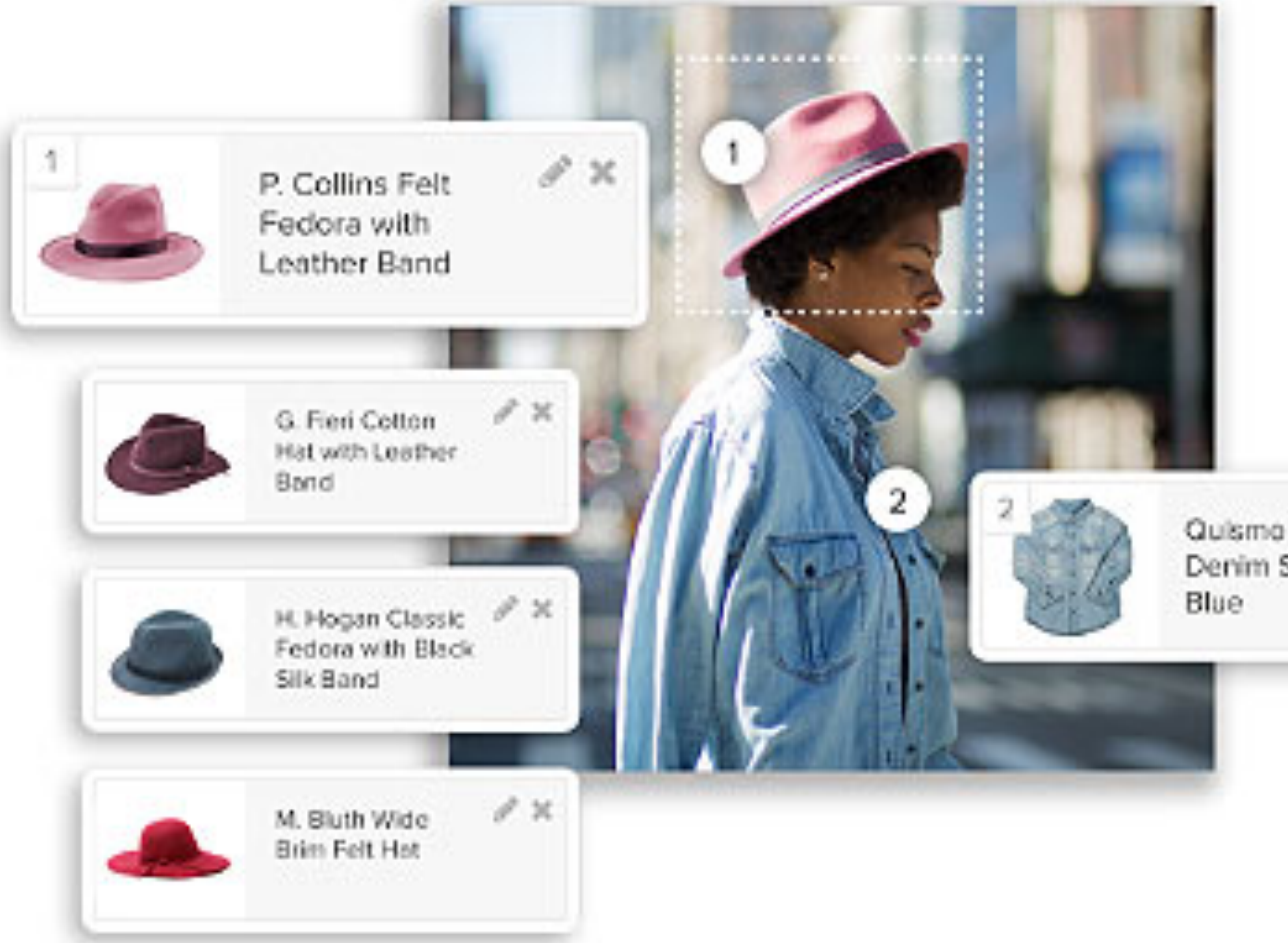
First CNN was developed by Yann LeCun on 1988, called **LeNet**, but CNNs became popular when in 2012 **AlexNet** was the first CNN to win the ImageNet Large Scale Visual Recognition Challenge (ILSVCR). Since then, only DNN model where used (and won) the following editions.



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.

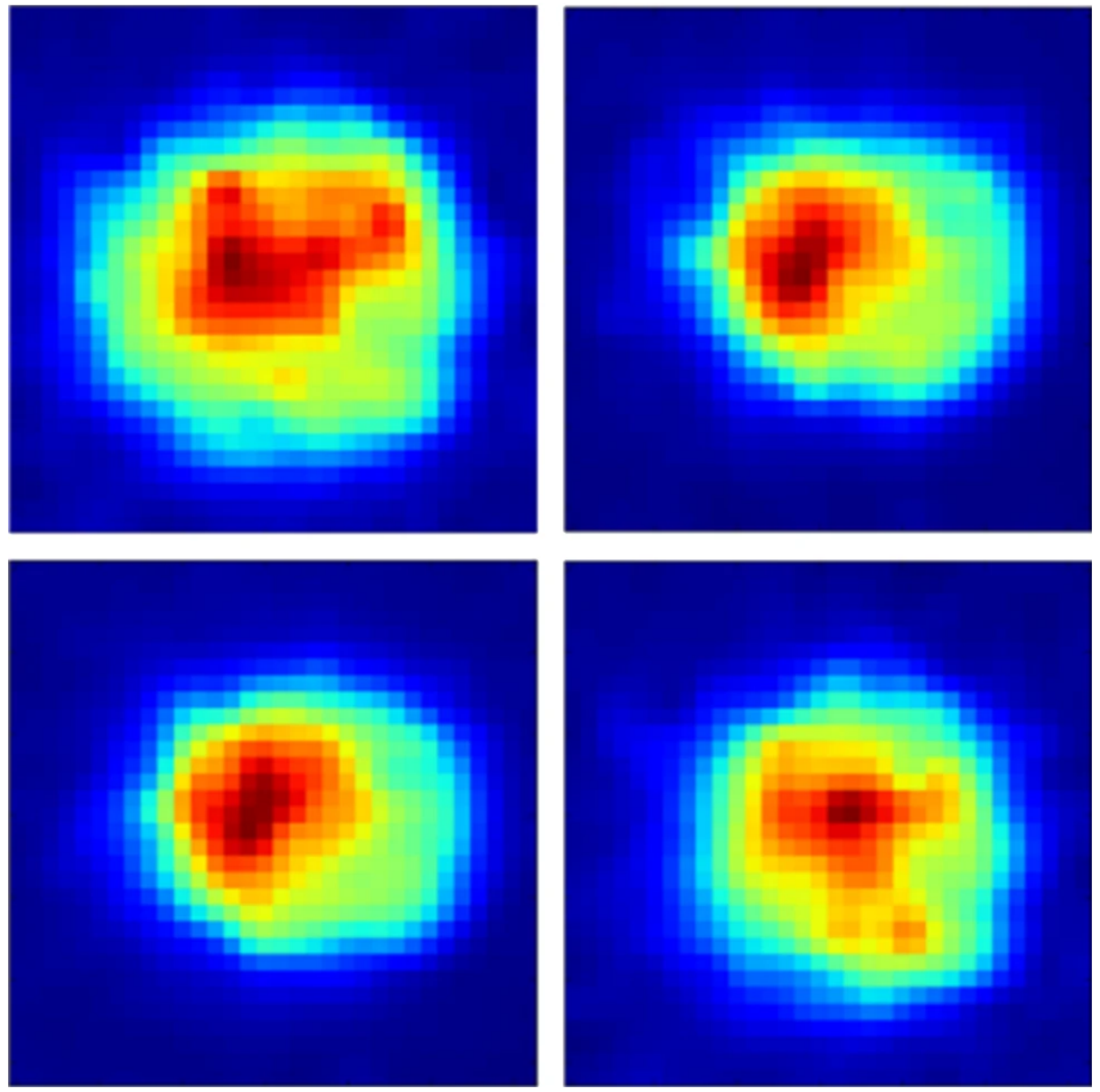
- Automatically tag products
- Cut down on workload to categorize products
- Show related products
- Find cheaper version of high end products
- Find complimentary products
- Find products usage on social media

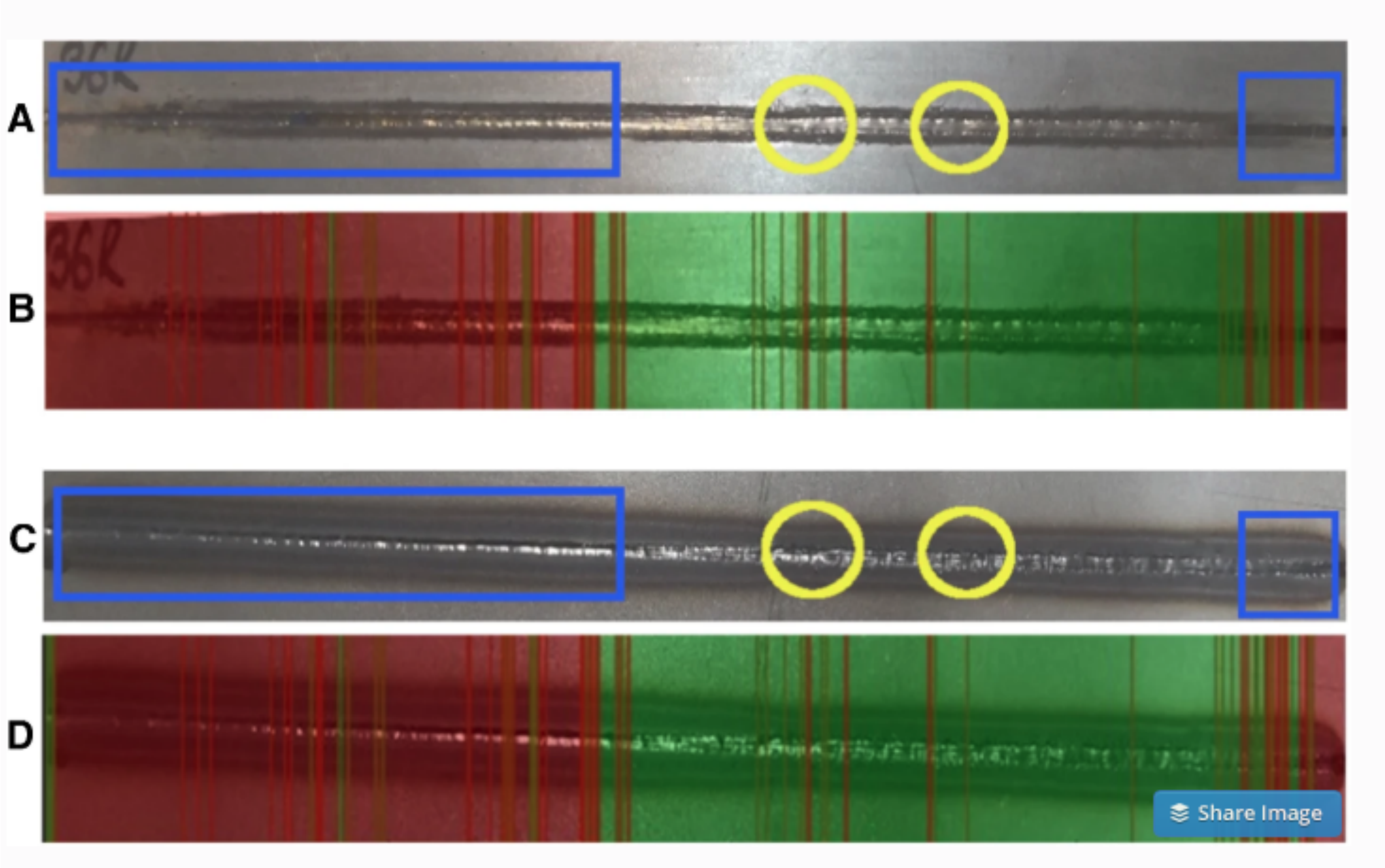


Quality inspection

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.



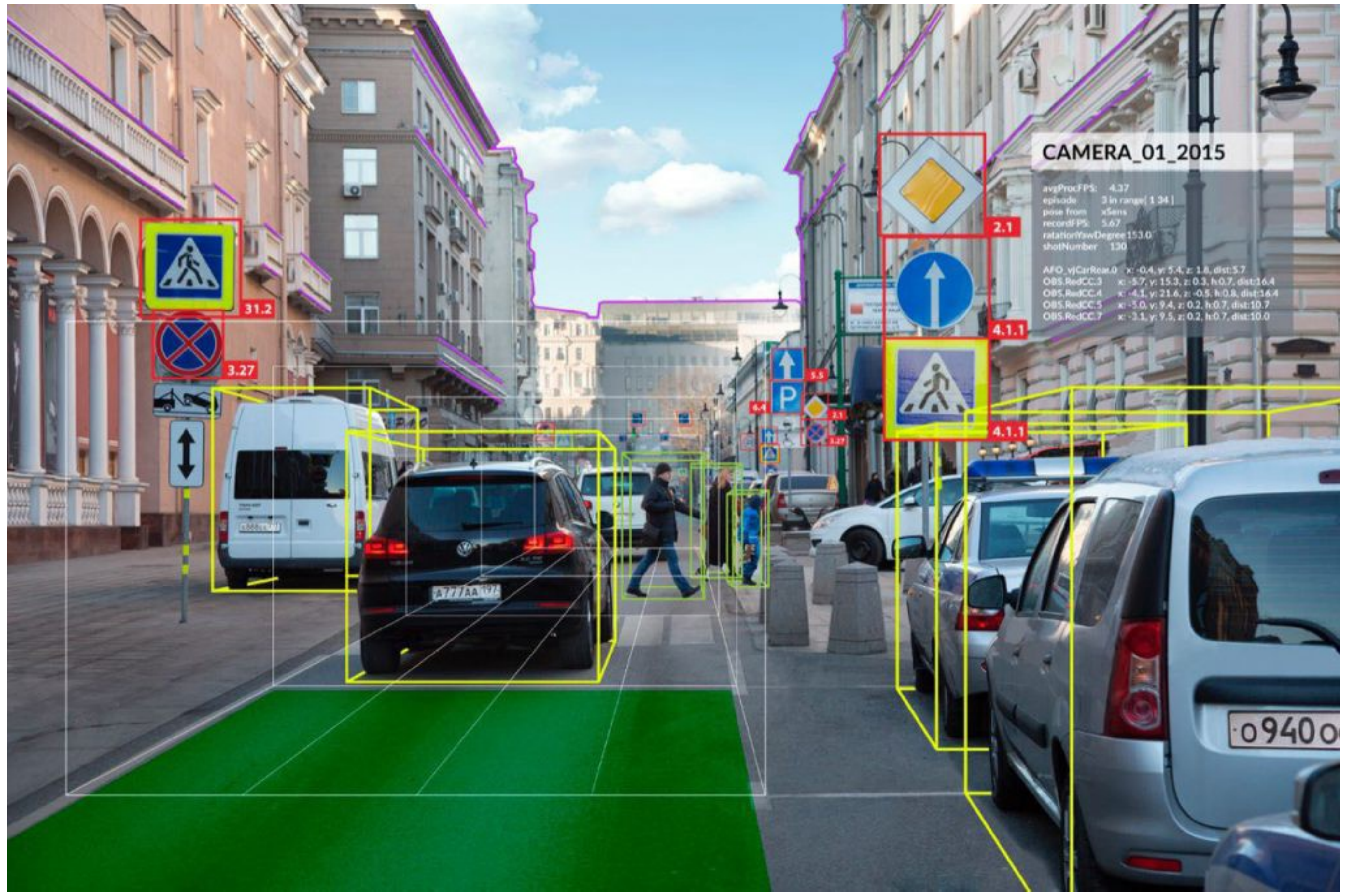


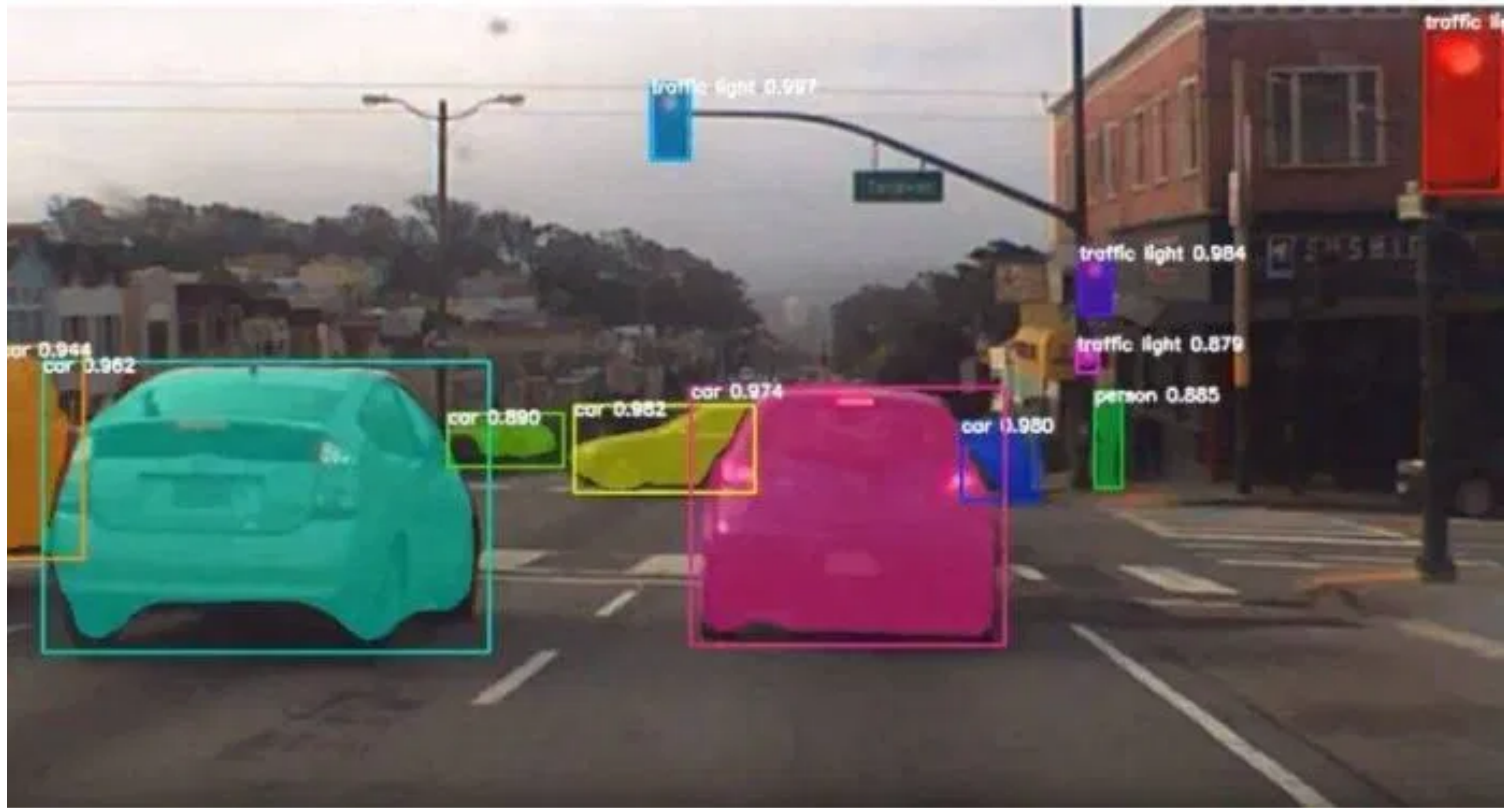
Real-time defect detection on a laser weld bead. a and c show two side views of the weld bead where the blue rectangles mark a defective section in the first and final segments due to undercuts and the yellow ellipses mark a region where some points have excessive porosity

Self Driving Cars

Detect accesses, obstacles and react

- Uses a model ensemble 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

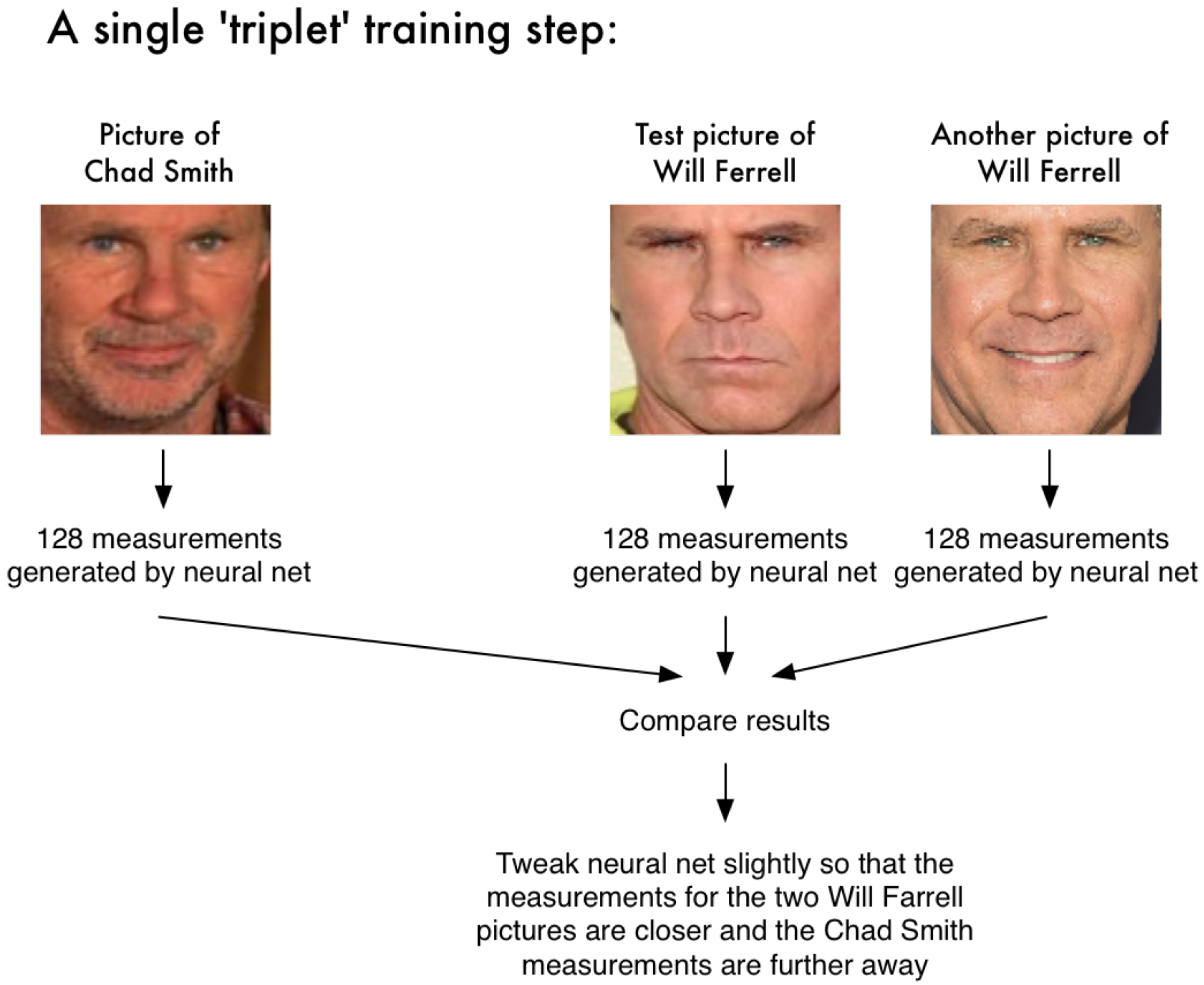
Use customer face as key to unlock credit card informations in a third party store

- 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

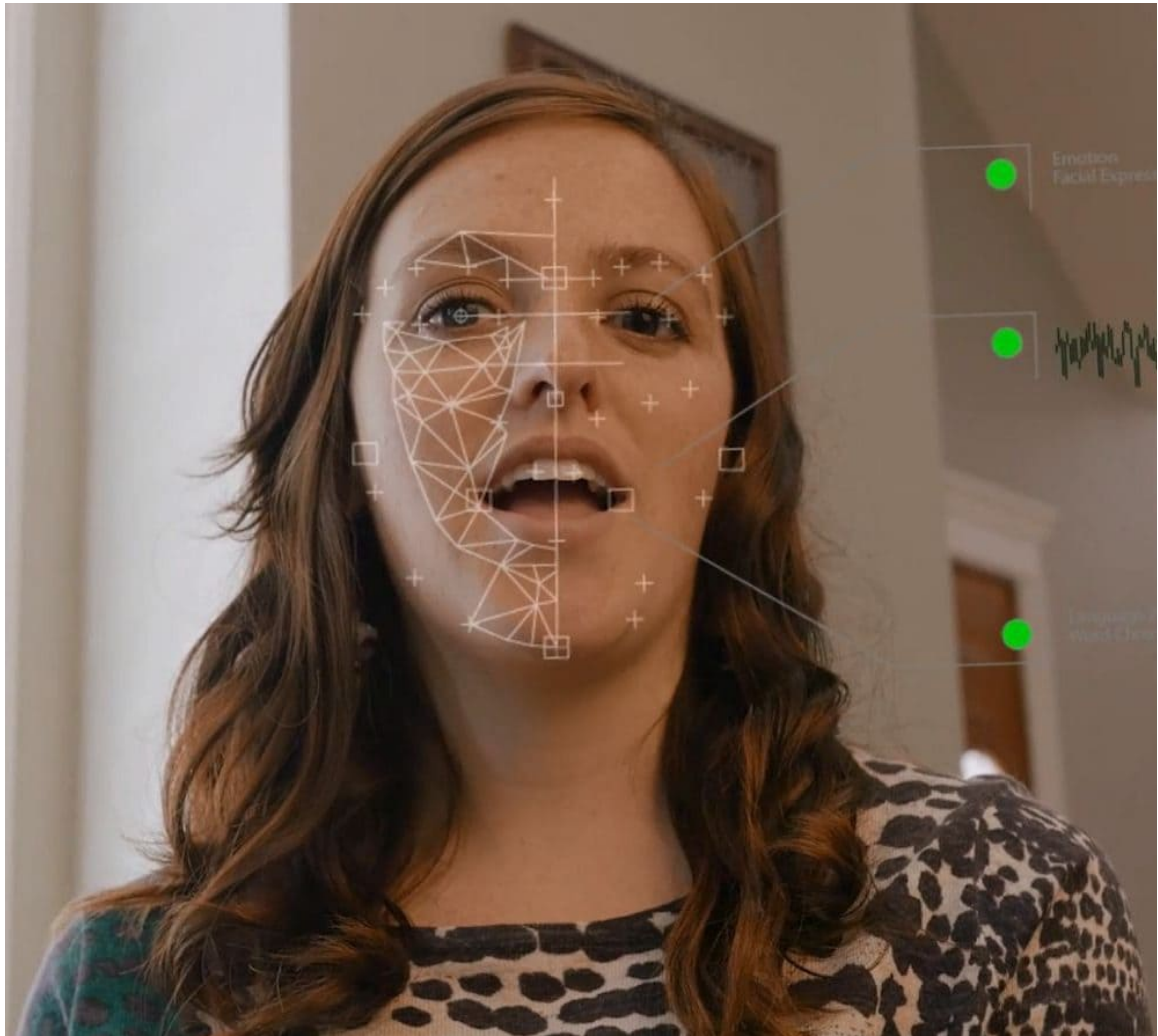
- Sometimes we have to train a network not to recognize a given object, but to tell whether an image is or is not a given person of interest.
- A common technique is to define a particular loss function named Triplet Loss.
- DLIB network extracts landmarks from a face (named measurements), then trains 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.



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

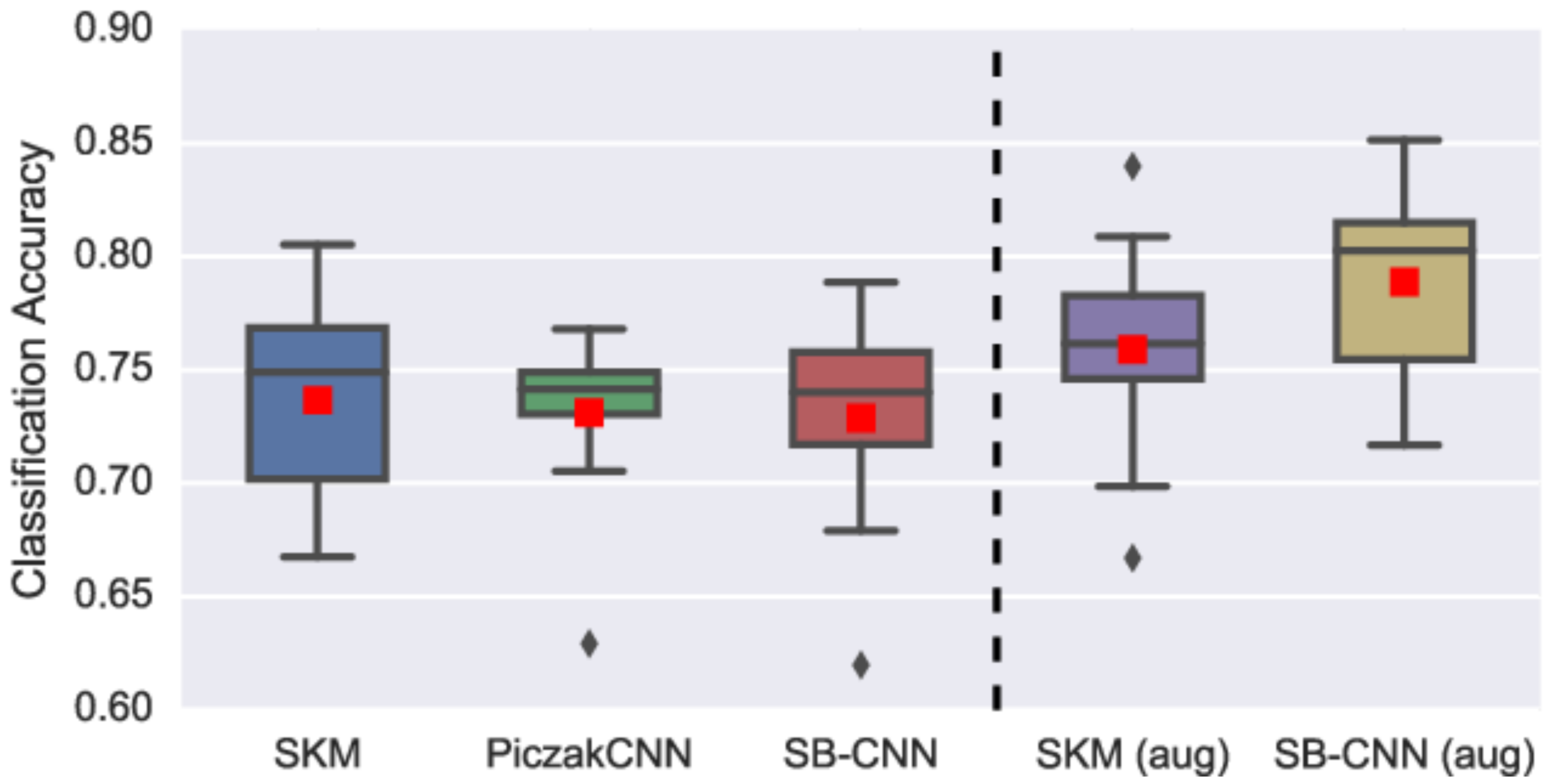
China is the current biggest investor on Computer Vision applications, with focus on schools and performance monitoring

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

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



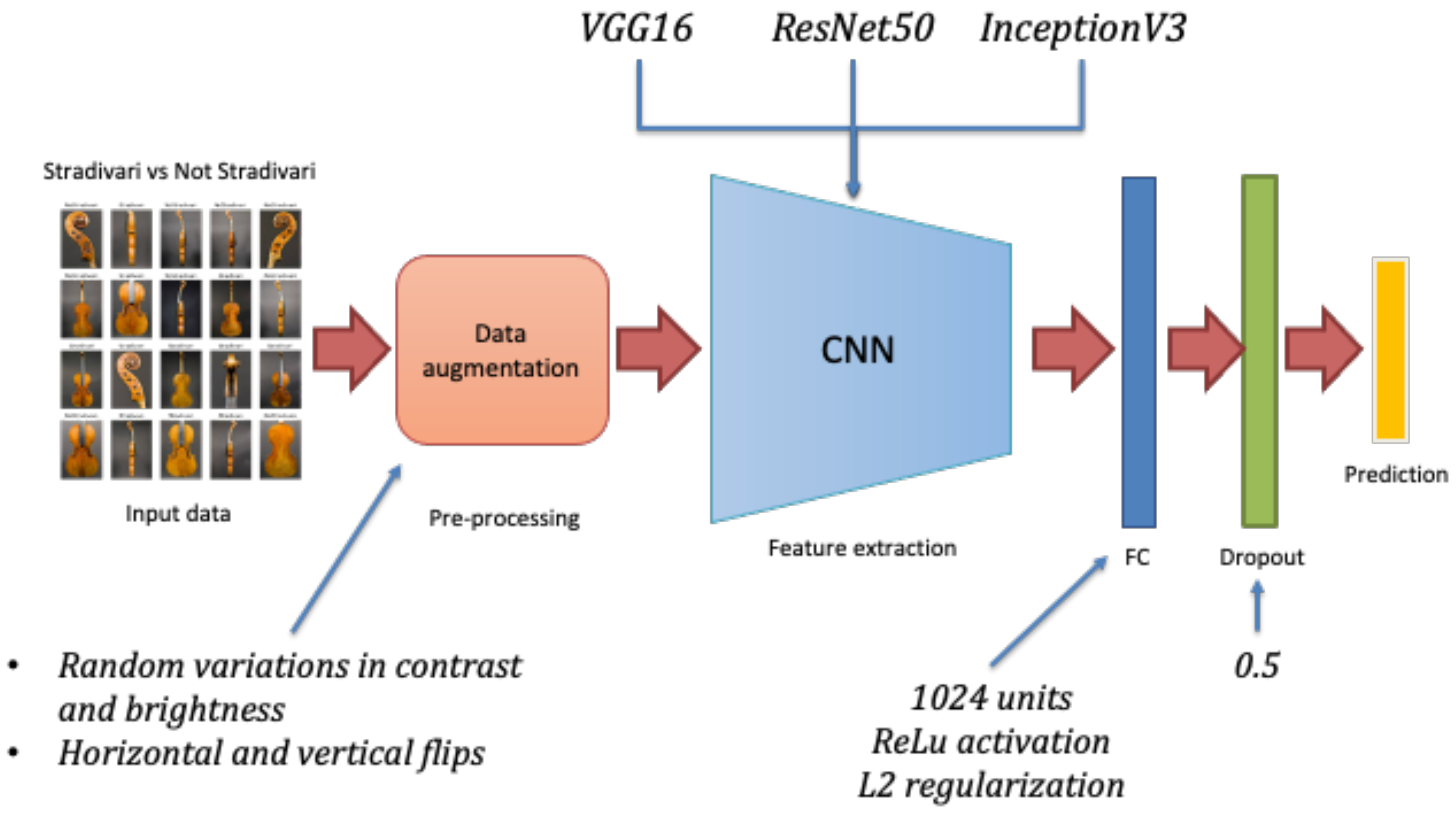
stylistic analysis

Stylistic analysis

Dondi P., Lombardi L., Malagodi M., Licchelli M. (2021)

"Stylistic classification of historical violins: a deep learning approach"

in ICPR International Workshops and Challenges. Lecture Notes in Computer Science, vol 12667, pp. 112-125, DOI: 10.1007/978-3-030-68787-8_8



Stylistic analysis

Average performances using all images

Model	Accuracy	Recall	Precision	F1-score
VGG16	0.7727	0.6842	0.7647	0.7222
ResNet50	0.7500	0.6842	0.7222	0.7027
InceptionV3	0.6818	0.7895	0.6000	0.6818

Average performances using only images of the body

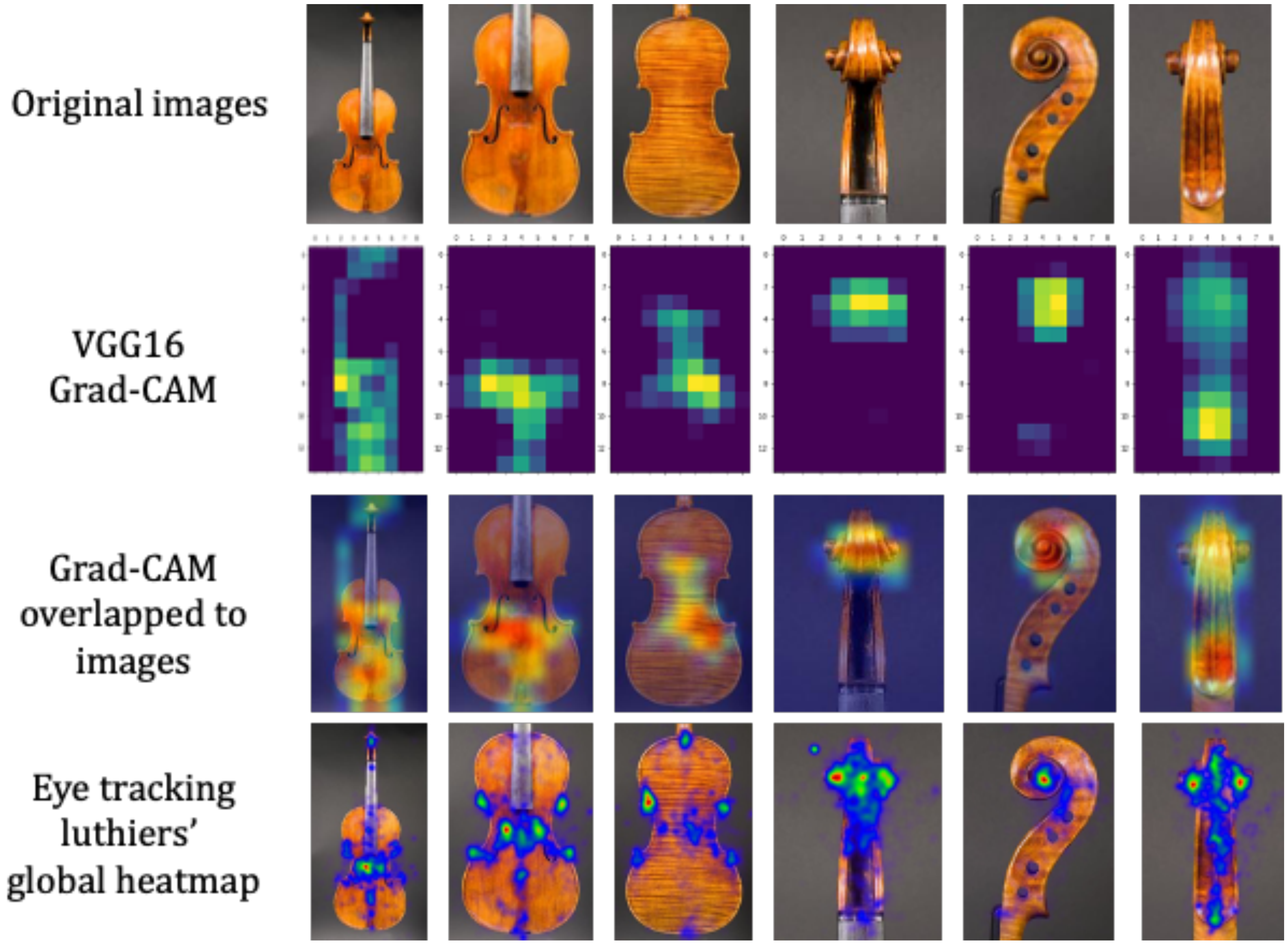
Model	Accuracy	Recall	Precision	F1-score
VGG16	0.7585	0.6260	0.7625	0.6859
ResNet50	0.7401	0.6466	0.7310	0.6825
InceptionV3	0.6715	0.6467	0.6201	0.6253

Average performances using only images of the head

Model	Accuracy	Recall	Precision	F1-score
VGG16	0.7646	0.7383	0.7597	0.7392
ResNet50	0.7704	0.7036	0.7743	0.7341
InceptionV3	0.7667	0.6667	0.7951	0.7233

CVMLab

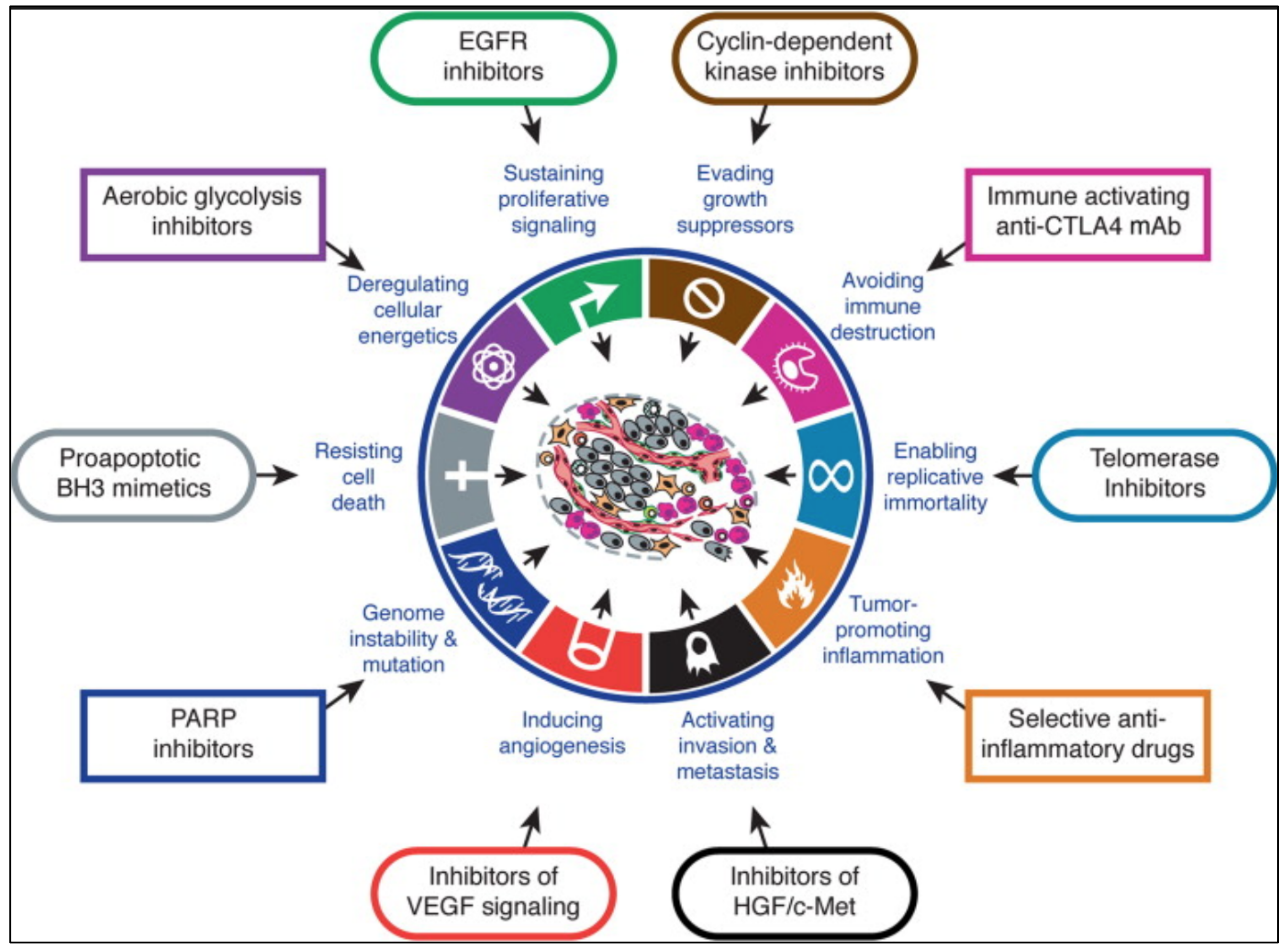
Stylistic analysis

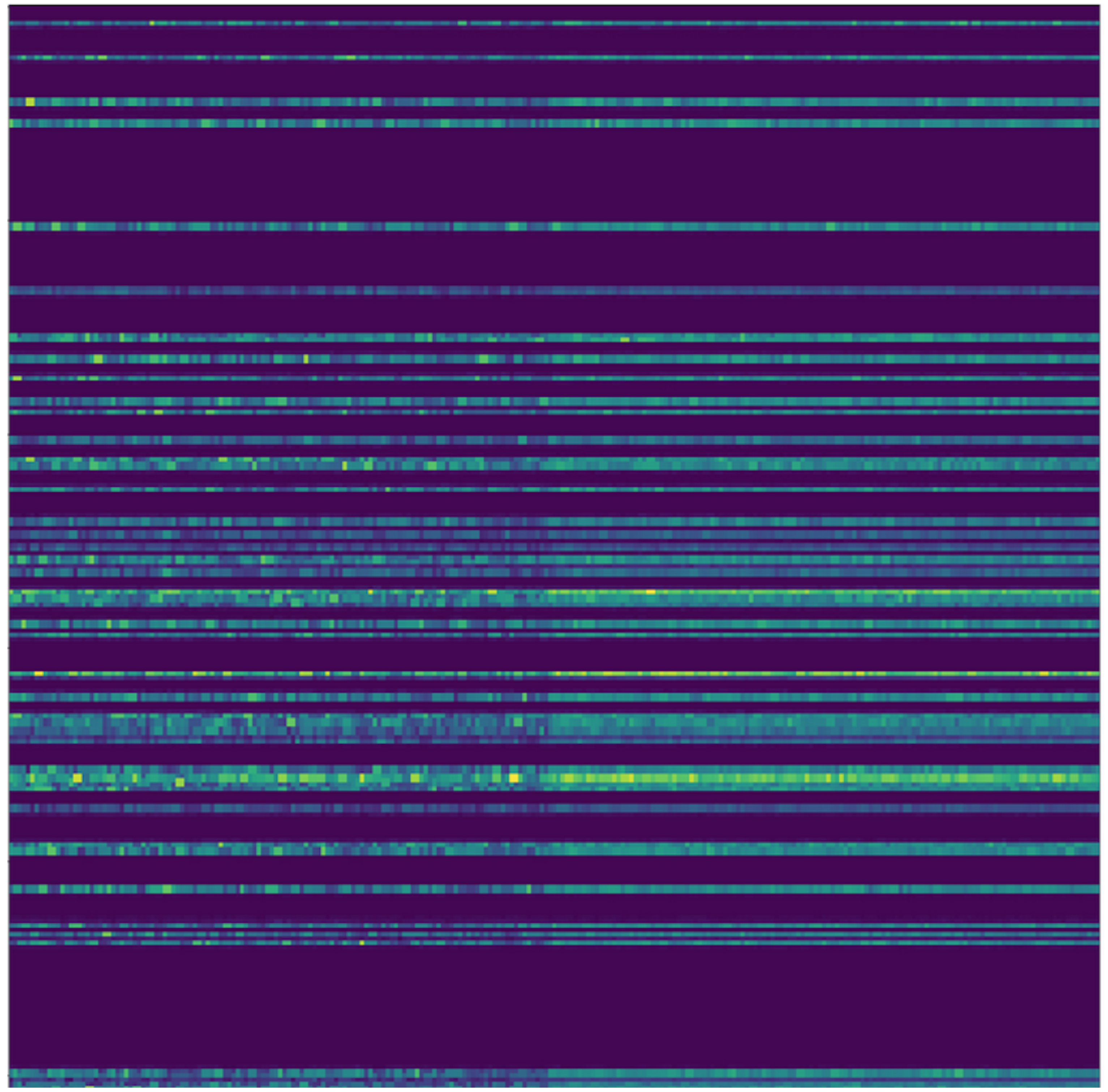


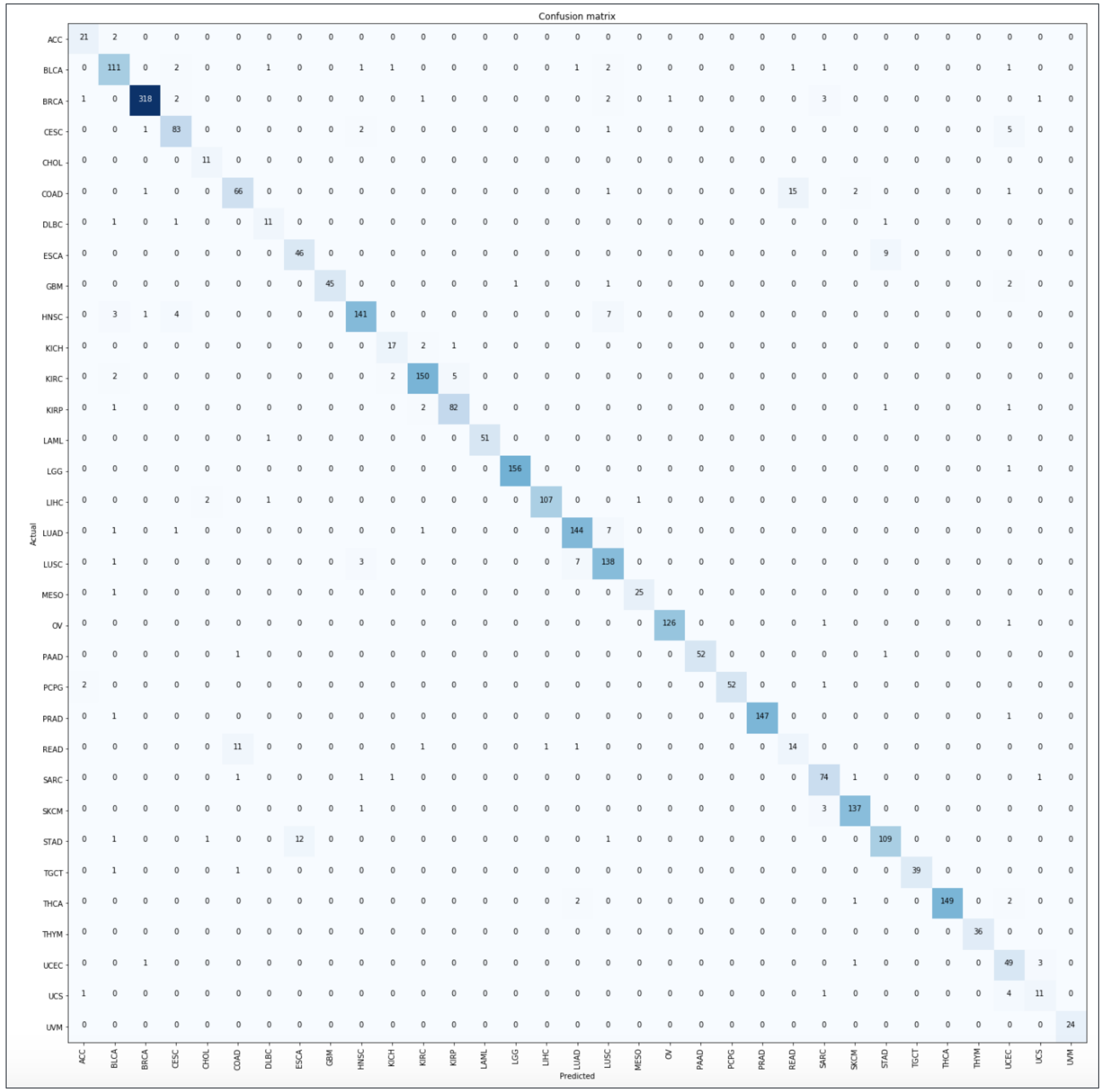
empowering health
and medical diagnosis

Cancer Type Classification

Cancer Type Classification using CNN and Fast.AI





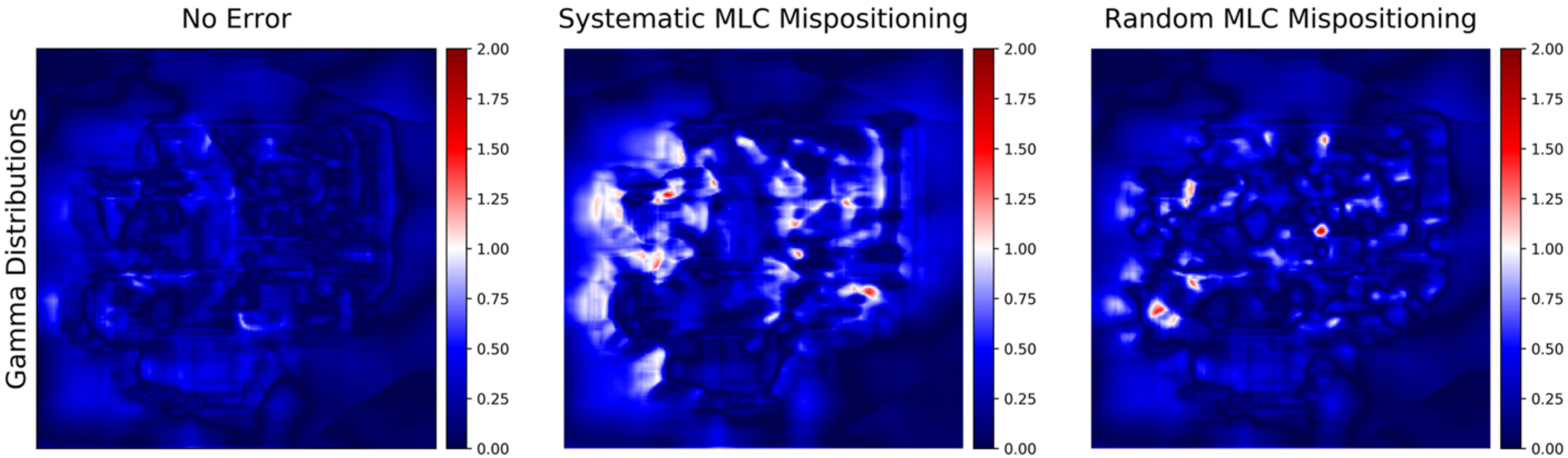


https://towardsdatascience.com/the-mystery-of-the-origin-cancer-type-classification-using-fast-ai-libray-212eaf8d3f4e

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.

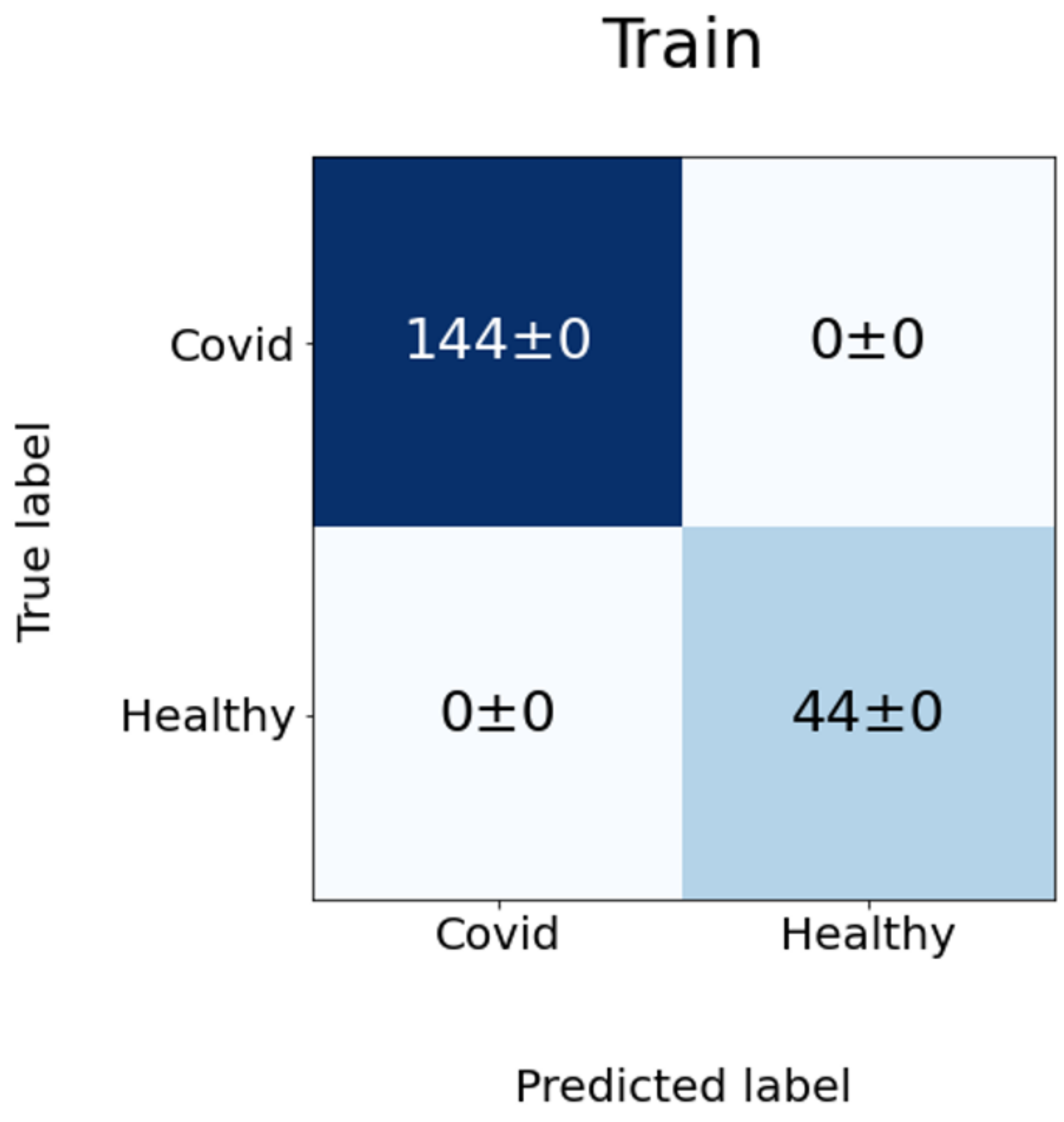


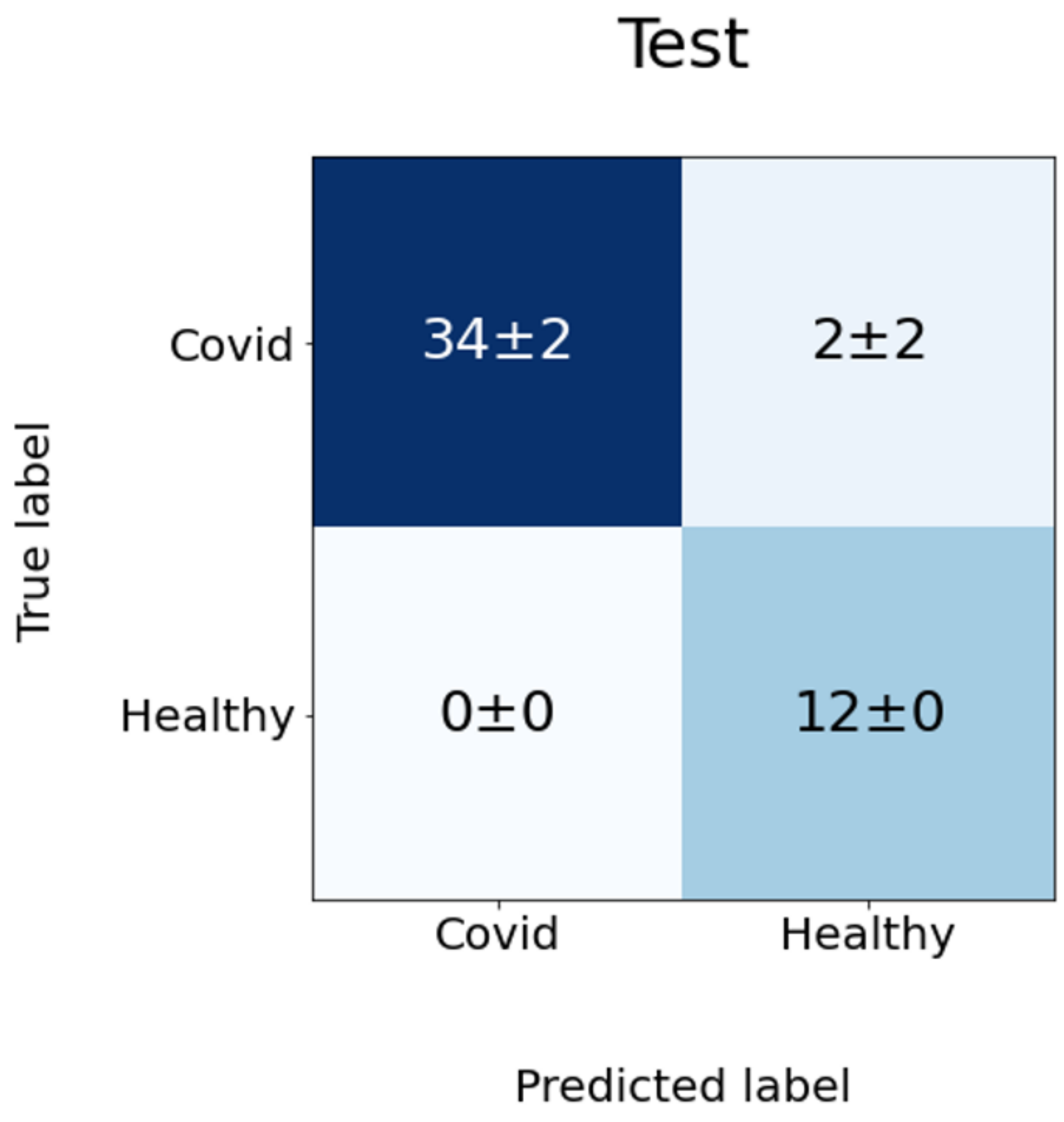
Dimasdia — COVID-19 RX Detector

Building a dataset for transfer learning

Dimasdia — COVID-19 RX Detector

Model performances





Dimasdia — COVID-19 RX Detector

Model Validation

380 x-rays
- 25 covid
- 150 healthy
- 205 pneumonia

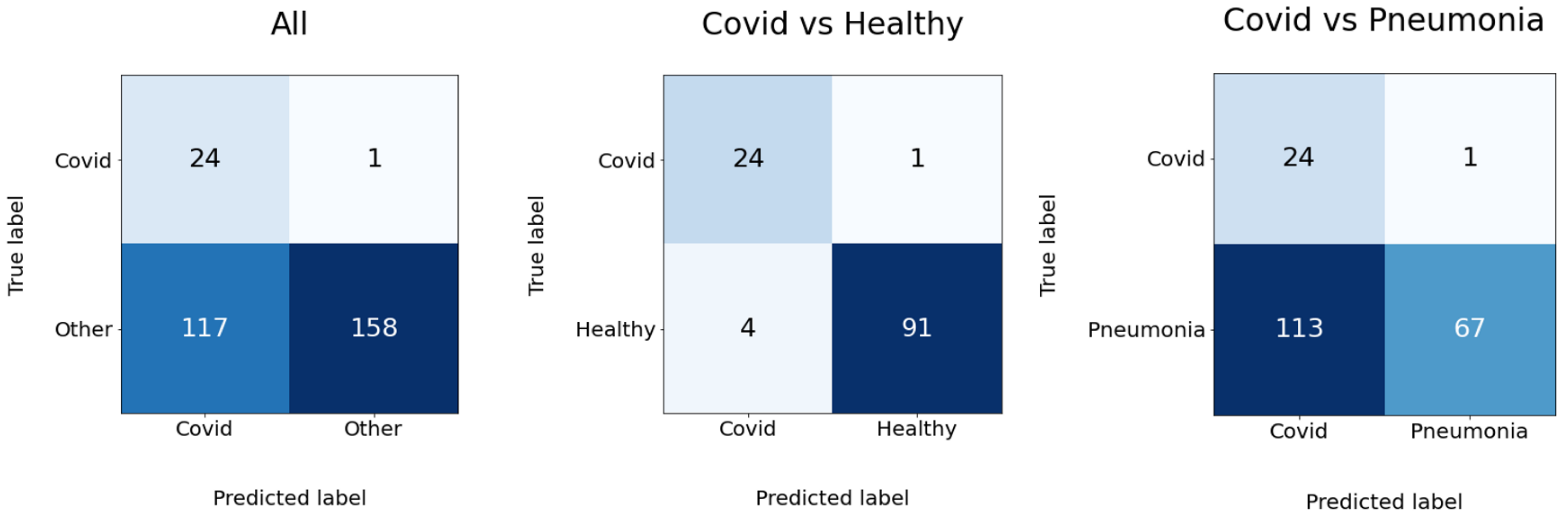
→

- remove 78 RL view
- remove 2 corrupted files

300 x-rays
- 25 covid
- 95 healthy
- 180 pneumonia

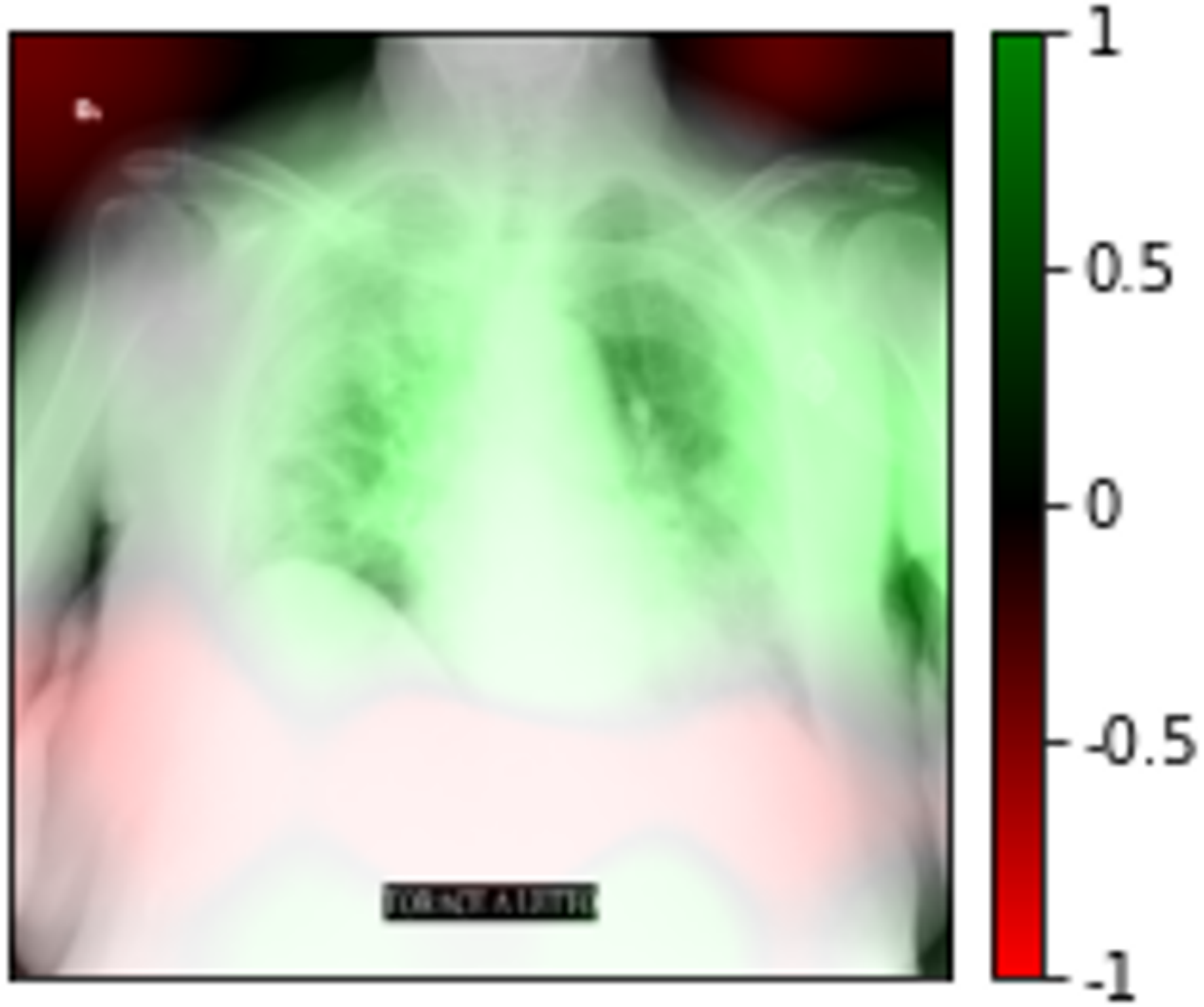
Dimasdia — COVID-19 RX Detector

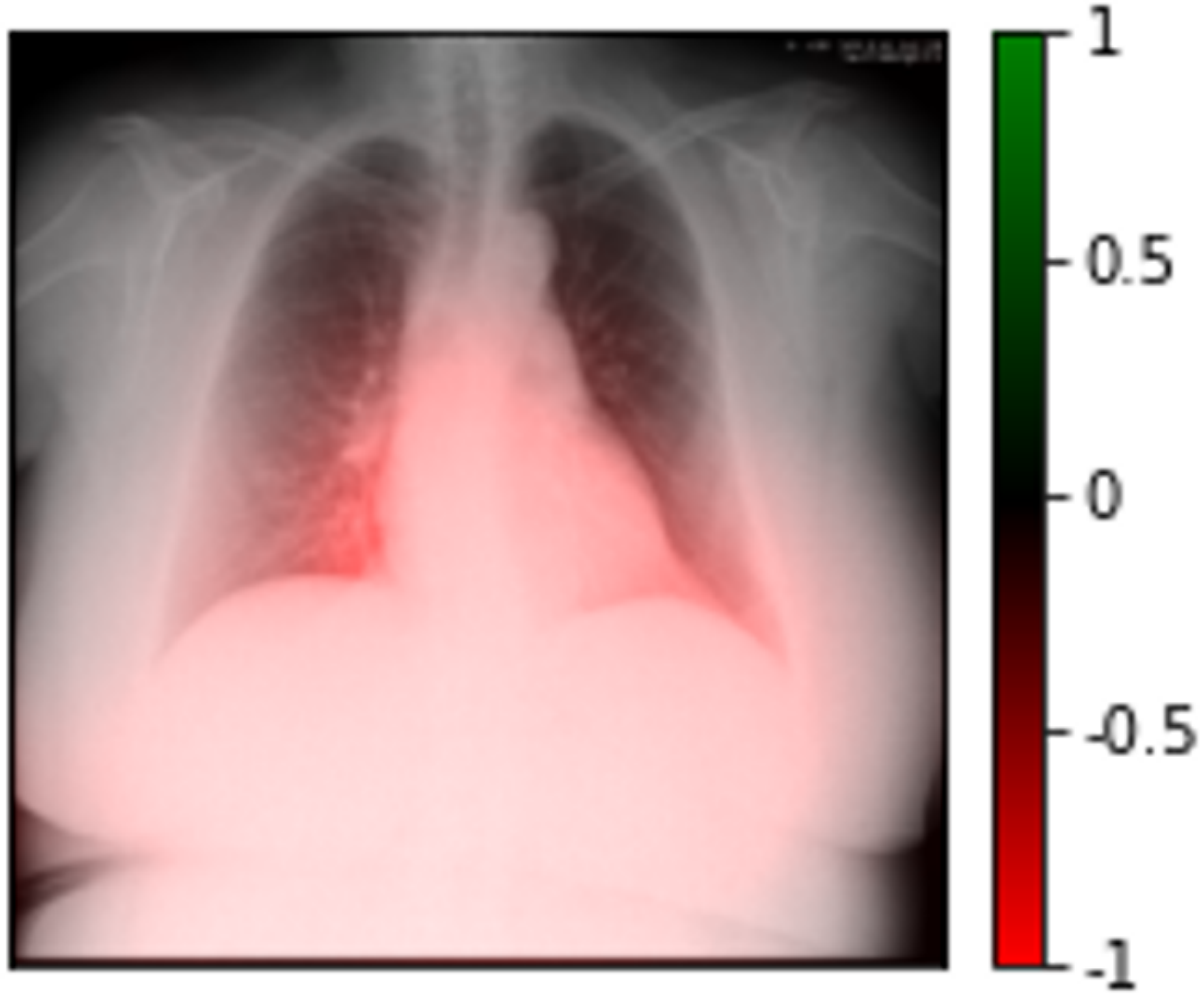
Model Evaluation: good to predict pneumonia, must be improved to be used after pandemic



Dimasdia — COVID-19 RX Detector

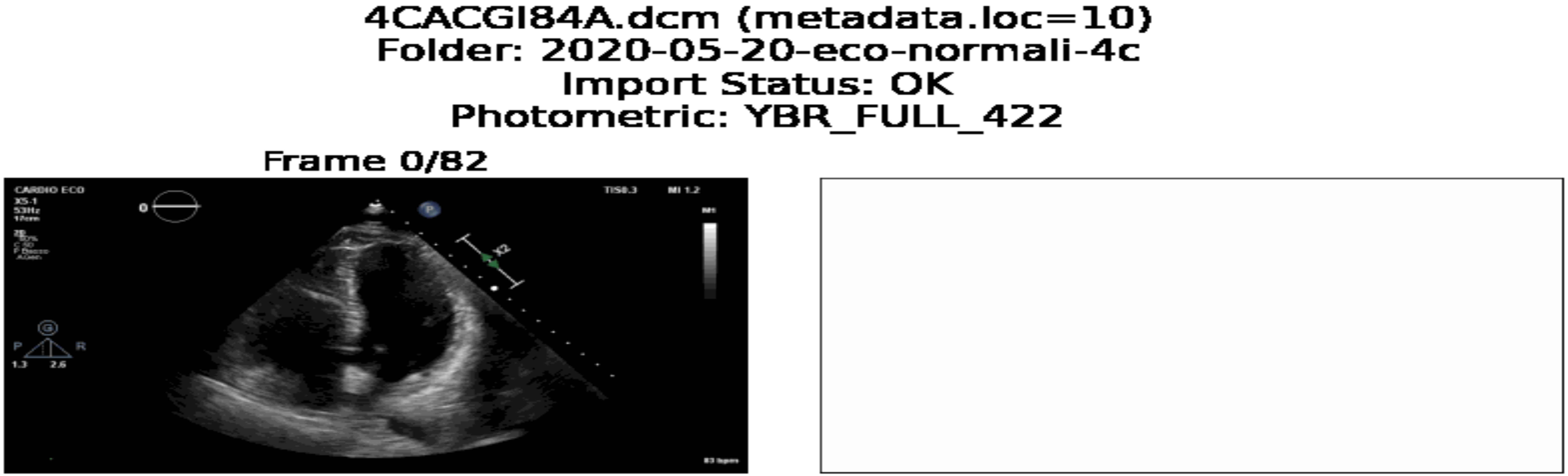
Model Validation





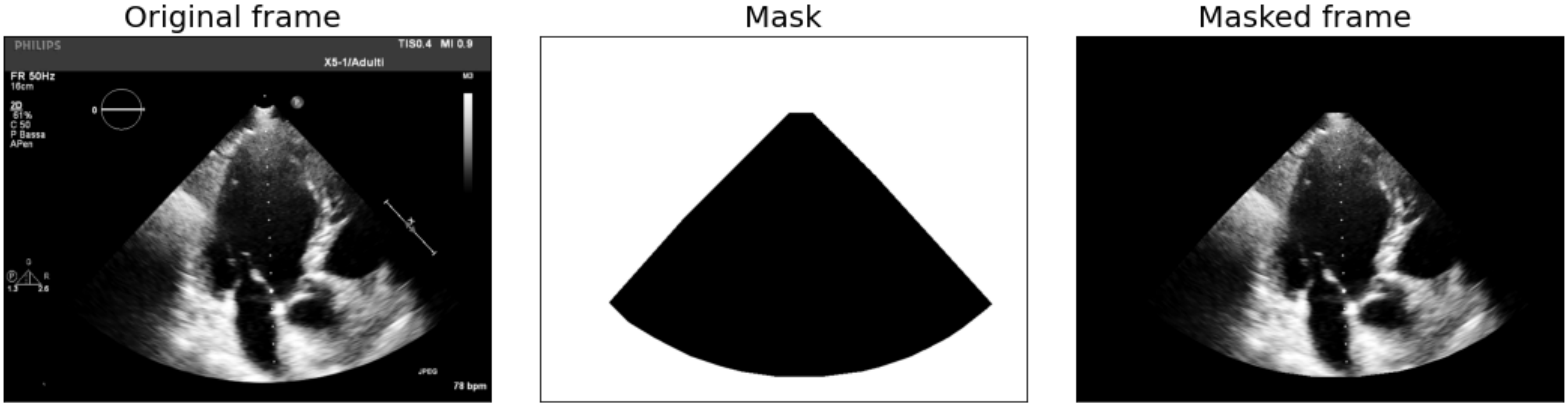
Dimasdia — miocardic akinesia

Ultrasound images



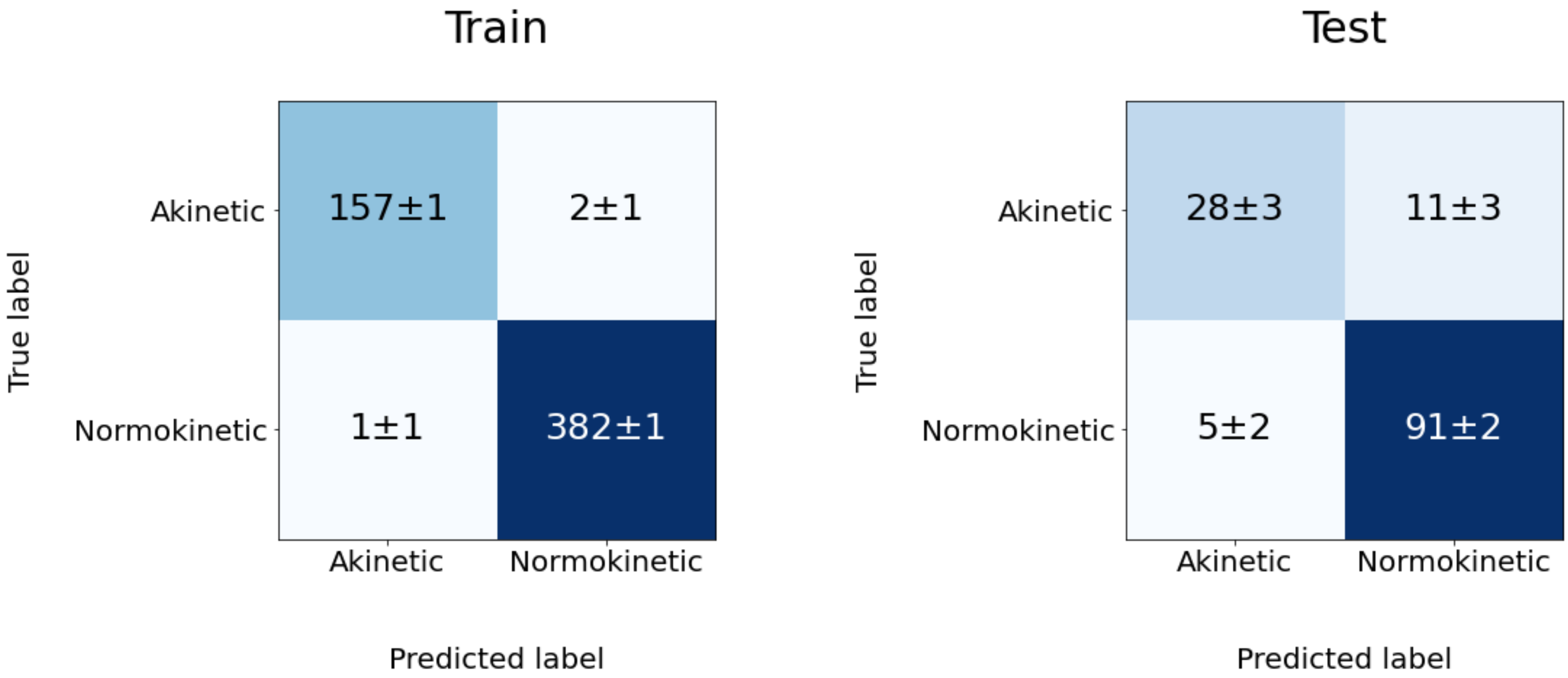
Dimasdia — miocardic akinesia

Ultrasound images



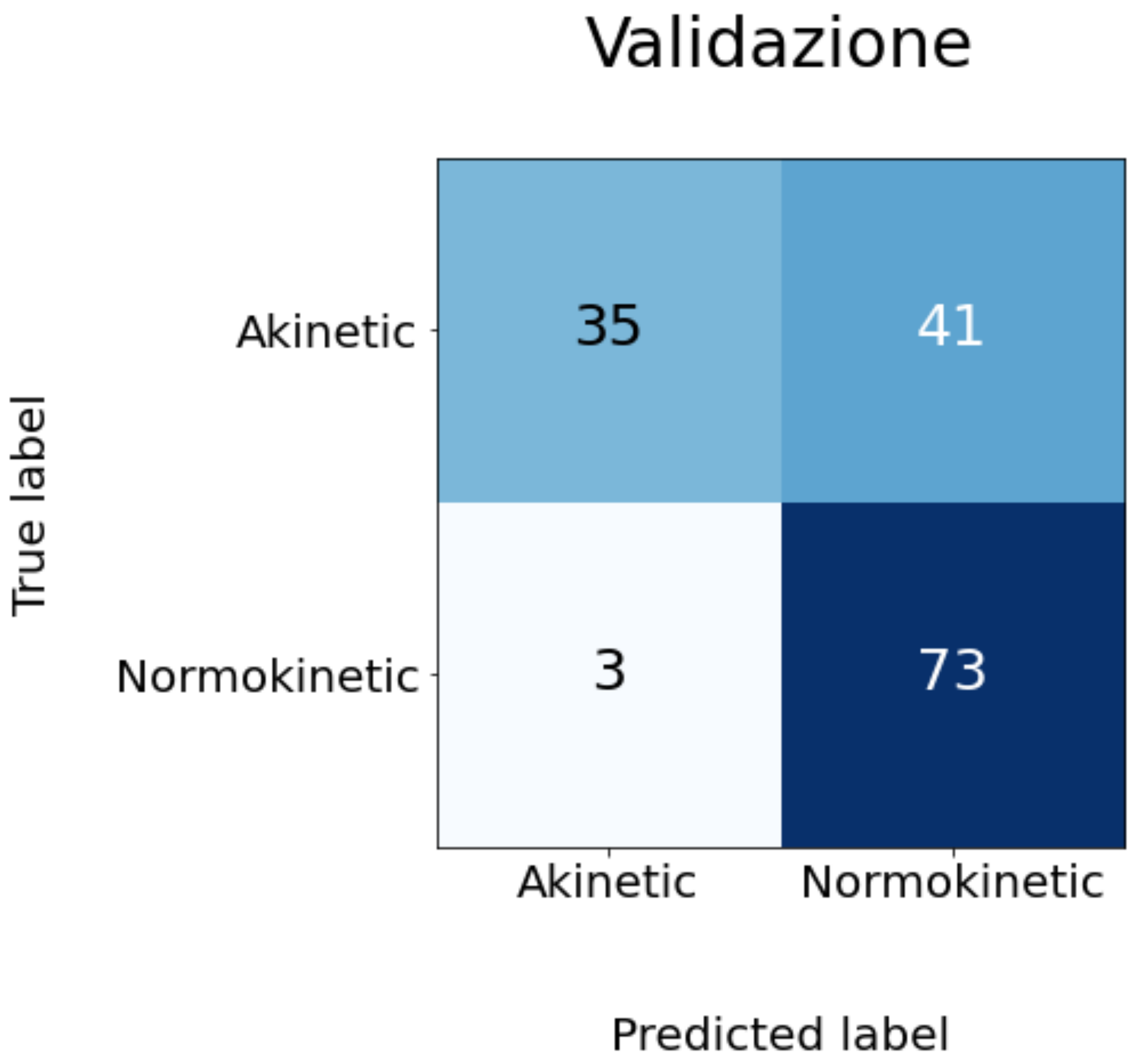
Dimasdia — miocardic akinesia

Model Training



Dimasdia — miocardic akinesia

Model Validation



Another COVID-19 detector

Voice and speech as predictors of COVID-19

Jing Han, Chloë Brown*, Jagmohan Chauhan*, Andreas Grammenos*, Apinan Hasthanasombat*, Dimitris Spathis*, Tong Xia*, Pietro Cicuta, and Cecilia Mascolo.

"**Exploring Automatic COVID-19 Diagnosis via Voice and Symptoms from Crowdsourced Data**."

arXiv preprint:2102.05225 (2021).

To appear at the proceedings of IEEE ICASSP 2021.

EXPLORING AUTOMATIC COVID-19 DIAGNOSIS VIA VOICE AND SYMPTOMS FROM CROWDSOURCED DATA

Jing Han, Chloë Brown, Jagmohan Chauhan*, Andreas Grammenos* Apinan Hasthanasombat*, Dimitris Spathis*, Tong Xia*, Pietro Cicuta, Cecilia Mascolo*

University of Cambridge
jh2298@cam.ac.uk

ABSTRACT

The development of fast and accurate screening tools, which could facilitate testing and prevent more costly clinical tests, is key to the current pandemic of COVID-19. In this context, some initial work shows promise in detecting diagnostic signals of COVID-19 from audio sounds. In this paper, we propose a voice-based framework to automatically detect individuals who have tested positive for COVID-19. We evaluate the performance of the proposed framework on a subset of data crowdsourced from our app, containing 828 samples from 343 participants. By combining voice signals and reported symptoms, an AUC of 0.79 has been attained, with a sensitivity of 0.68 and a specificity of 0.82. We hope that this study opens the door to rapid, low-cost, and convenient pre-screening tools to automatically detect the disease.

Index Terms— COVID-19, Crowdsourced data, Speech analysis, Symptoms analysis

1. INTRODUCTION

On 11 March 2020, the World Health Organisation announced the COVID-19 outbreak as a global pandemic. At the time of writing this paper, more than 37 million confirmed COVID-19 cases and one million deaths globally have been reported. Nowadays, in addition to developing drugs and vaccines for treatment and protection [1, 2], scientists and researchers are also investigating primary screening tools that ideally should be accurate, cost-effective, rapid, and meanwhile easily accessible to the mass at large.

Amongst the efforts towards rapid screening [3, 4], audio-based diagnosis appears promising, mainly due to its non-invasive and ubiquitous character, which would allow for individual pre-screening 'anywhere', 'anytime', in real-time, and available to 'anyone' [5]. Many applications have been developed for monitoring health and wellbeing in recent times via intelligent speech and sound analysis [6, 7, 8].

In this paper, we propose machine learning models for voice-based COVID-19 diagnosis. More specifically, we analyse a subset of data from 343 participants crowdsourced via our app, and show the discriminatory power of voice for the diagnosis. We demonstrate how voice can be used as signal to distinguish symptomatic positive tested individuals, from non-COVID-19 (tested) individuals, who also have developed symptoms akin to COVID-19. We further show performance improvement by combining sounds and symptoms for the diagnosis, yielding a specificity of 0.82 and an AUC of 0.79.

2. RELATED WORK

With the advent of COVID-19, researchers have started to explore if respiratory sounds could be diagnostic [5]. For instance, in [4], breathing and cough sounds have been targeted and researchers demonstrate that COVID-19 individuals are distinguishable from healthy controls as well as asthmatic patients. In [13], an interpretable COVID-19 diagnosis framework has been devised to distinguish COVID-19 cough from other types of cough. Likewise, in [12], a detectable COVID-19 signature has been found from cough sounds and can help increase the testing capacity.

However, none of the aforementioned efforts have analysed the potential of voice. Recently, the feasibility for COVID19 screening using voice has been introduced in [14]. Similarly, in [15], significant differences in several voice characteristics are observed between COVID-19 patients and healthy controls. Moreover, in [16], speech recordings from hospitalised COVID-19 patients are analysed to categorise their health state of patients. Our work differs from these works, as we utilise an entirely crowdsourced dataset, for which we have to deal with the complexity of the data such as recordings in different languages and varied environmental noises. Furthermore,

Predicting 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

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

Image Memorability — A business 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

HVAC cleanliness

Alisea — Transfer learning example

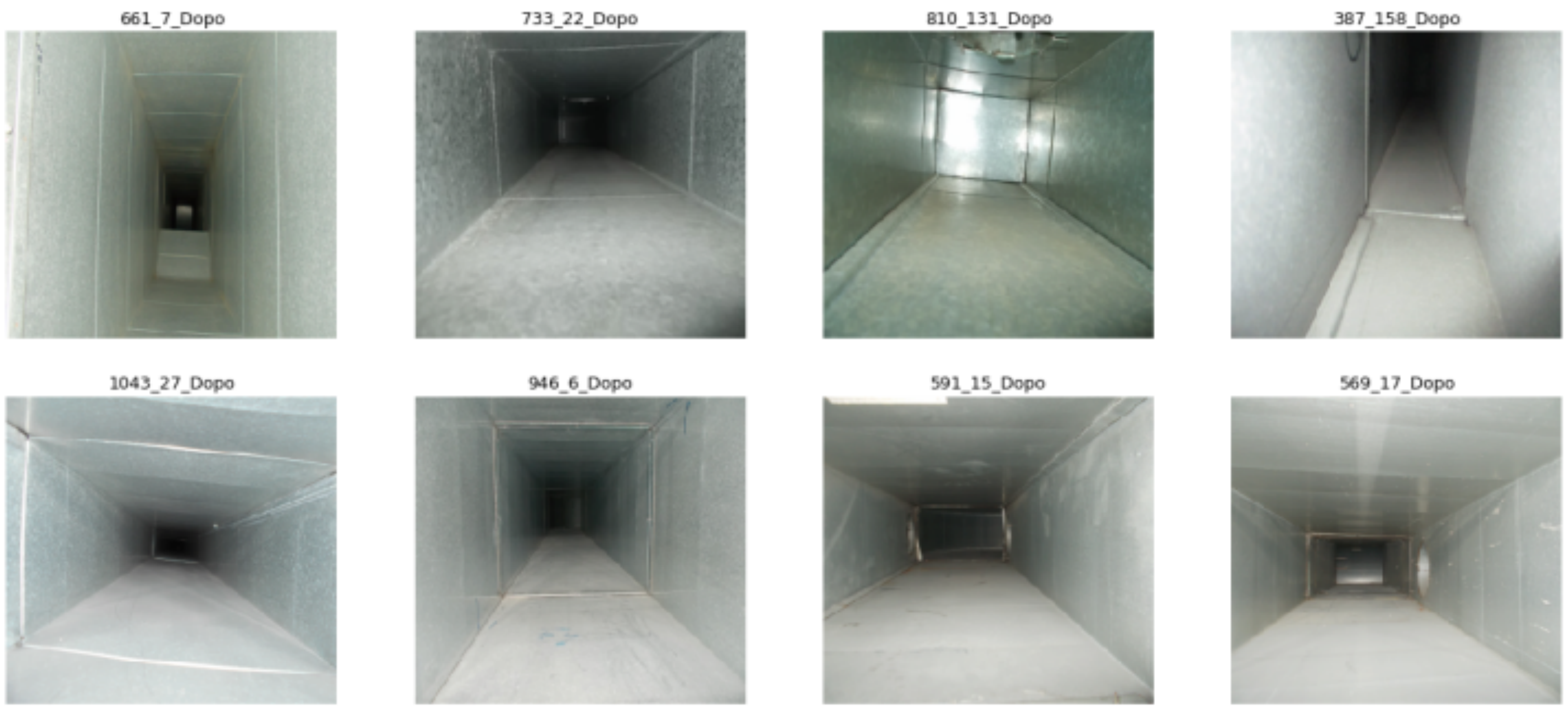
PROBLEM: Classify images of air duct/pipes as ‘dirty’ or ‘clean’

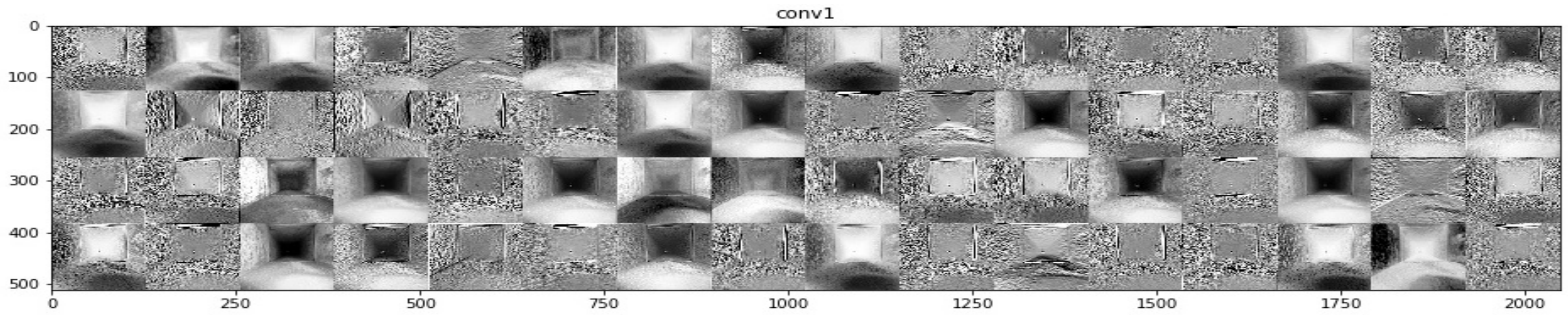
- Dataset composed of hundreds of images of different air pipes, taken with different cameras, in different sizes.
- Balanced dataset: 50% labelled ‘dirty’, 50% labelled ‘clean’.
- RGB color channel

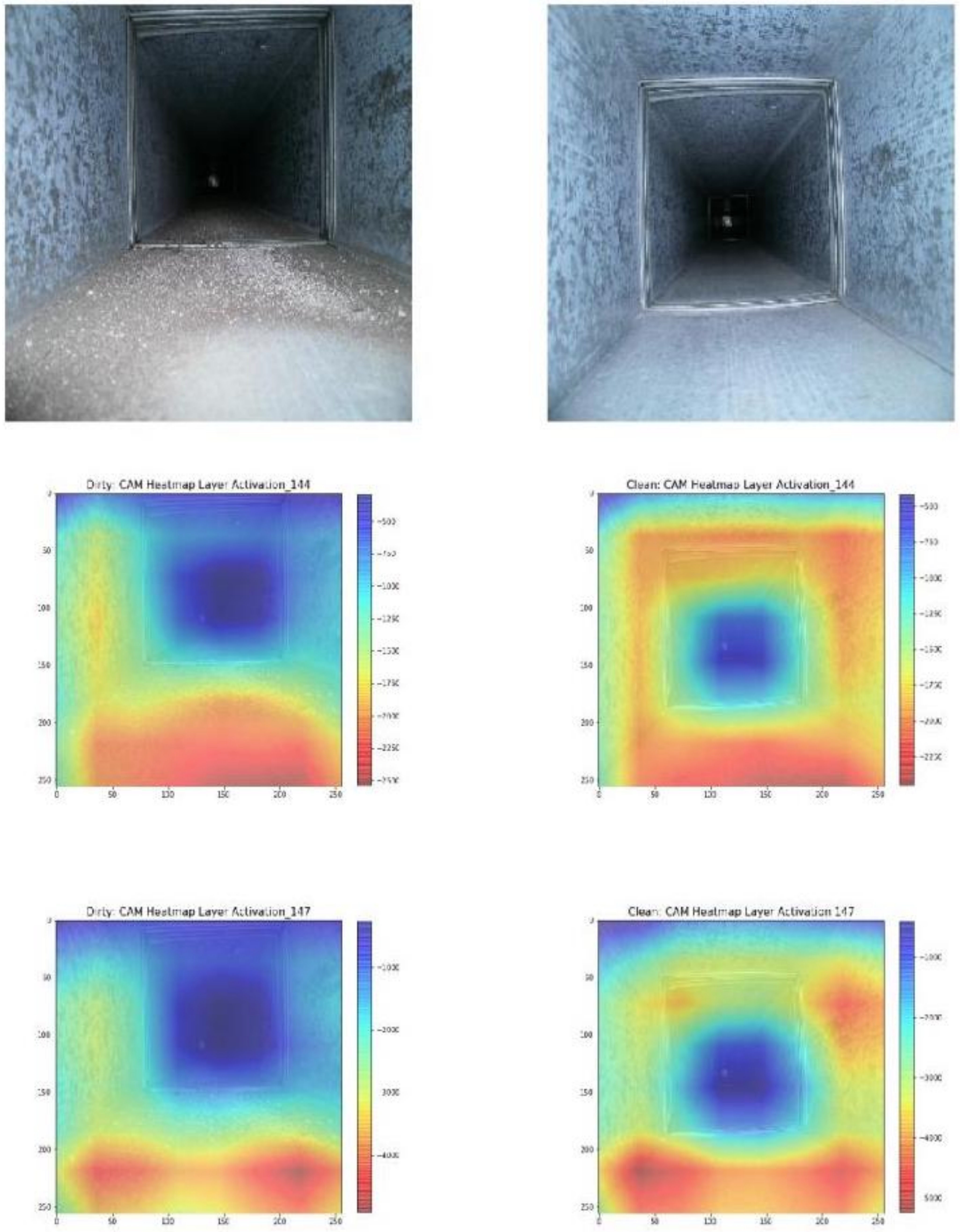
Which images size to use? Which color channels?

What does the model see?

False negative images (truth: clean, predicted: dirty)







Neosperience
People Analytics

Why count people in store?

People number is a KPI used to estimate ROI

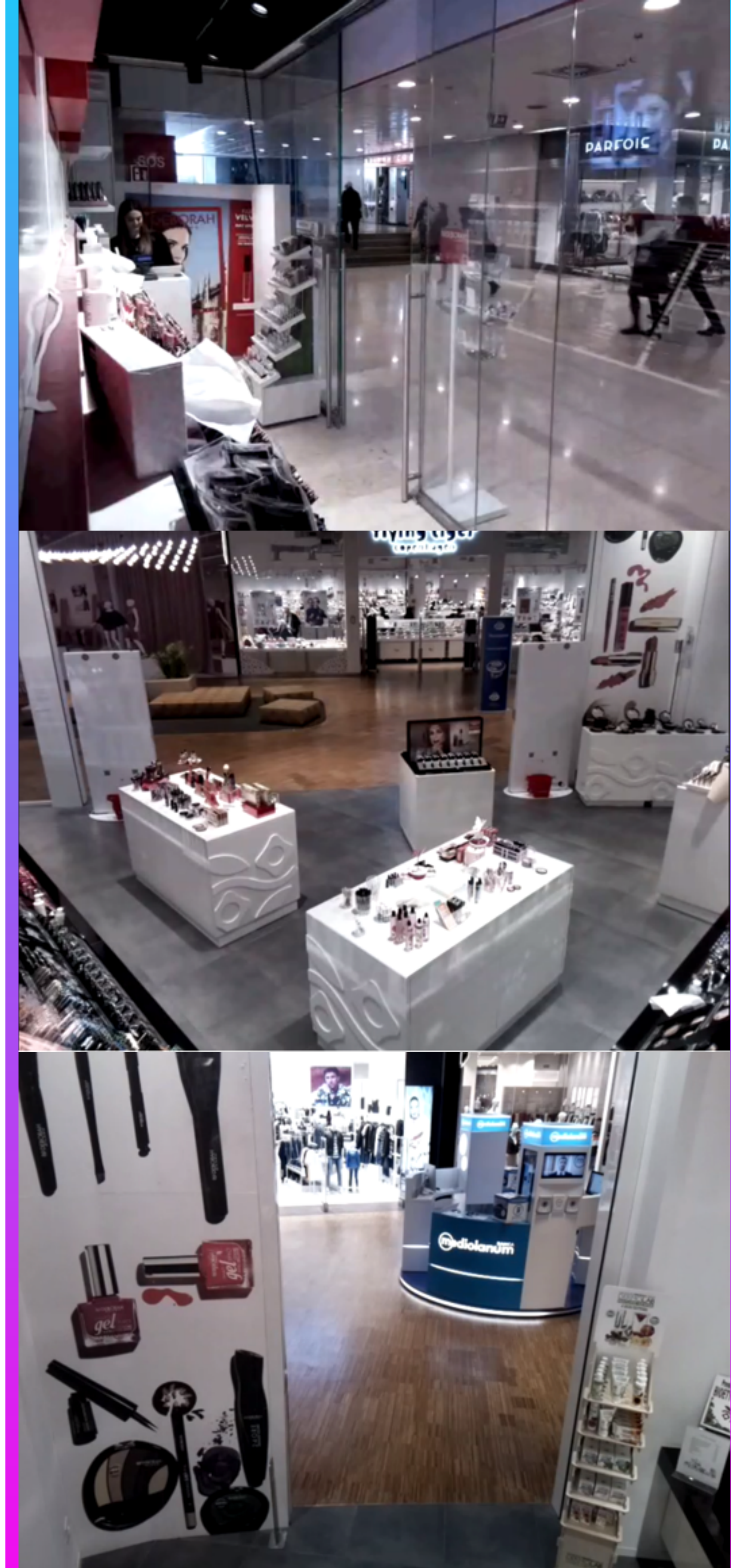
- Understanding the number of people is considered a good way to estimate the average return of a given store
- The daily income of a store divided by the overall number of people detected gives a ROI
- Understanding high traffic stores can led to strategic decisions
- Low traffic or lower ROI can be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

Introducing Neosperience People 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

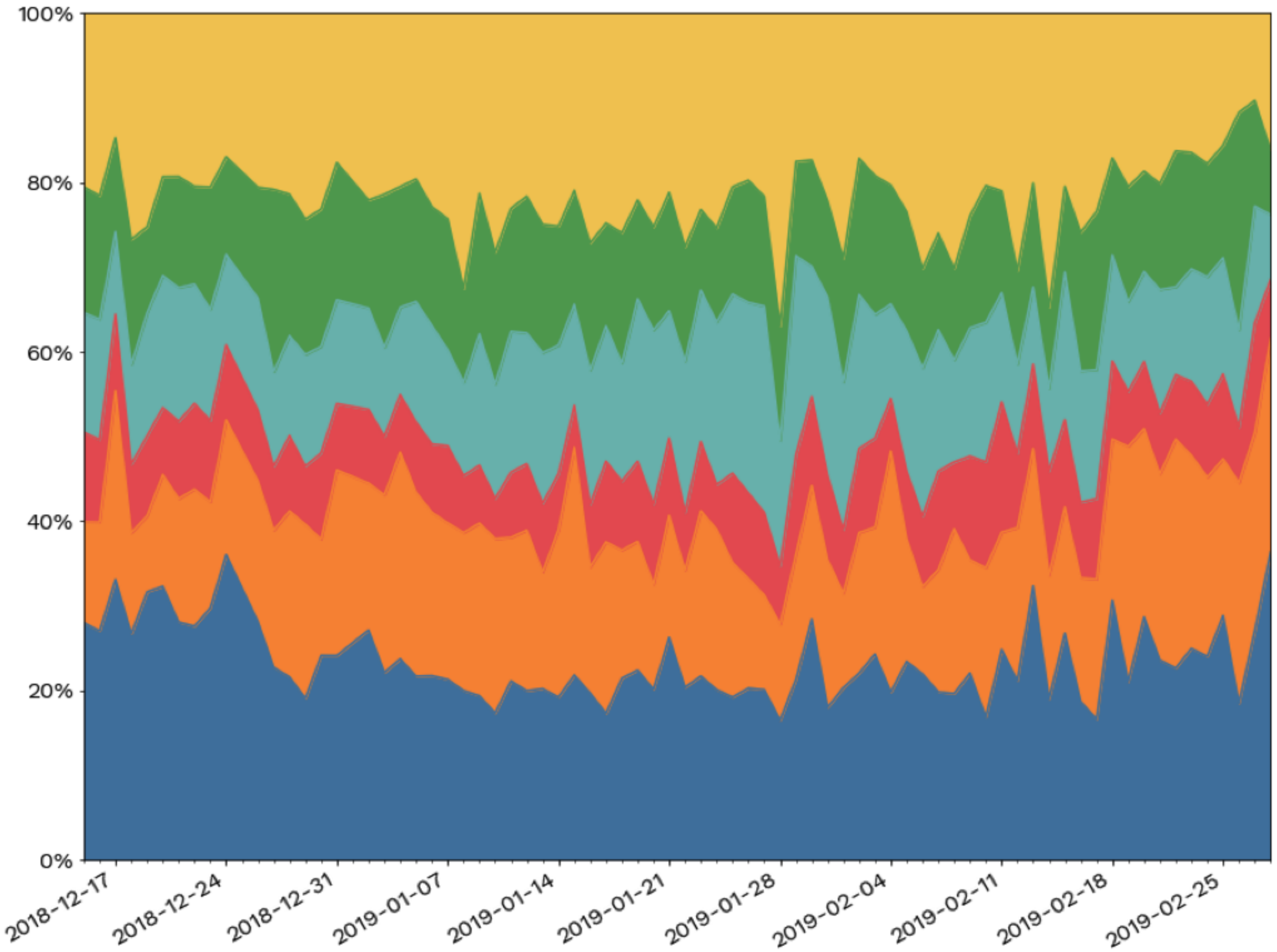
Mapping people presence within a given area of interest

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

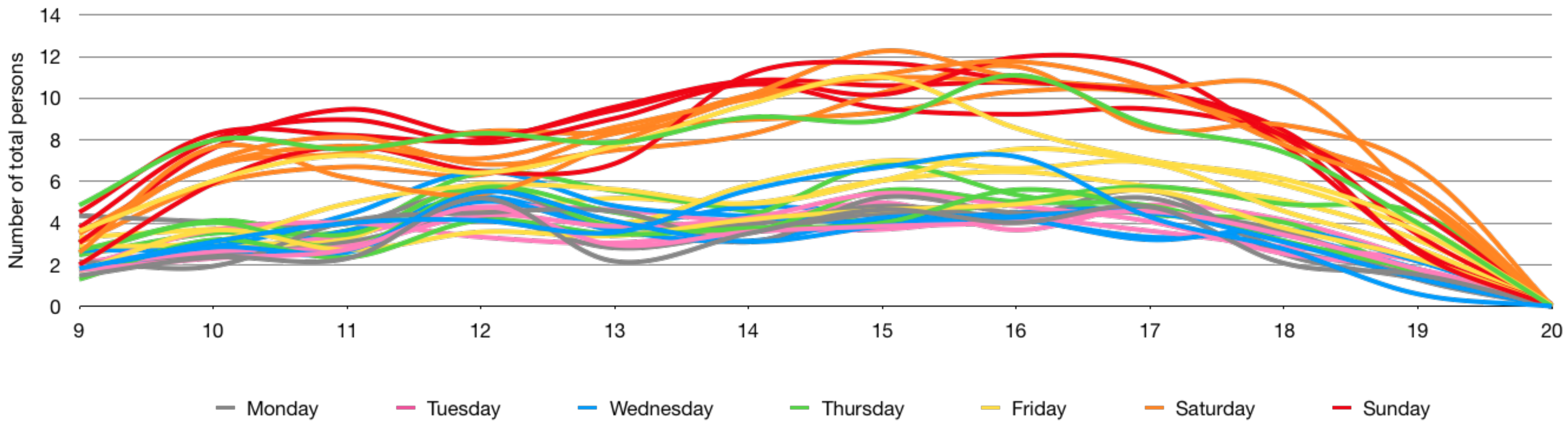
Outcome

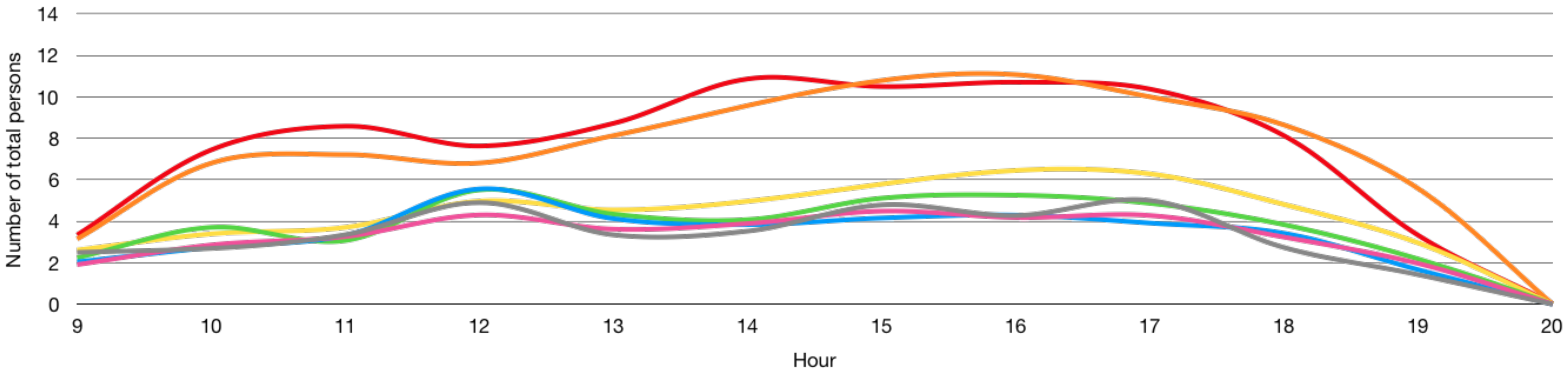
Store area clustering and occupancy timeline

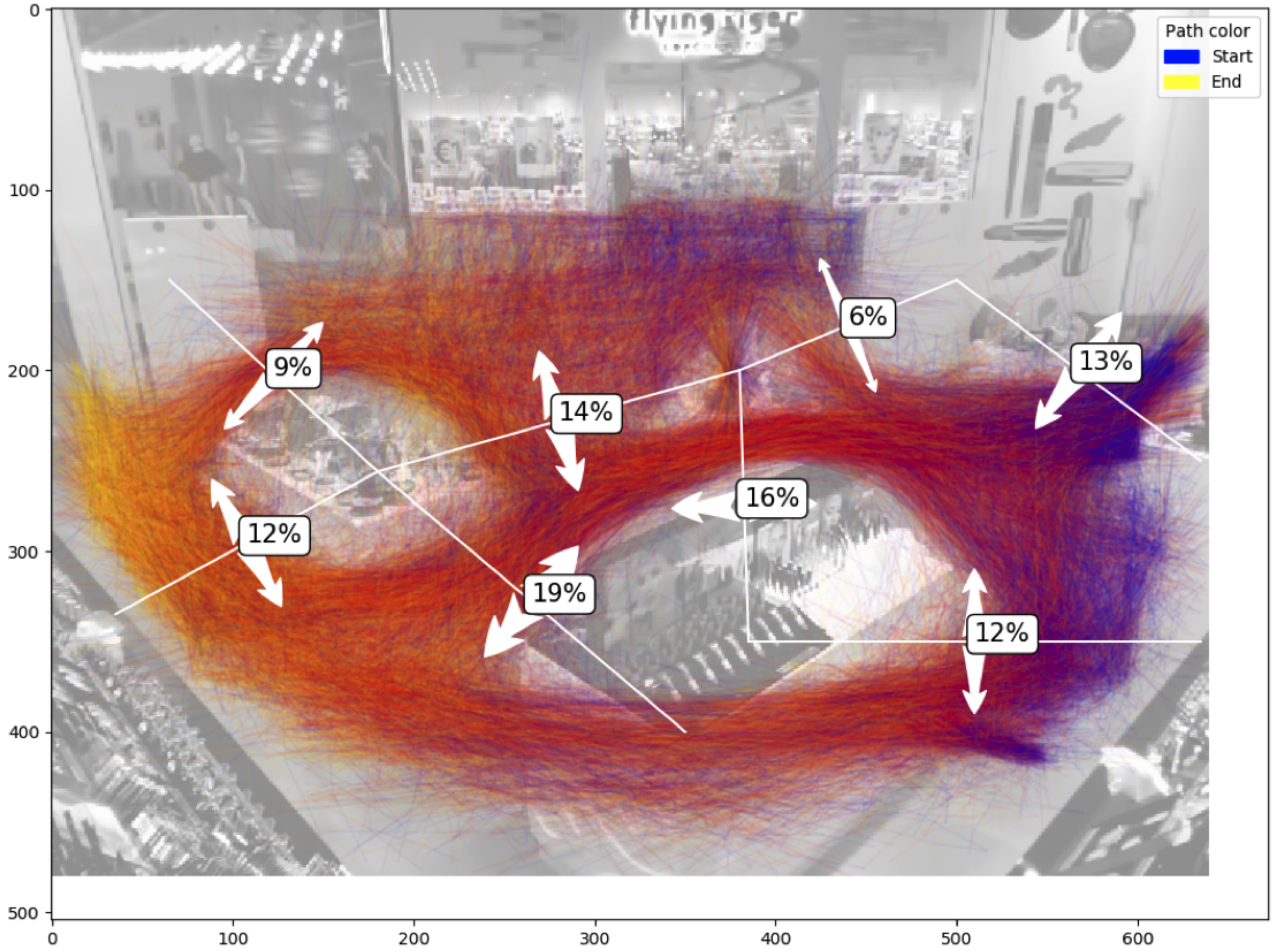


Outcome

Trajectories







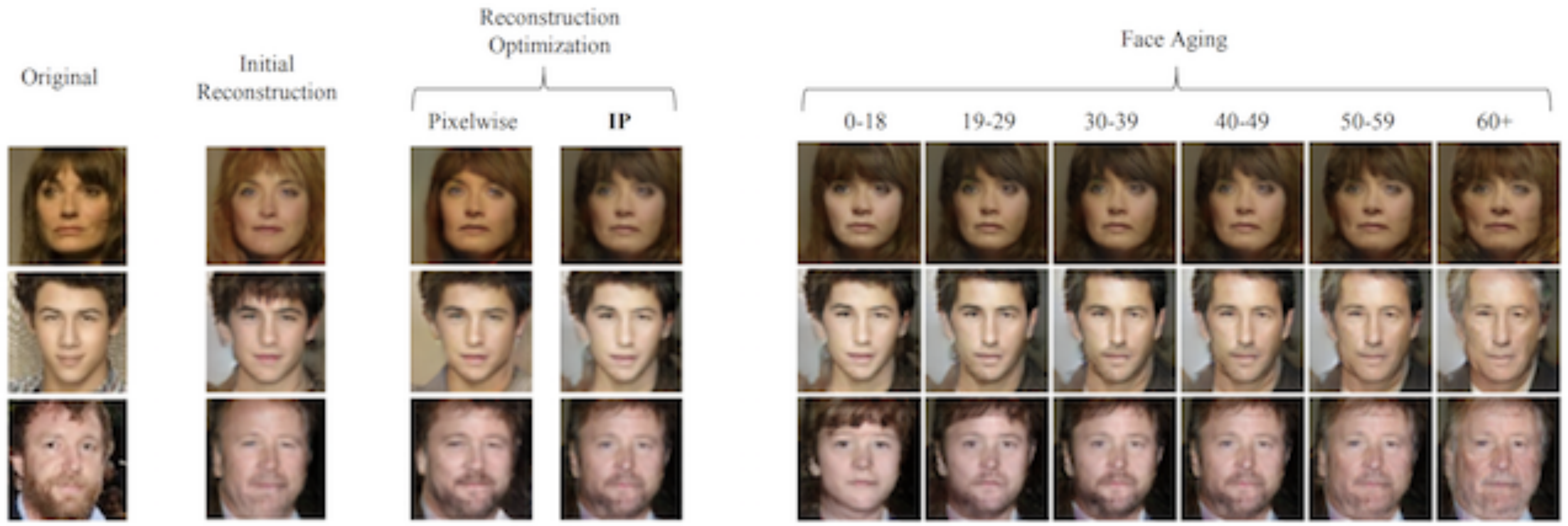
dataset generation

GAN to generate artificial images

Use GANs to generate images and augment (or balance) a dataset

GAN can be used to simulate face ageing 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

Amazon DeepRacer

Use Reinforcement Learning to win an autonomous driving car 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.

OpenAI

Jukebox

We're introducing Jukebox, a neural net that generates music, including rudimentary singing, as raw audio in a variety of genres and artist styles. We're releasing the model weights and code, along with a tool to explore the generated samples.

https://jukebox.openai.com/

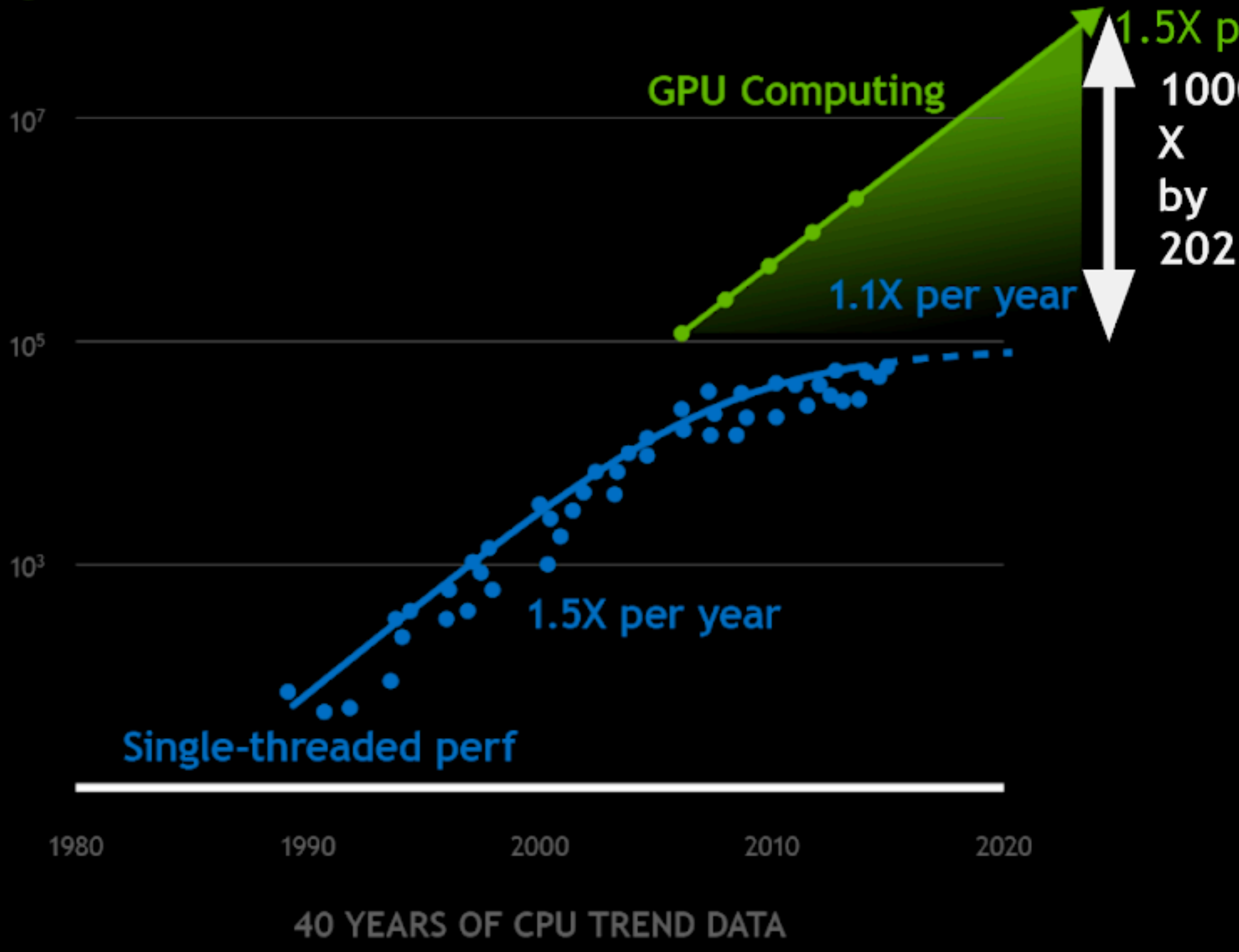
https://openai.com/blog/emergent-tool-use/

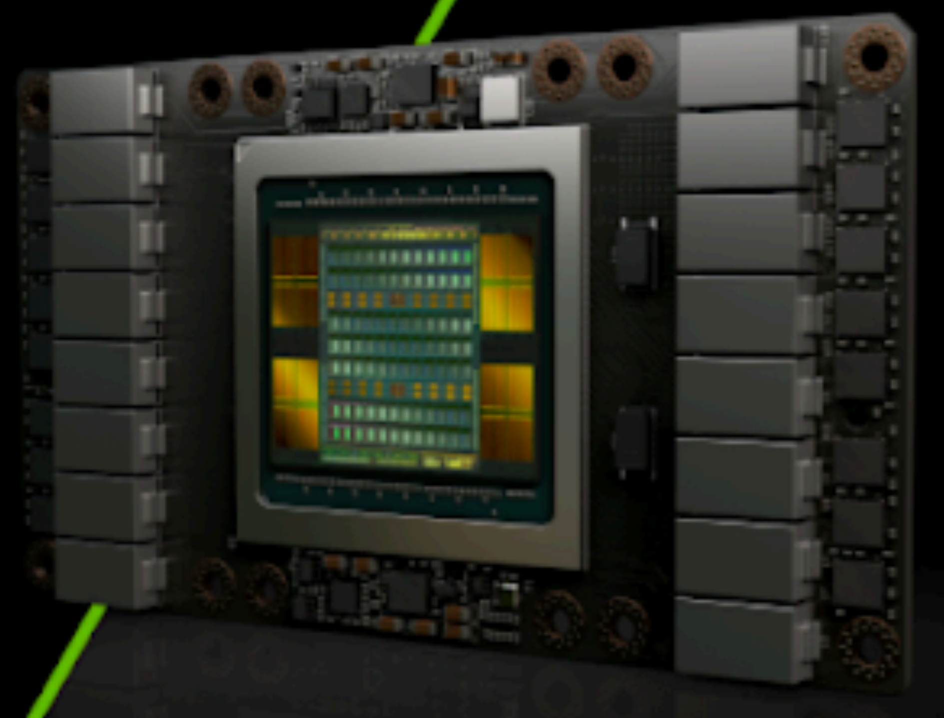
“the future is here,
but it’s not
evenly distributed”

computing power

THE TIME HAS COME FOR GPU COMPUTING

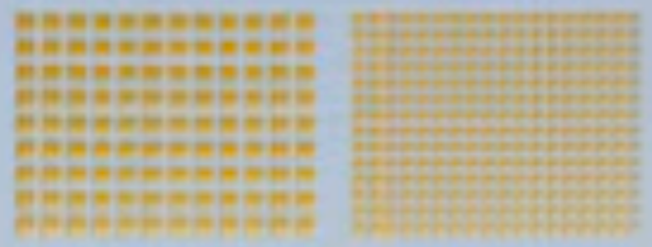
For 30 years, the dynamics of Moore's law held true. Microprocessor performance advanced at a rate
of 50 percent per year as more and more transistors were fit onto a single chip. But that approach is hitting the limits of semiconductor physics, and, today, CPU performance only grows by 10 percent per year. NVIDIA GPU computing has given the industry a path forward — and will provide a 1,000X speed-up by 2025.





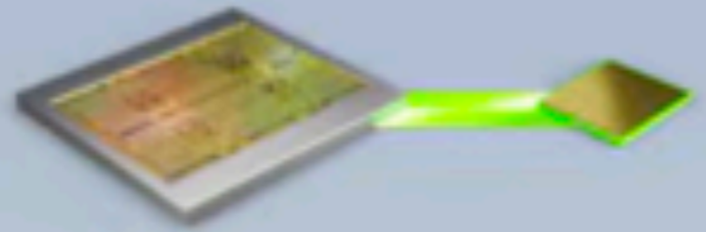
ANNOUNCING NVIDIA A100 80GB

Supercharging The World's Highest Performing AI Supercomputing GPU



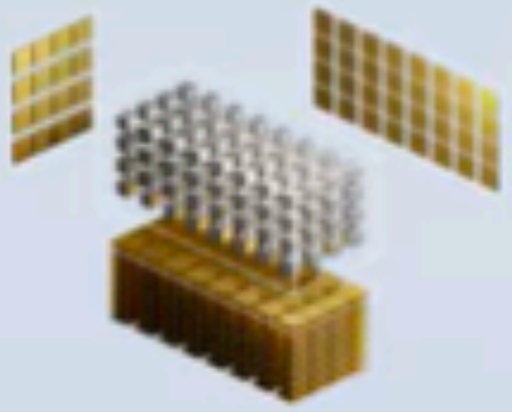
80GB HBM2e

For largest datasets and models



2TB/s +

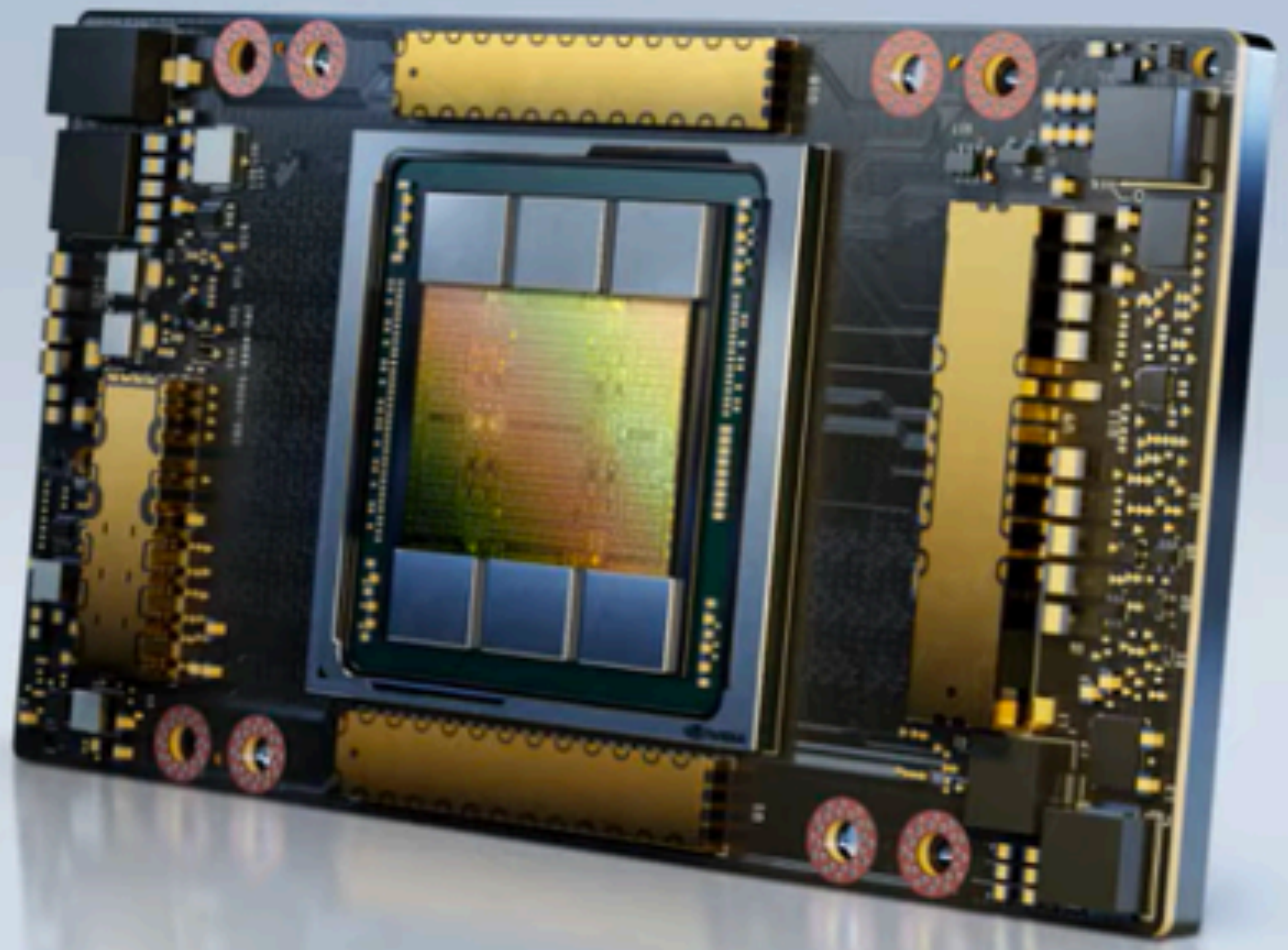
World's highest memory bandwidth
to feed the world's fastest GPU



3rd Gen Tensor Core

Multi-Instance GPU

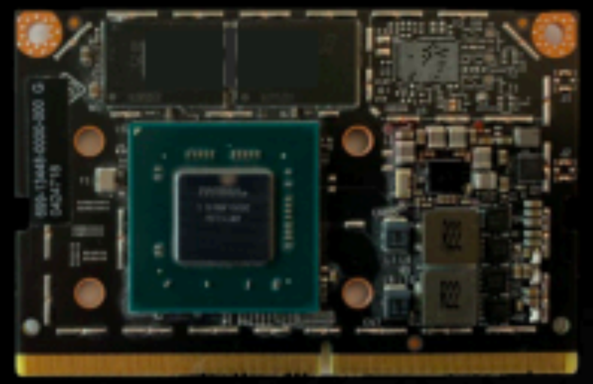
3rd Gen NVLink



NEW JETSON FAMILY

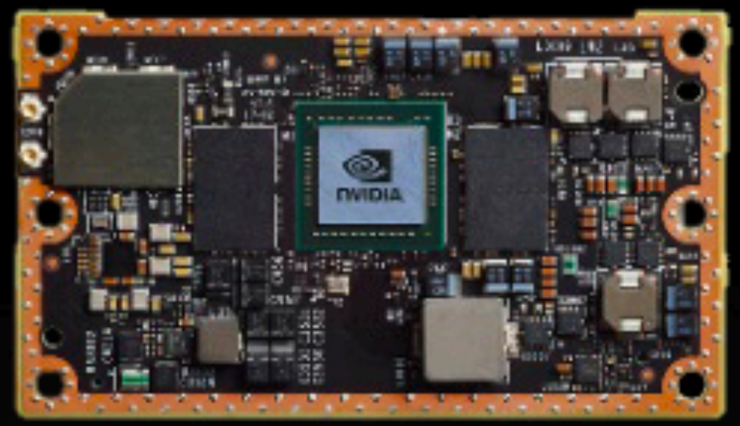
Top-to-Bottom Embedded AI Computer Lineup

JETSON NANO



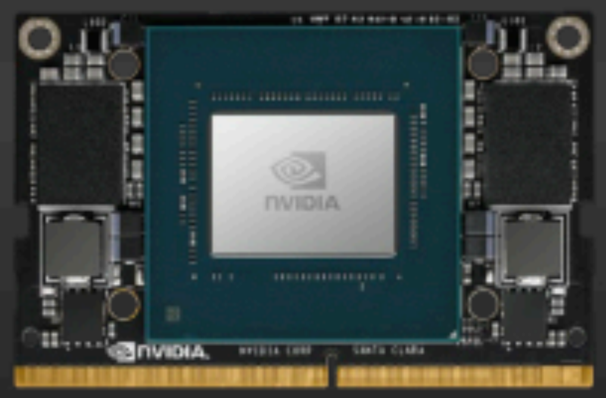
0.5 TFLOPS (FP16)
5-10 W
45 mm x 70 mm
$129

JETSON TX2 SERIES
(TX2, TX2 4GB, TX2i*)



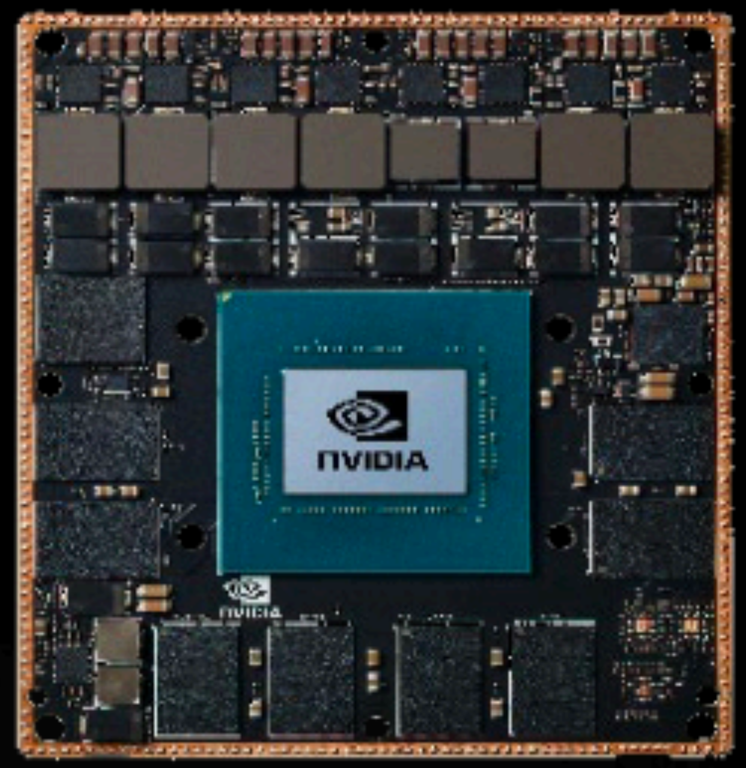
1.3 TFLOPS (FP16)
7.5-15 W*
50 mm x 87 mm
Starting at $249

JETSON XAVIER NX



6 TFLOPS (FP16) | 21 TOPS (INT8)
10-15 W
45 mm x 70 mm
$399

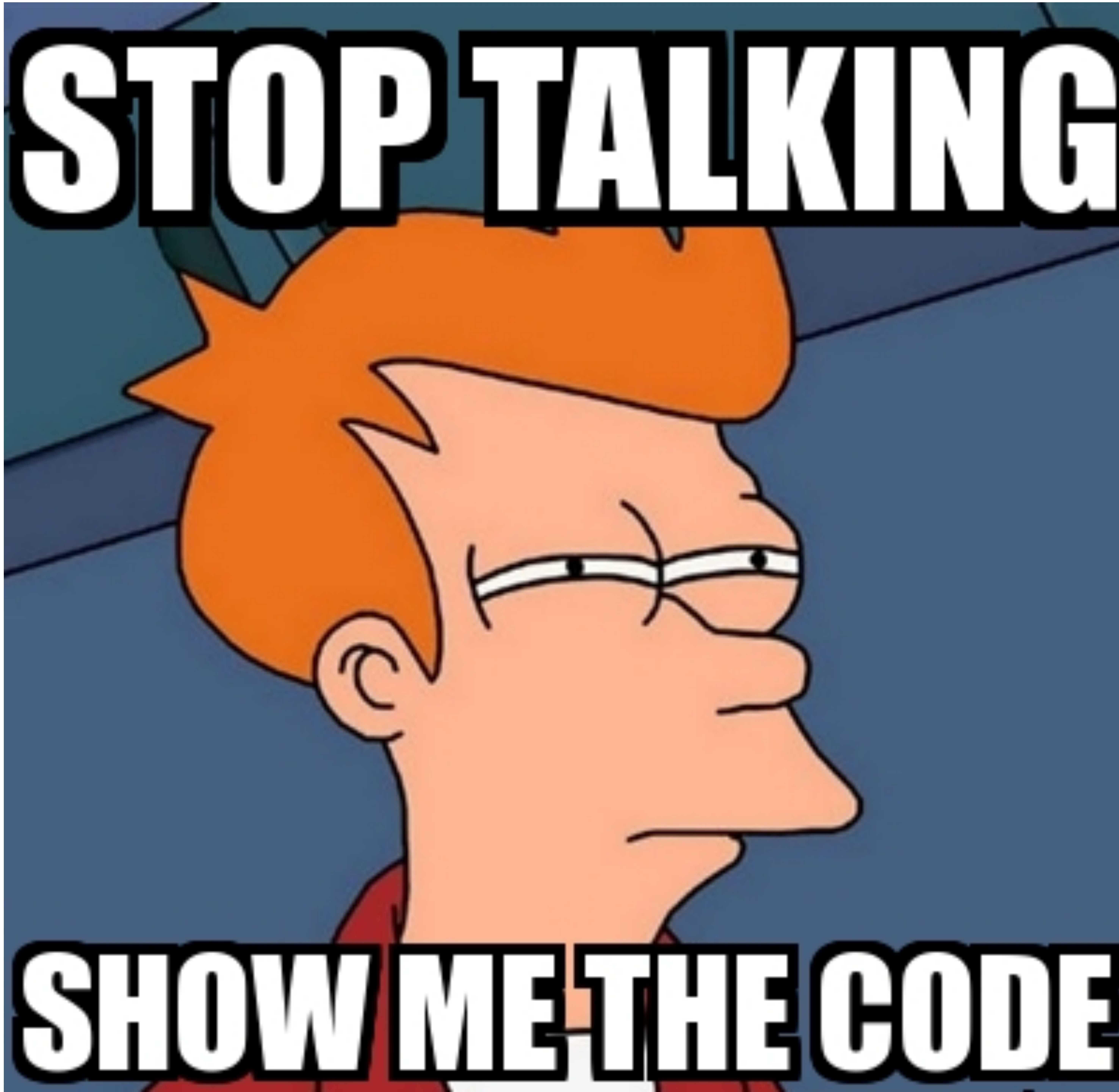
JETSON AGX XAVIER SERIES
(AGX Xavier 8GB, AGX Xavier)



20-32 TOPS (INT8)
5.5-11 TFLOPS (FP16)
10-30 W
100 mm x 87 mm
Starting at $599

One Software Architecture

Building a CNN



PyTorch + FastAI

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

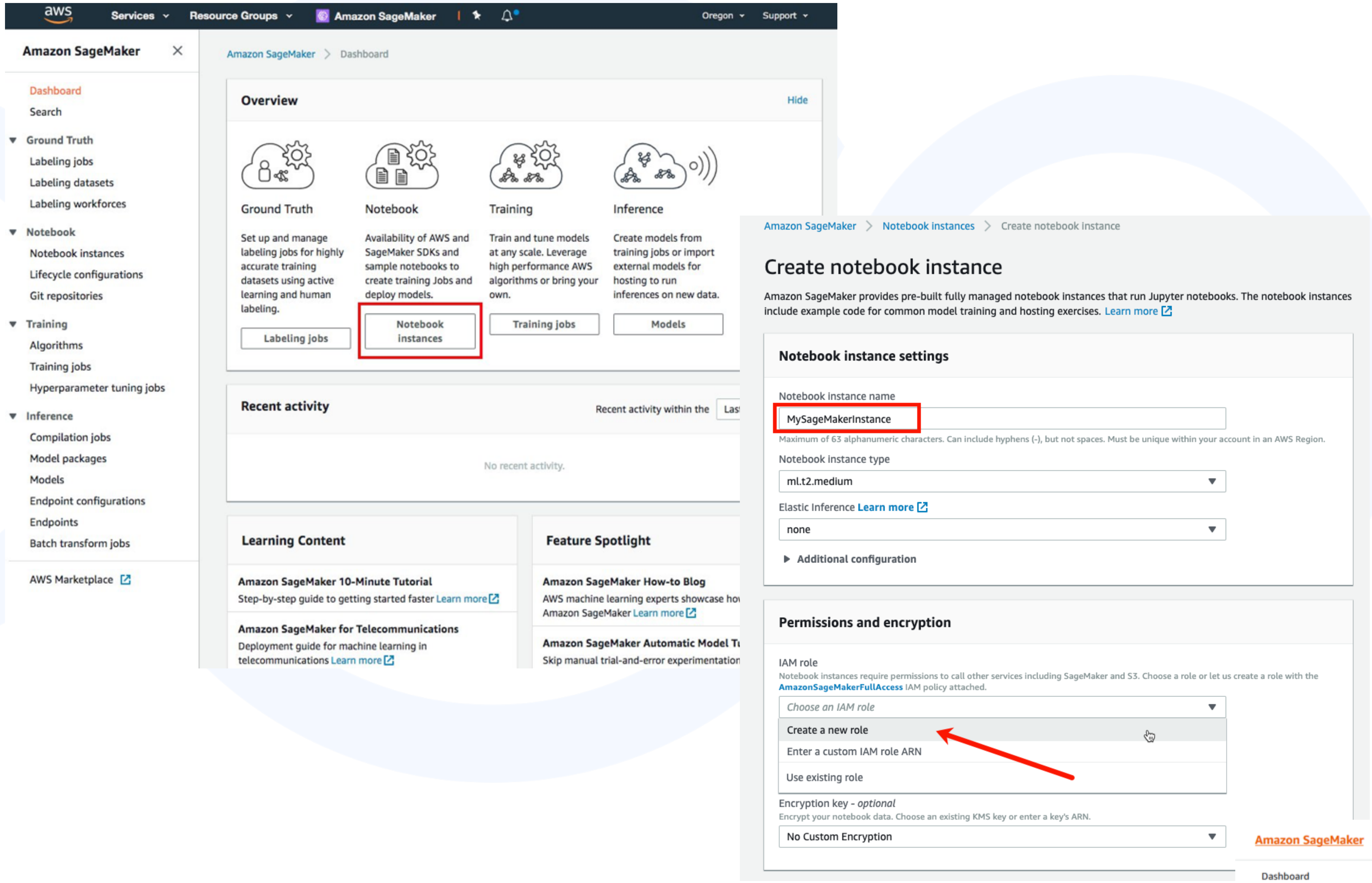
PyTorch

fast.ai

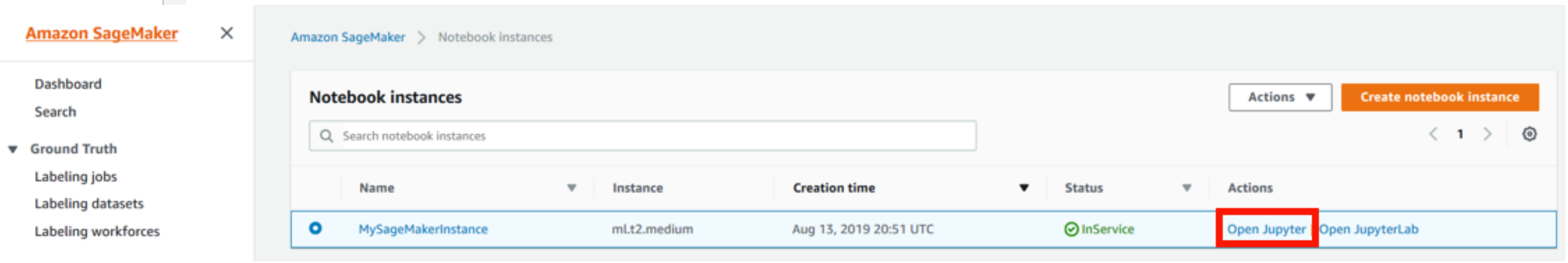
Making neural nets uncool again

Jupyter Notebooks: an IDE for Machine Learning

Amazon SageMaker -- with Deep Learning instances



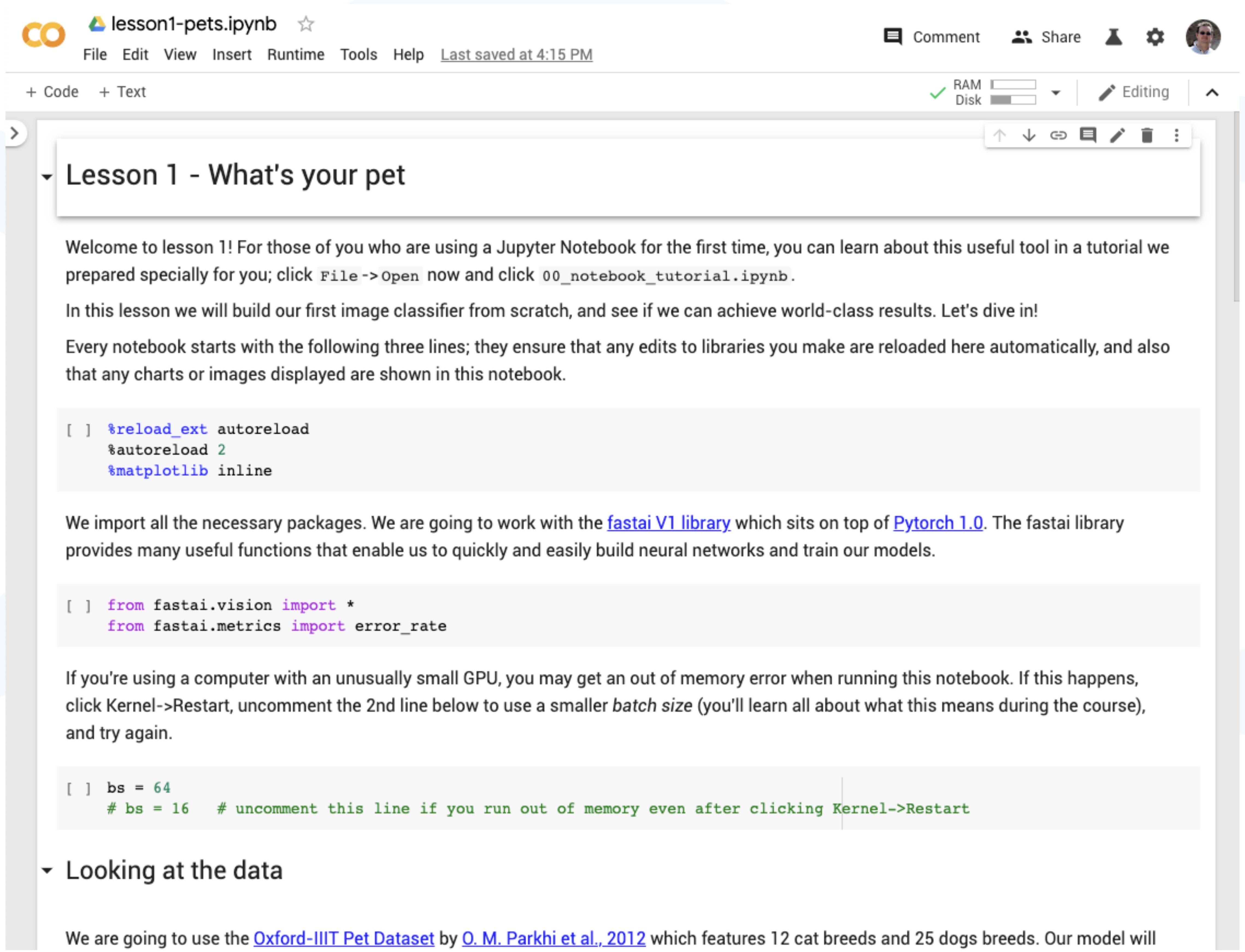
Amazon SageMaker offers fully managed Jupyter Notebook (Jupyter Lab) running on a virtual instances with on board GPUs to easily allow model training and model inference on a variety of trained models



Neosperience

Jupyter Notebooks: an IDE for Machine Learning

Google Colab -- free managed Notebooks

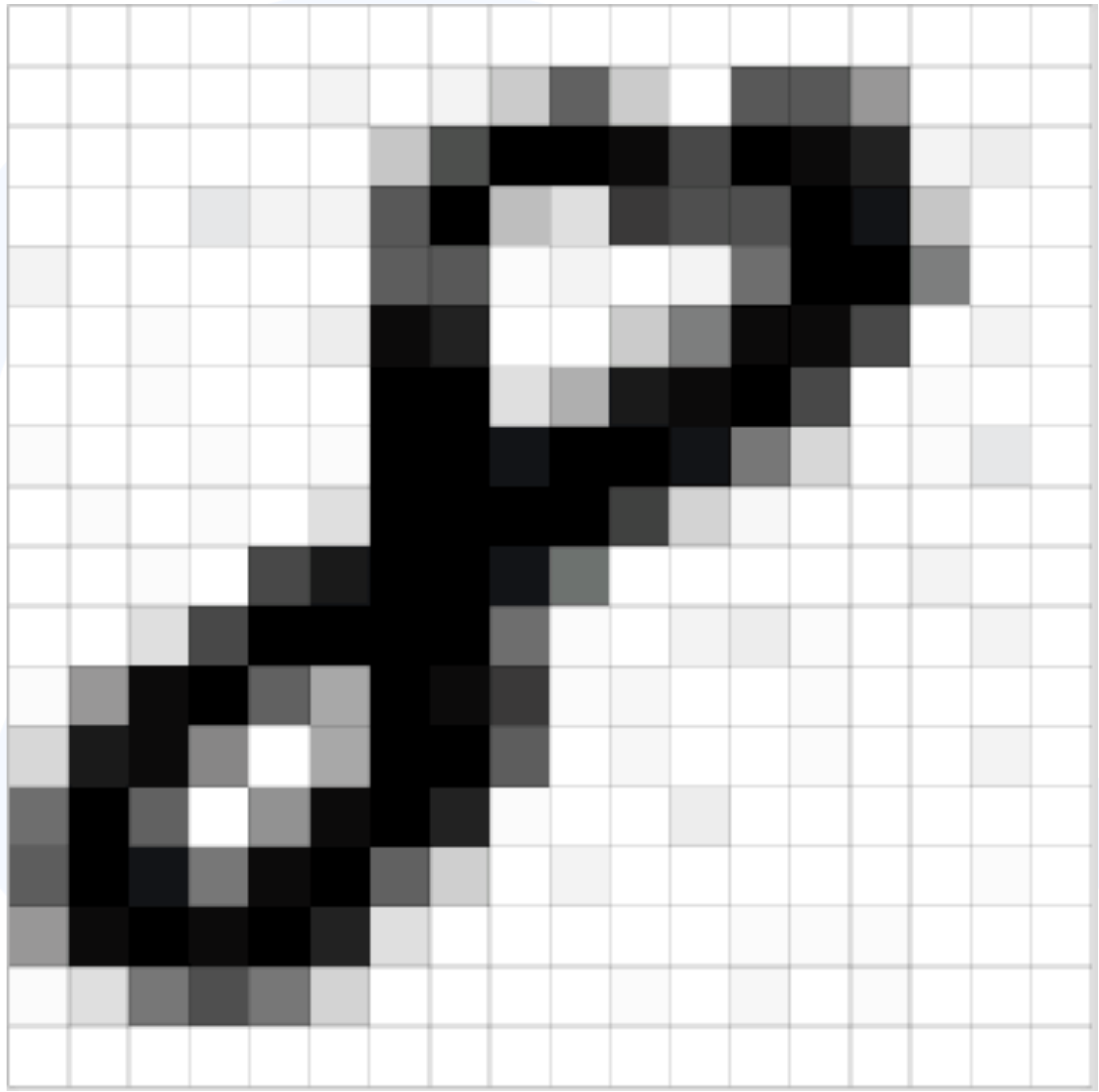


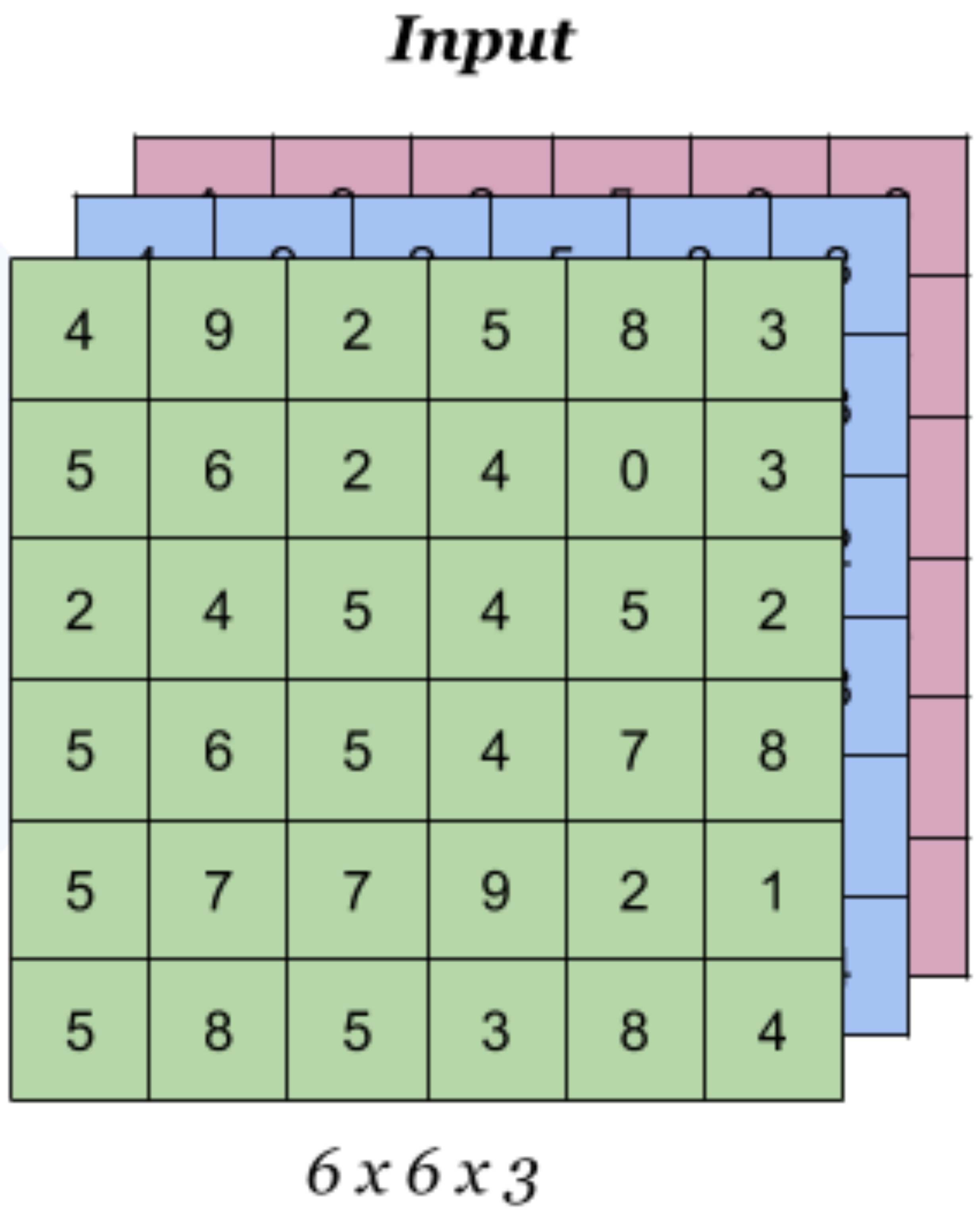
Google Colab is a completely free managed Jupyter Notebook instance, where can select runtime type between GPU and TPU.

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

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**

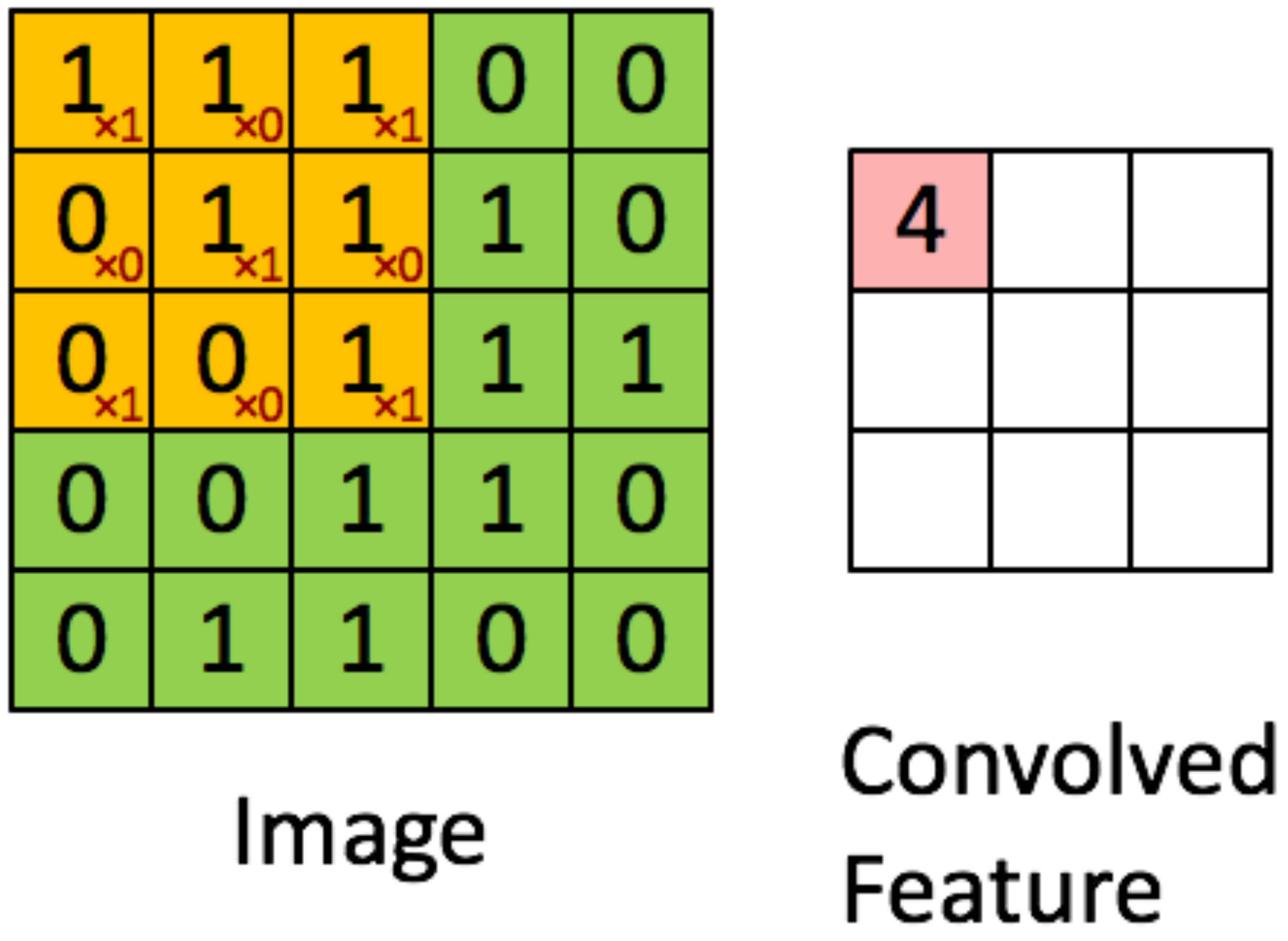
Input

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

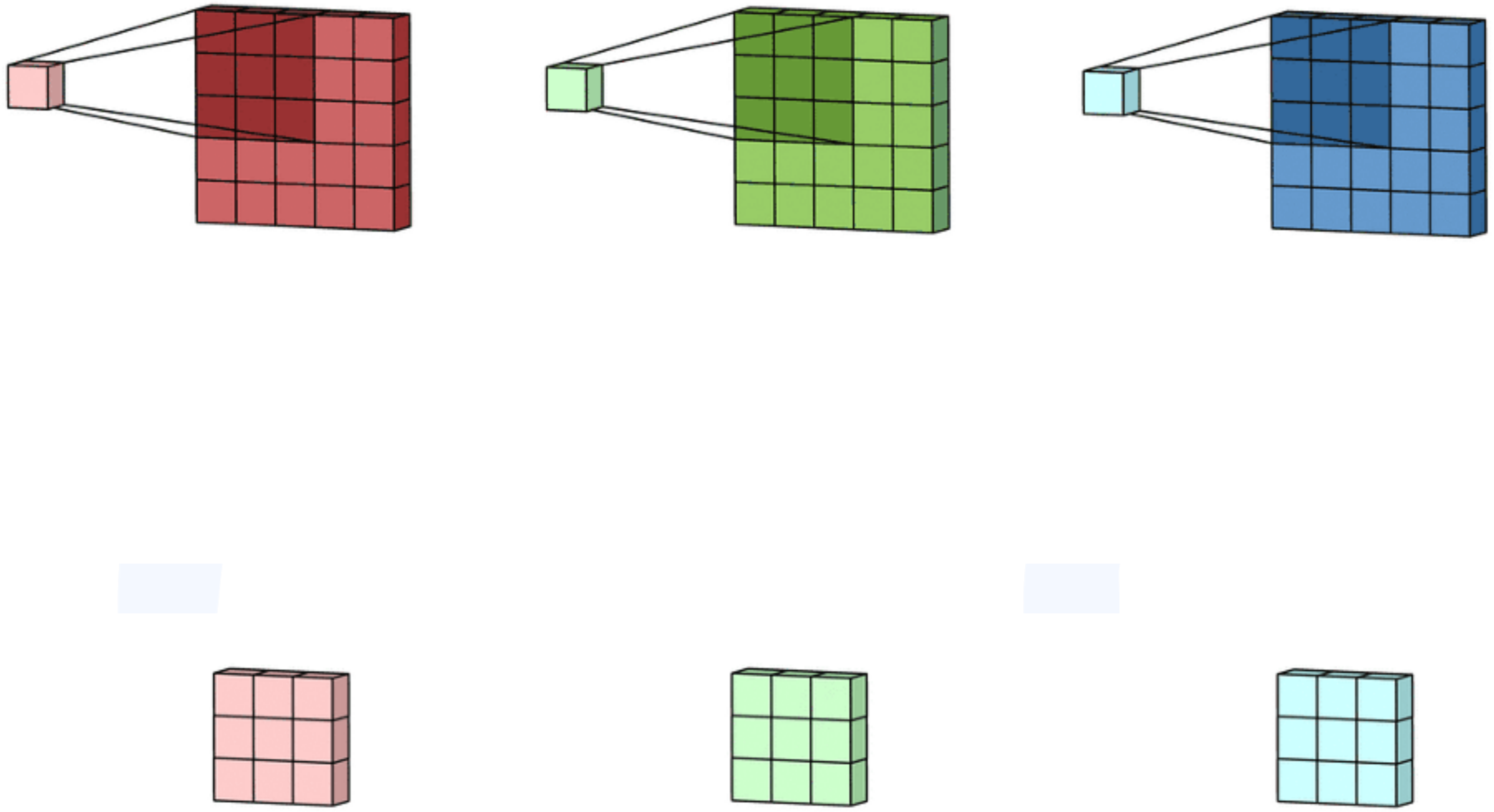
Filter

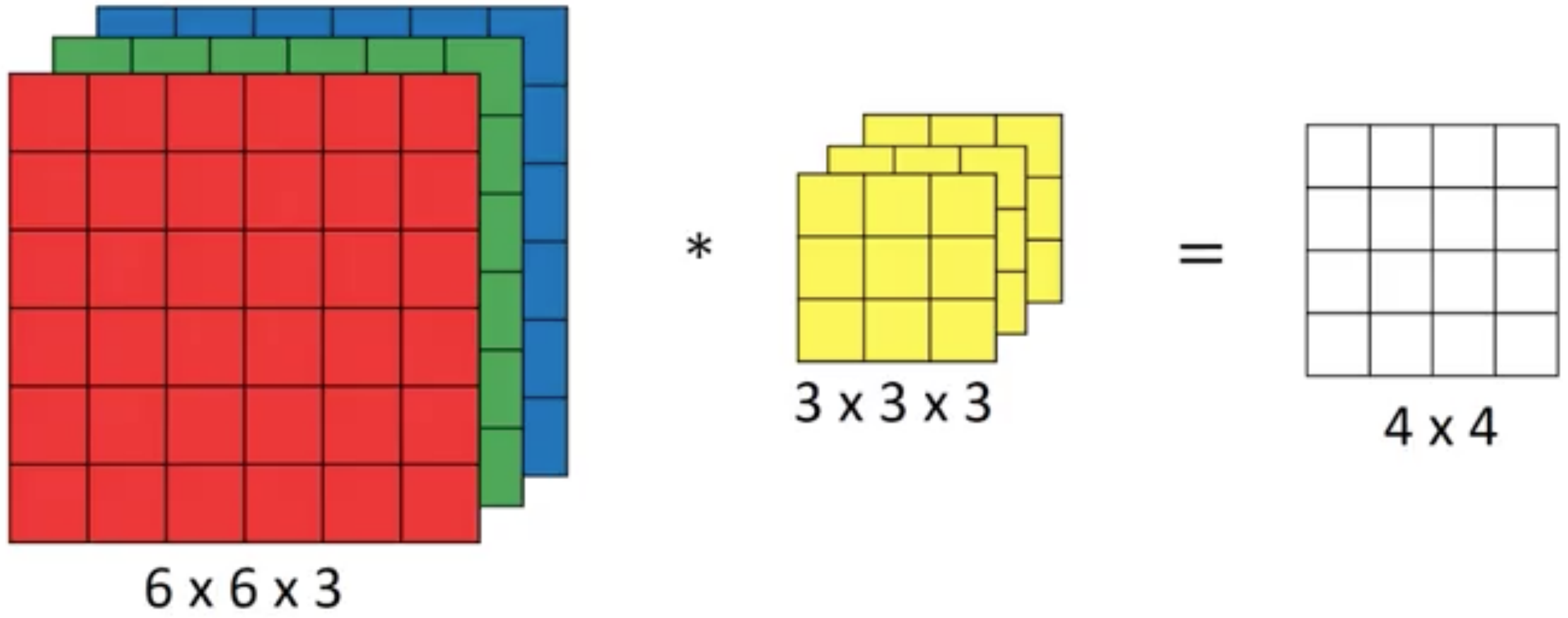
1	0	1
0	1	0
1	0	1

Convolution filter



Convolution filter - 3 channel example

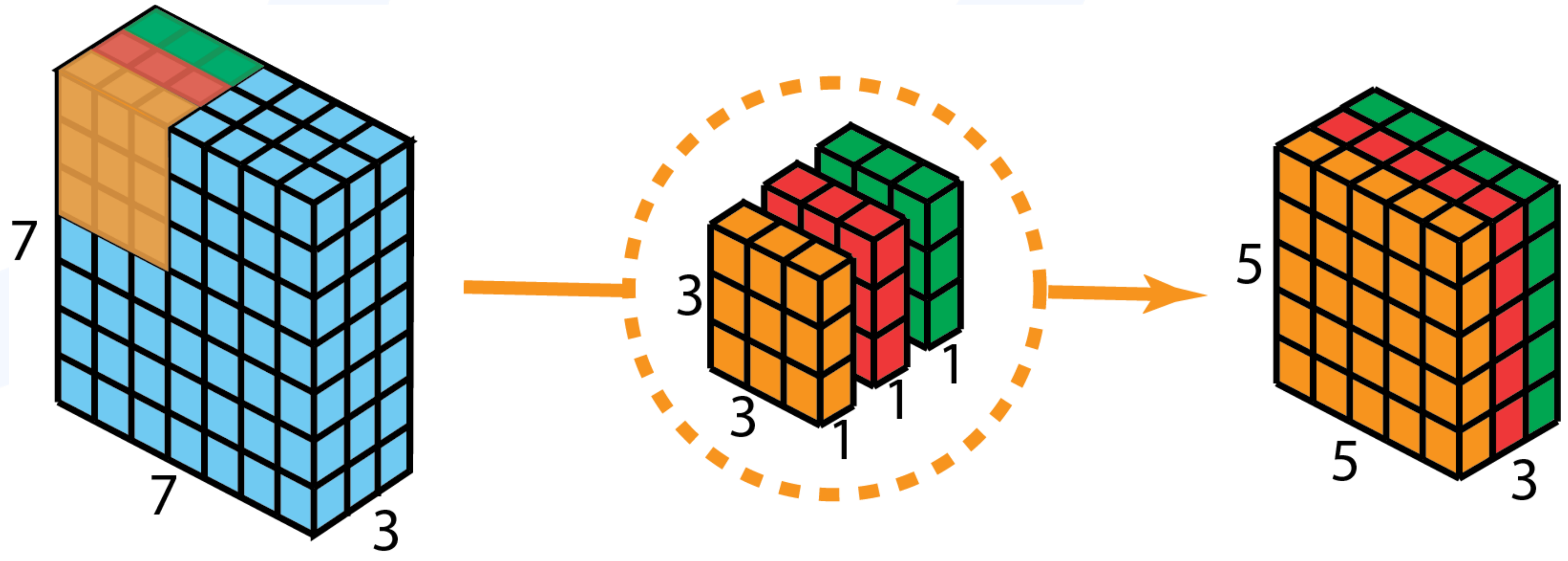




Convolution filter parameters

Each filter is characterized by the following parameters:

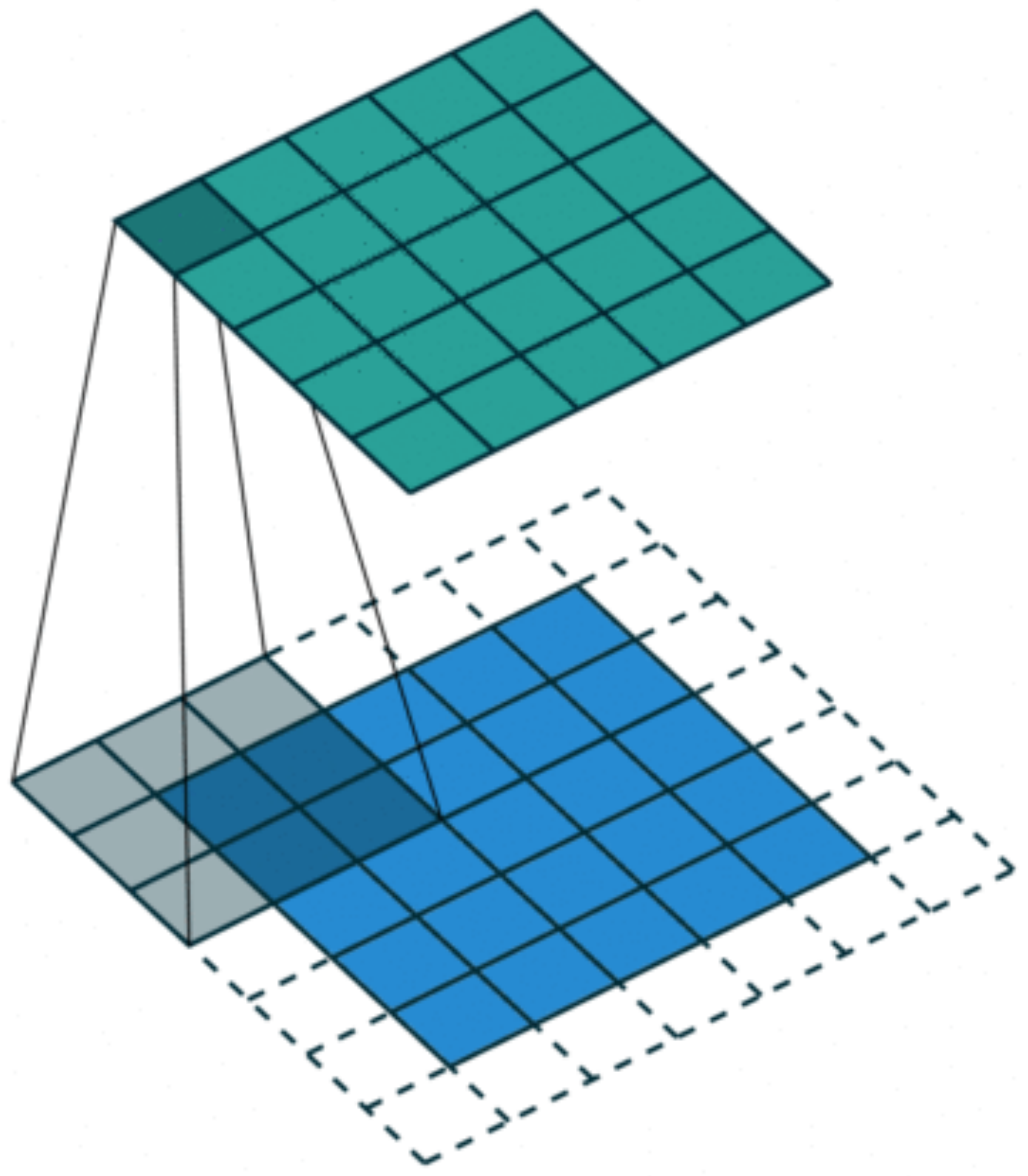
- **Depth:** number of distinct filters we use for the convolution operation. Multiple filters are used to detect different "features" of the images



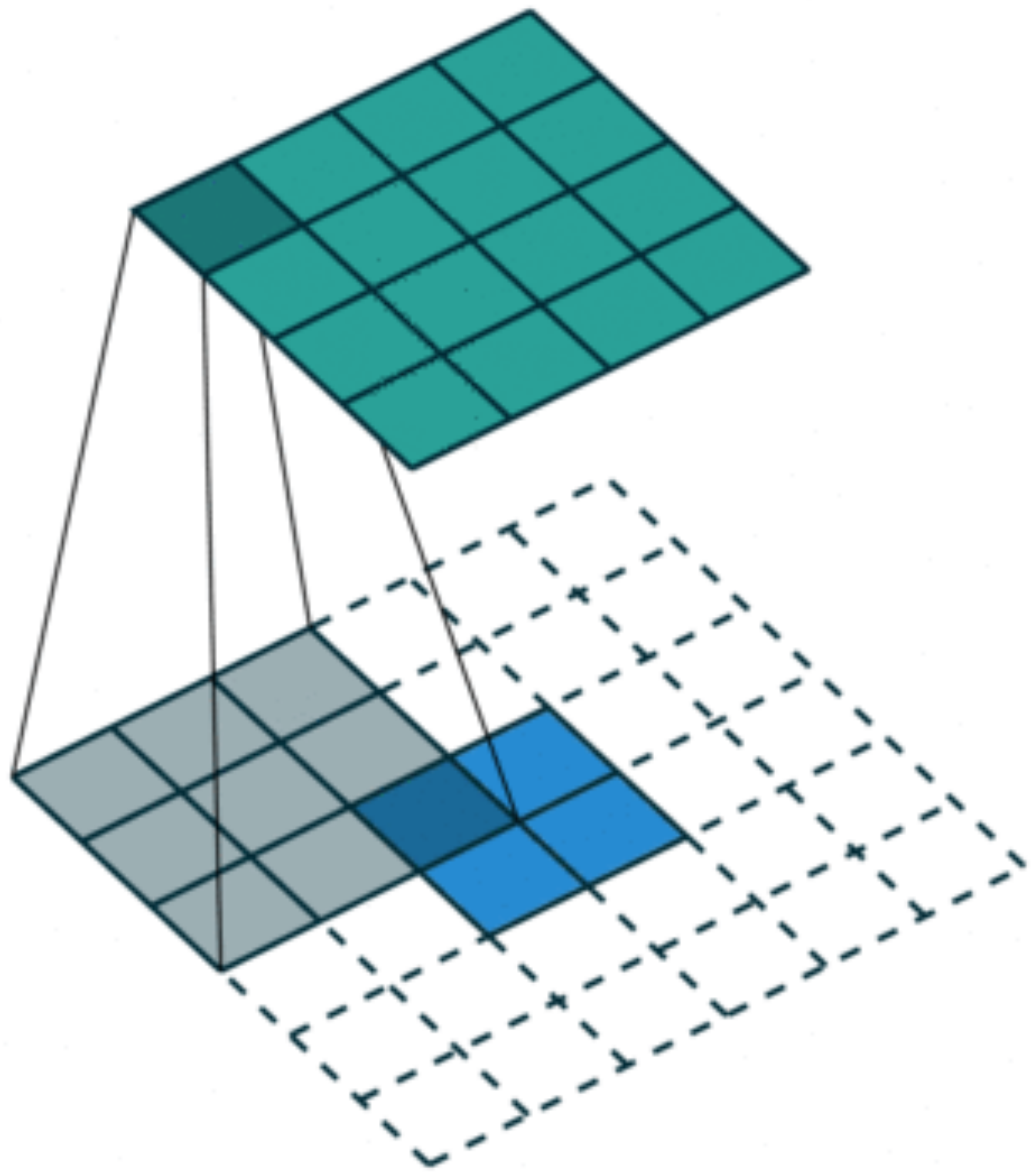
Neosperience

Convolution filter parameters

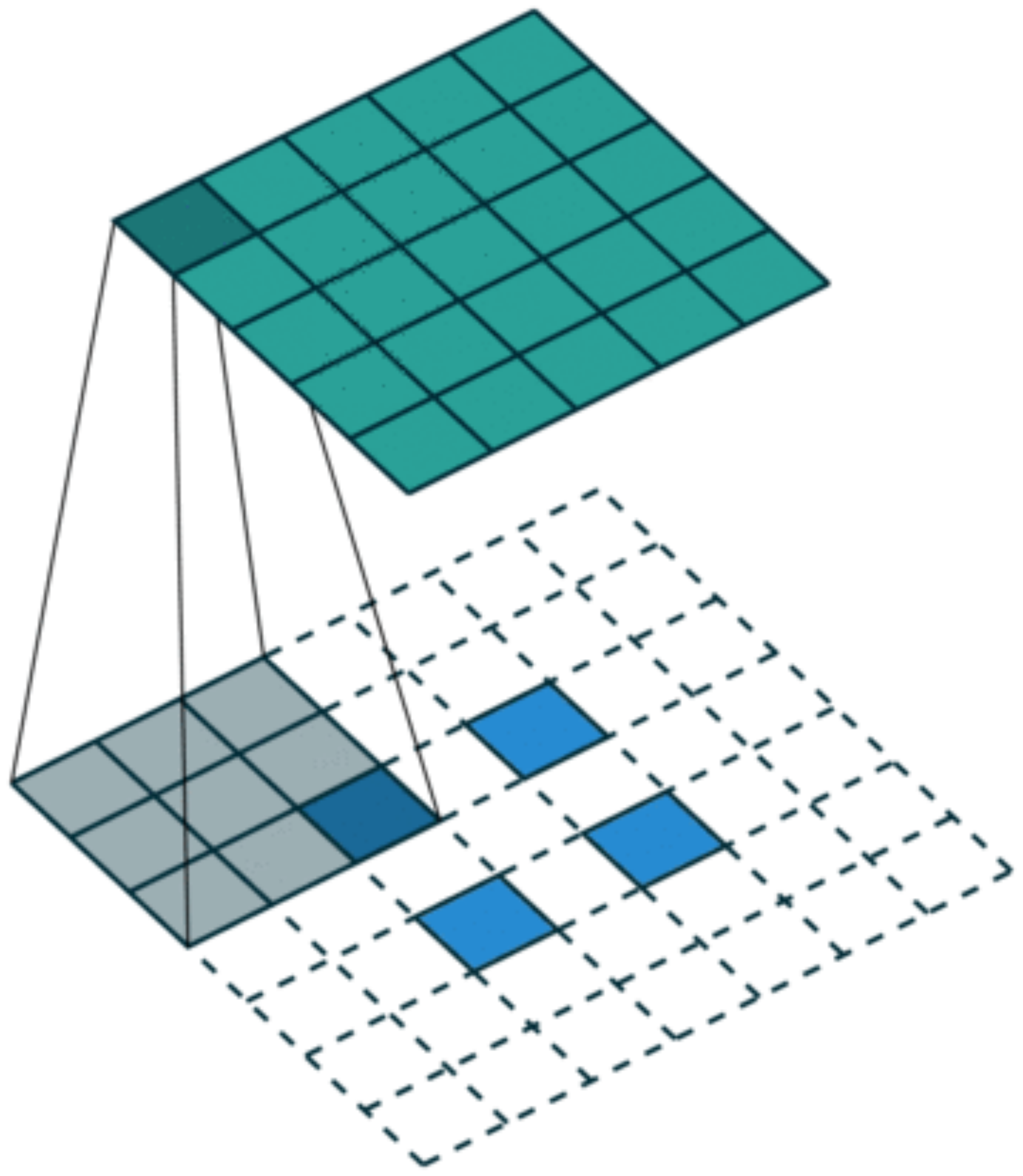
- **Zero-Padding:** pad the input matrix with zeros around the border. it allows us to control the size of the feature maps



1-padding



2-padding

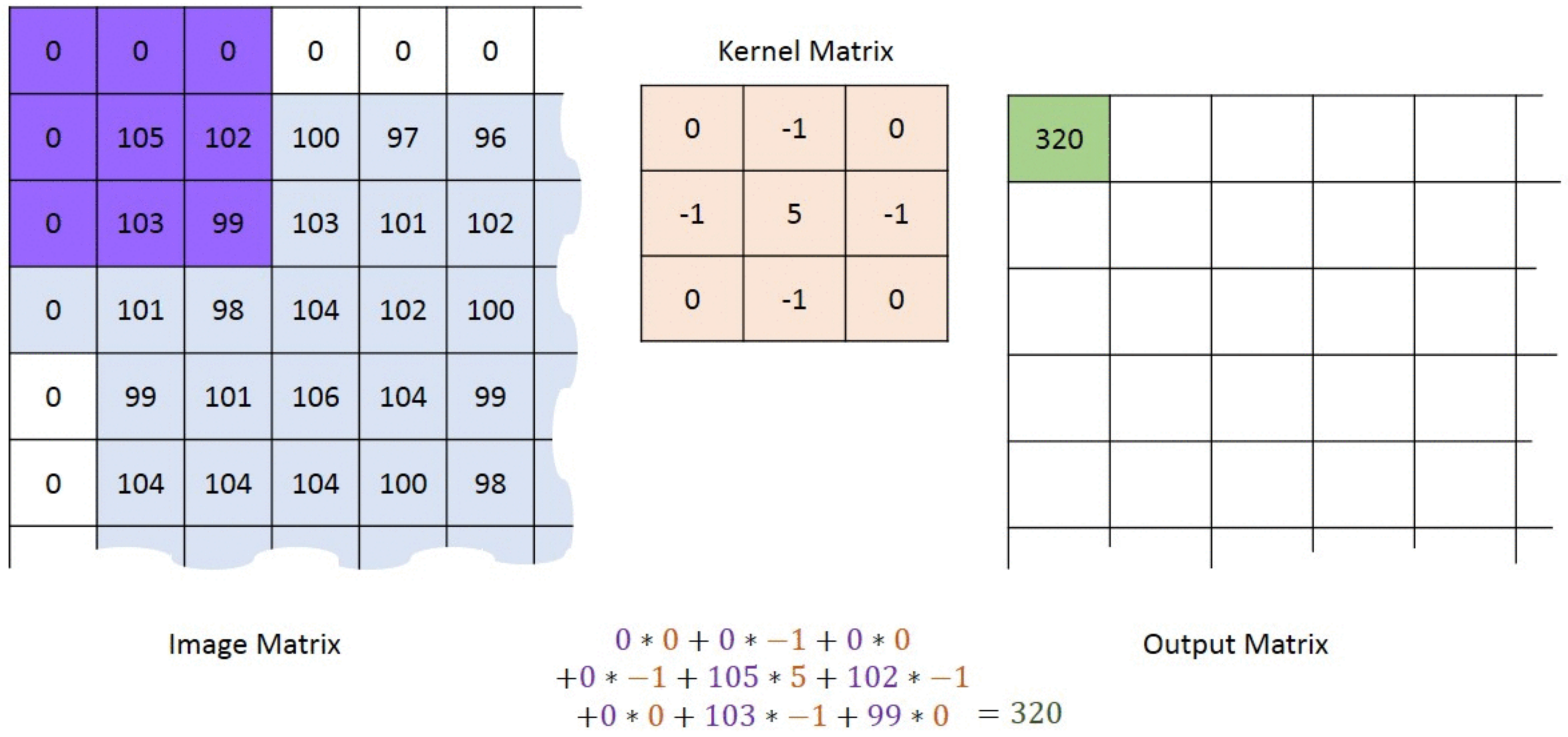


2-padding with up-sampling

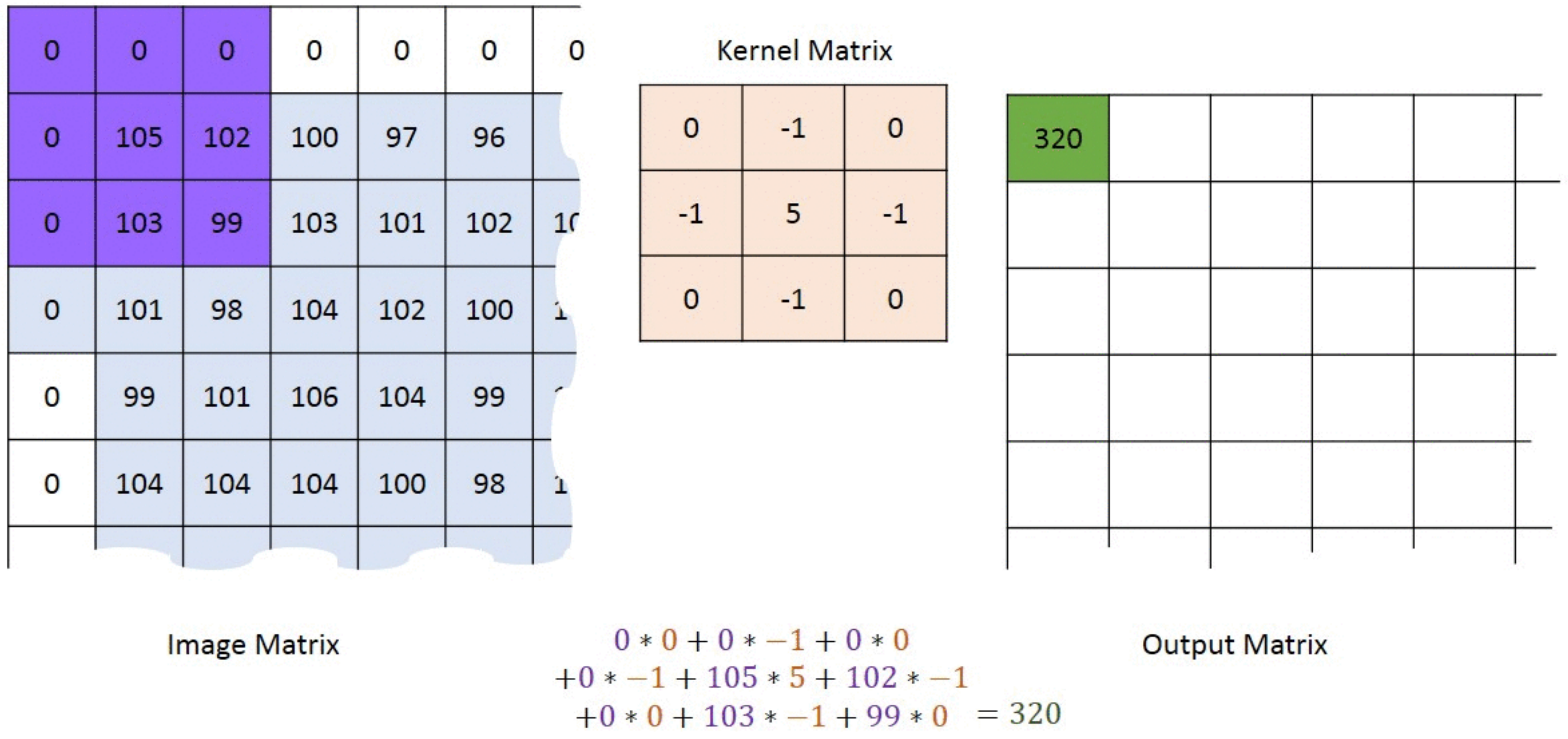
Neosperience

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

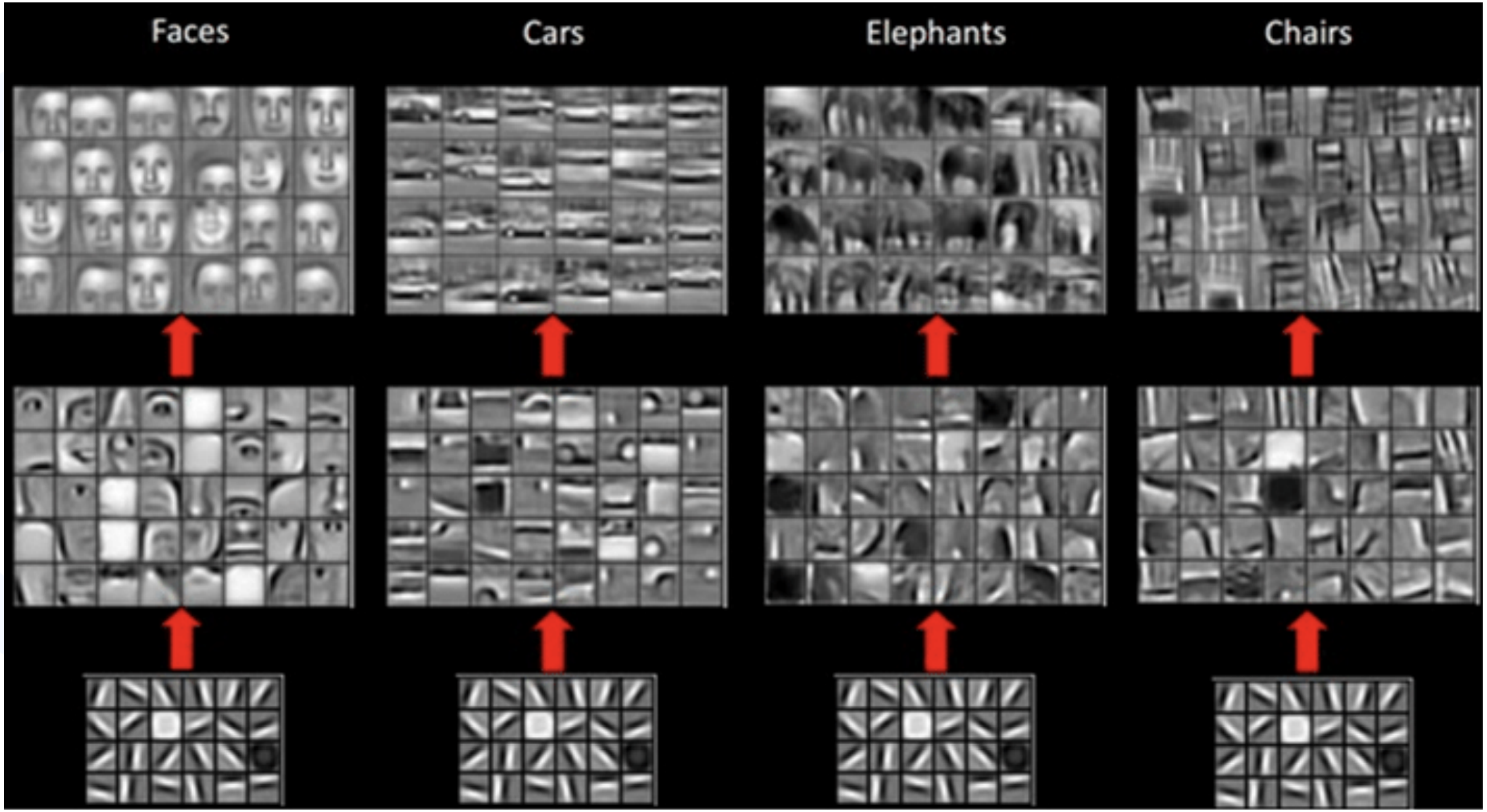
Classic Computer Vision filters

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)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

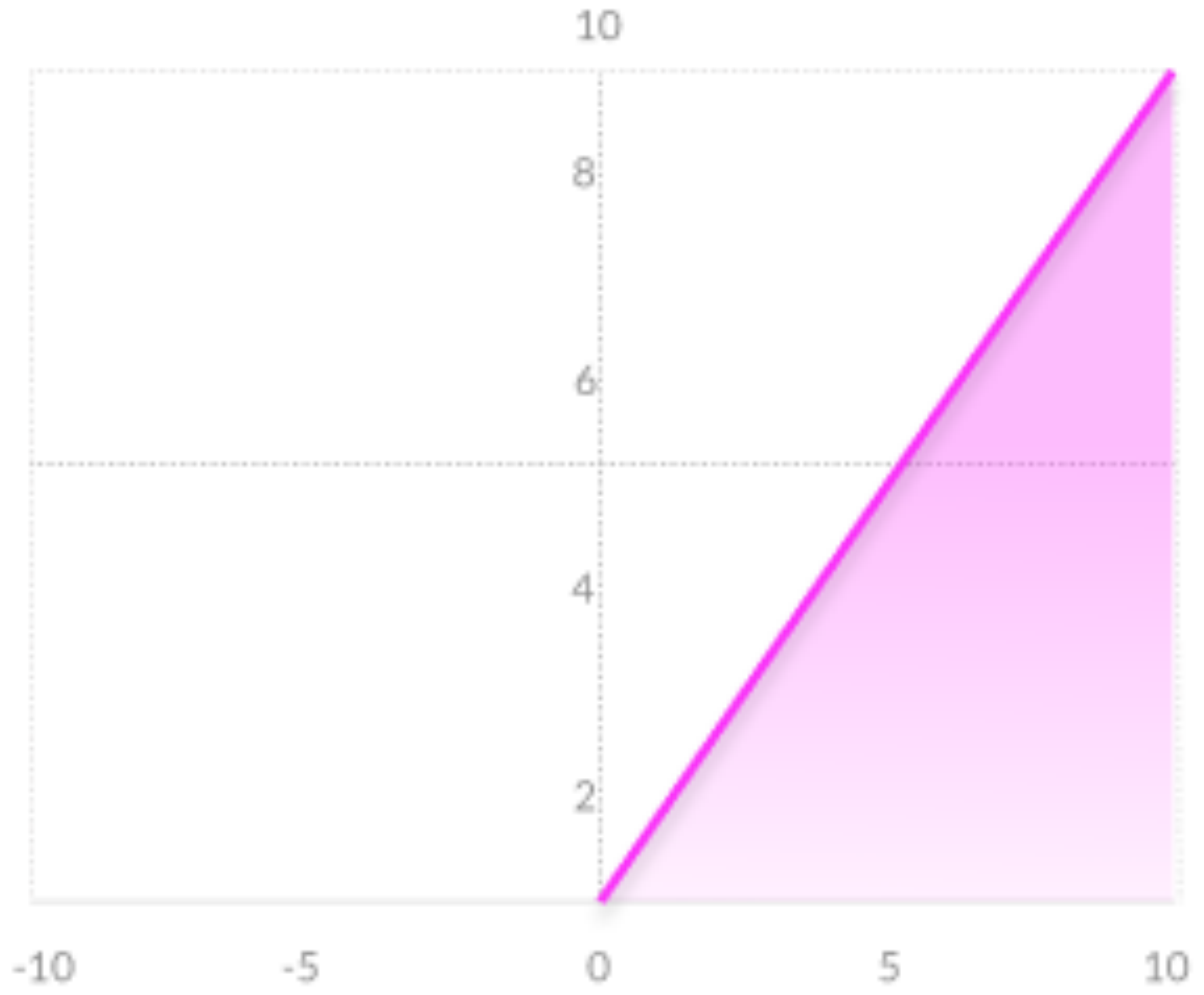
CNN learned filters

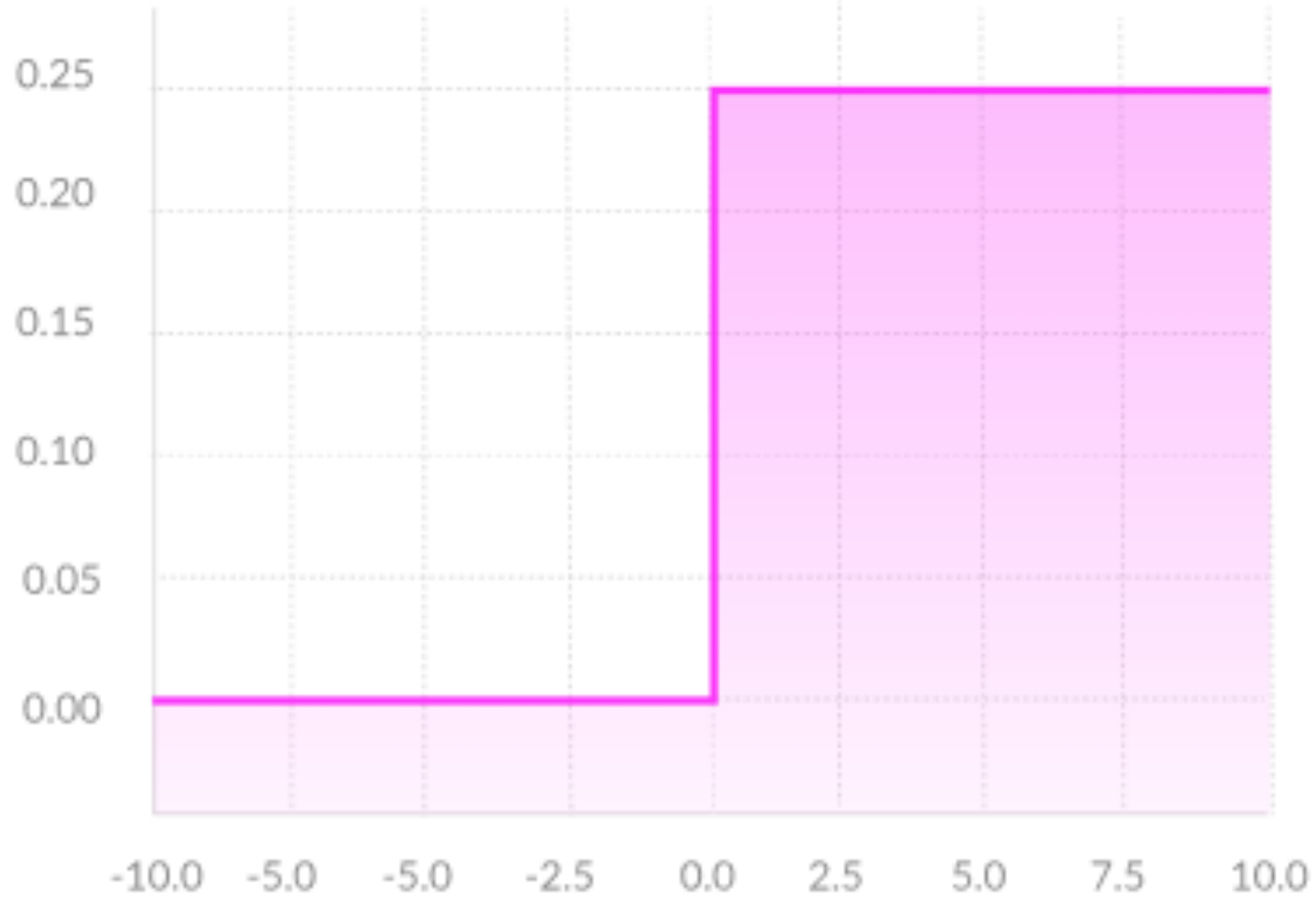
CNN filters are learned by the network itself, surprisingly identifying understandable context features



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.





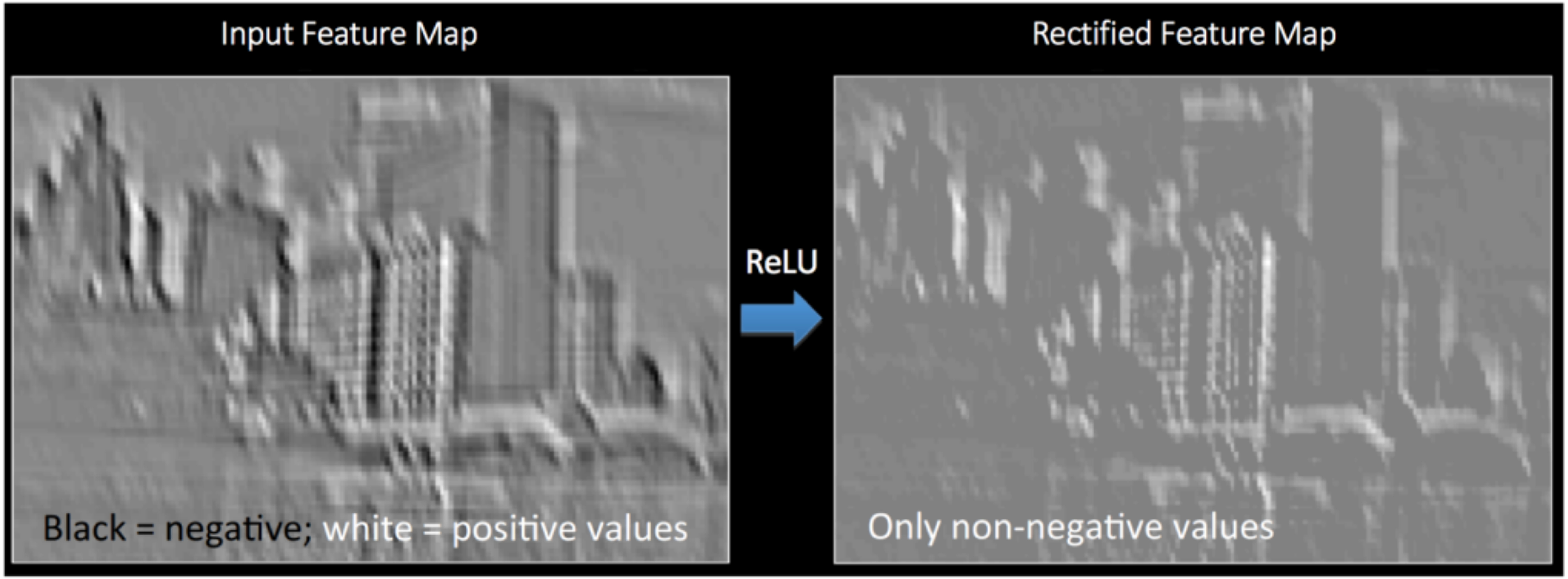
ReLU function

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

ReLU derivative

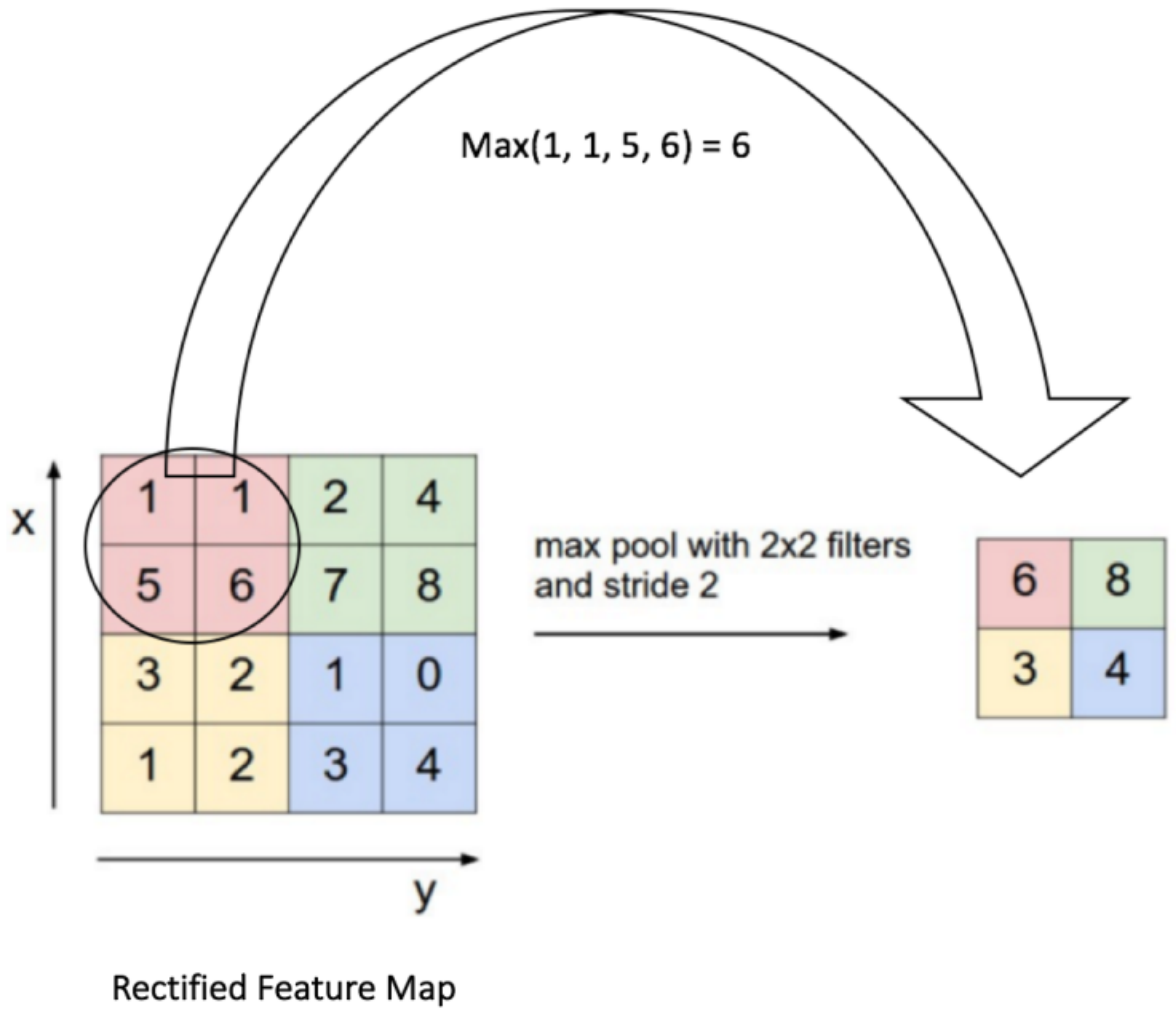
$$\phi'(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases}$$

Non linearity



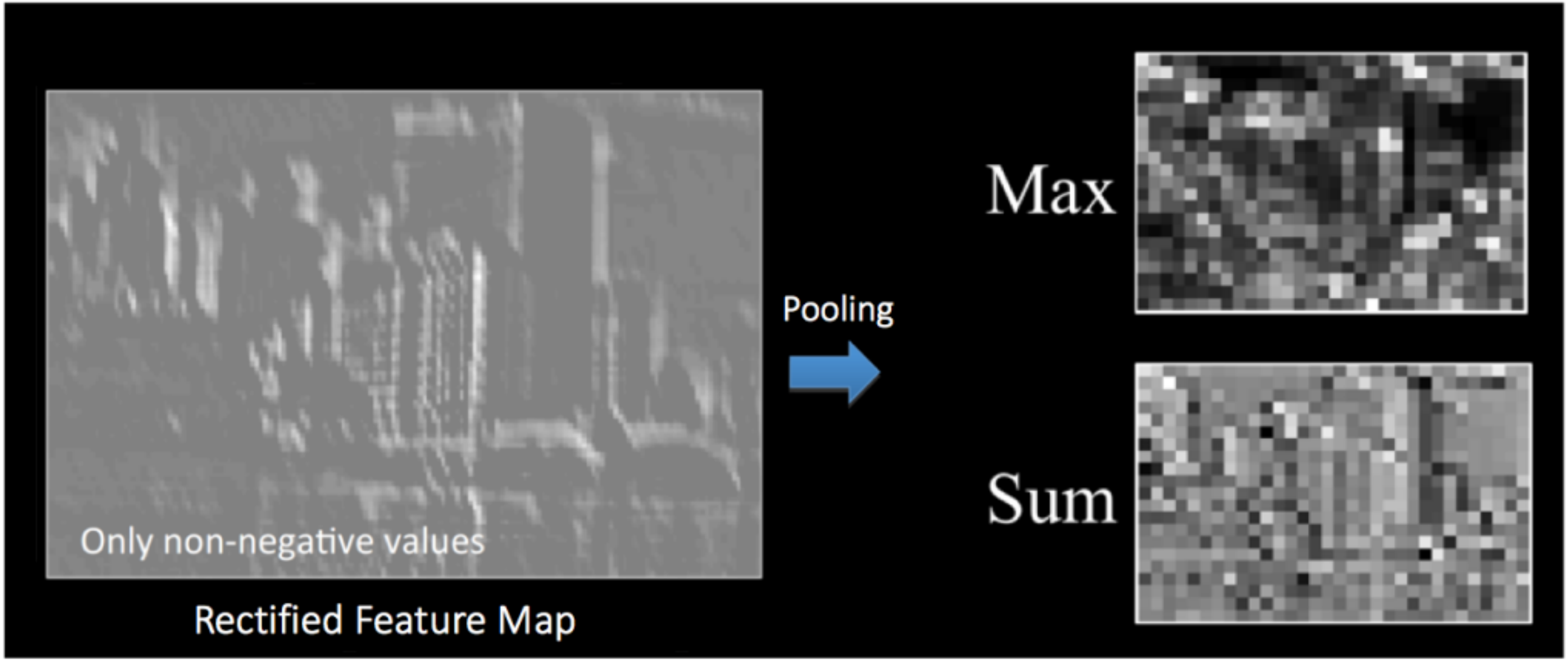
Pooling

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.



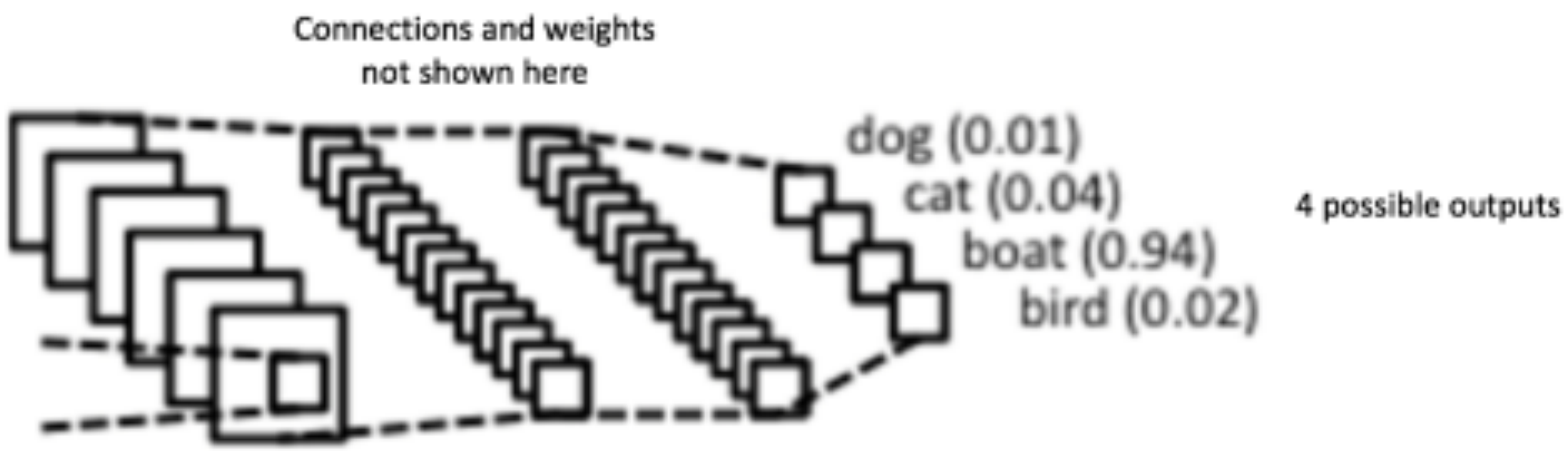
- makes the **input representations** (feature dimension) **smaller** and more manageable
- **reduces the number of parameters** and computations in the network
- makes the network **invariant to small transformations**, distortions and translations in the input image (a small distortion in input will 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

Pooling


Only non-negative values
Rectified Feature Map
Pooling
Max
Sum

Training and loss function

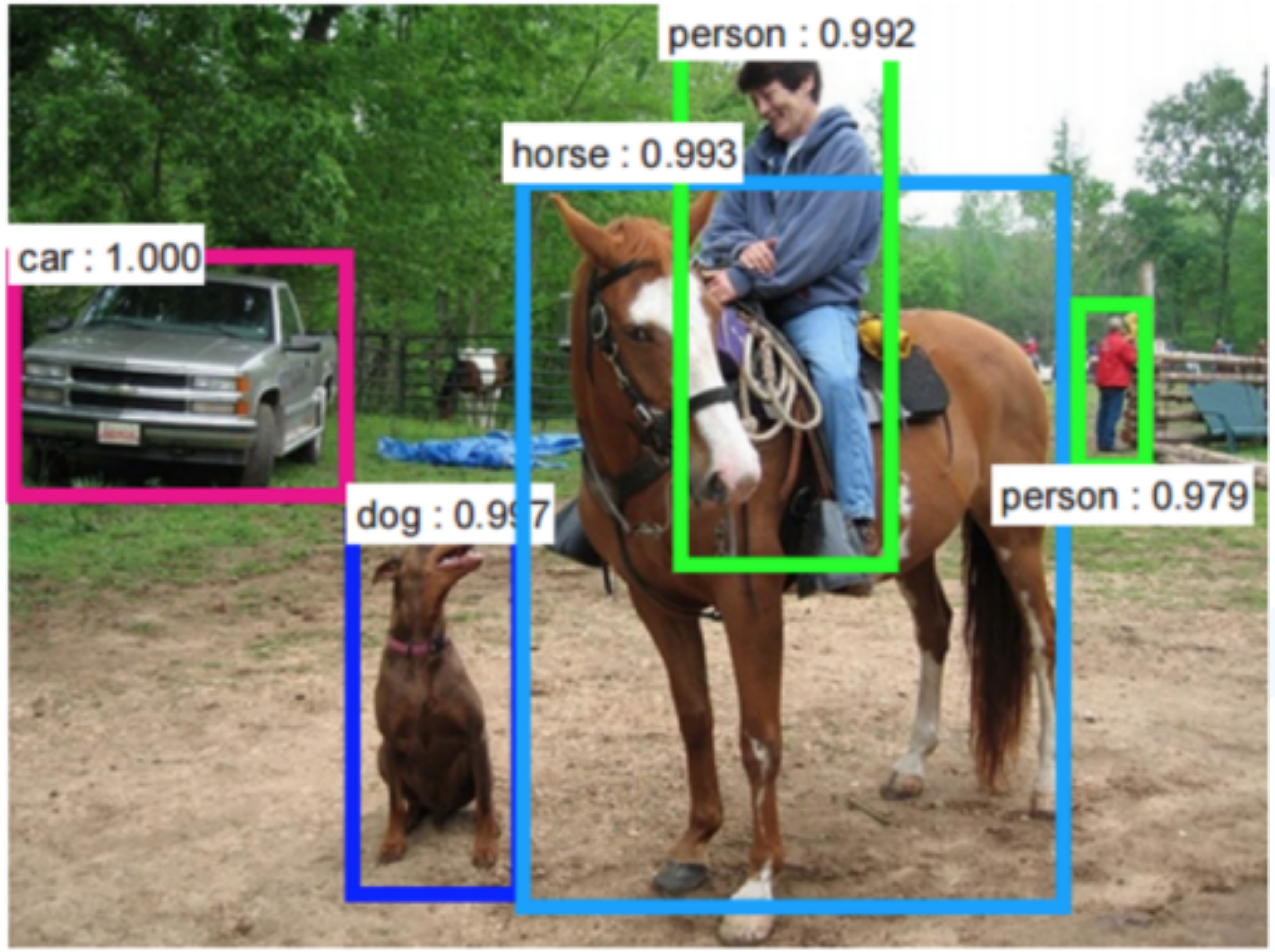
- The Fully Connected layer is a traditional Multi Layer Perceptron that uses a **Softmax** activation function in the output layer, flattening the output of convolutional and pooling layers
- The output from the convolutional and pooling layers represent high-level features of the input image
- The 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



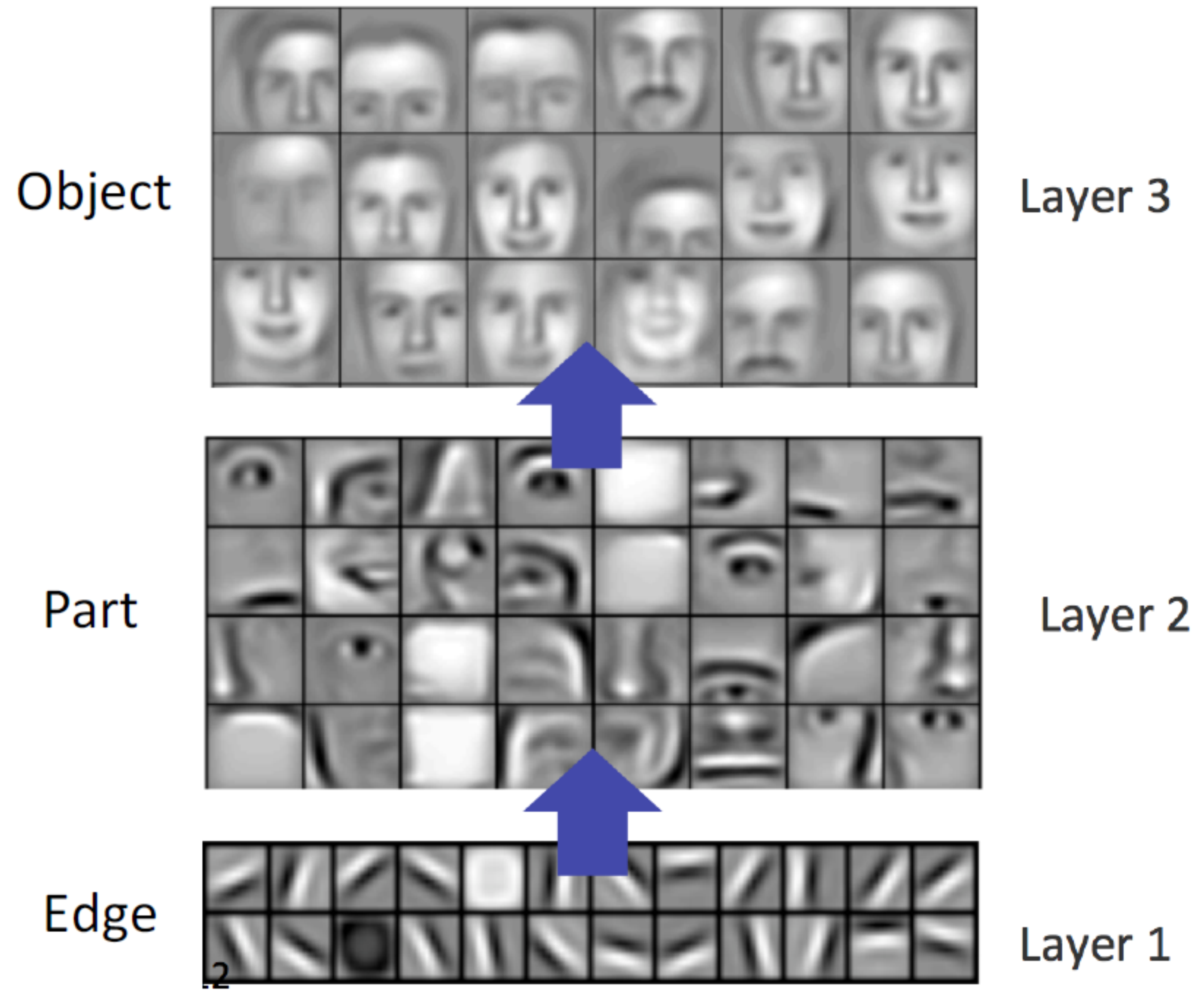
$$\mathrm{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

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.

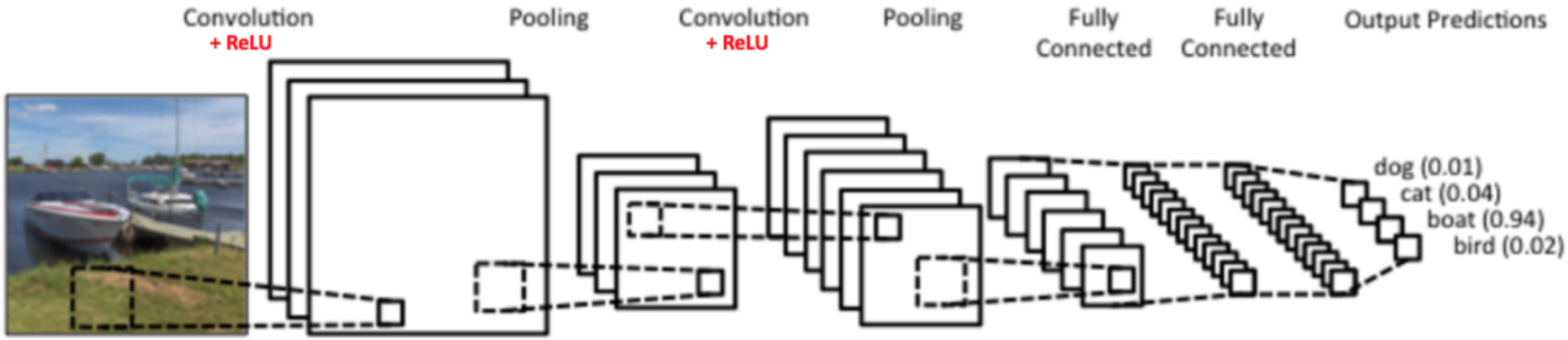


Neosperience

Convolutional Neural Network

Key components of a CNN are the following:

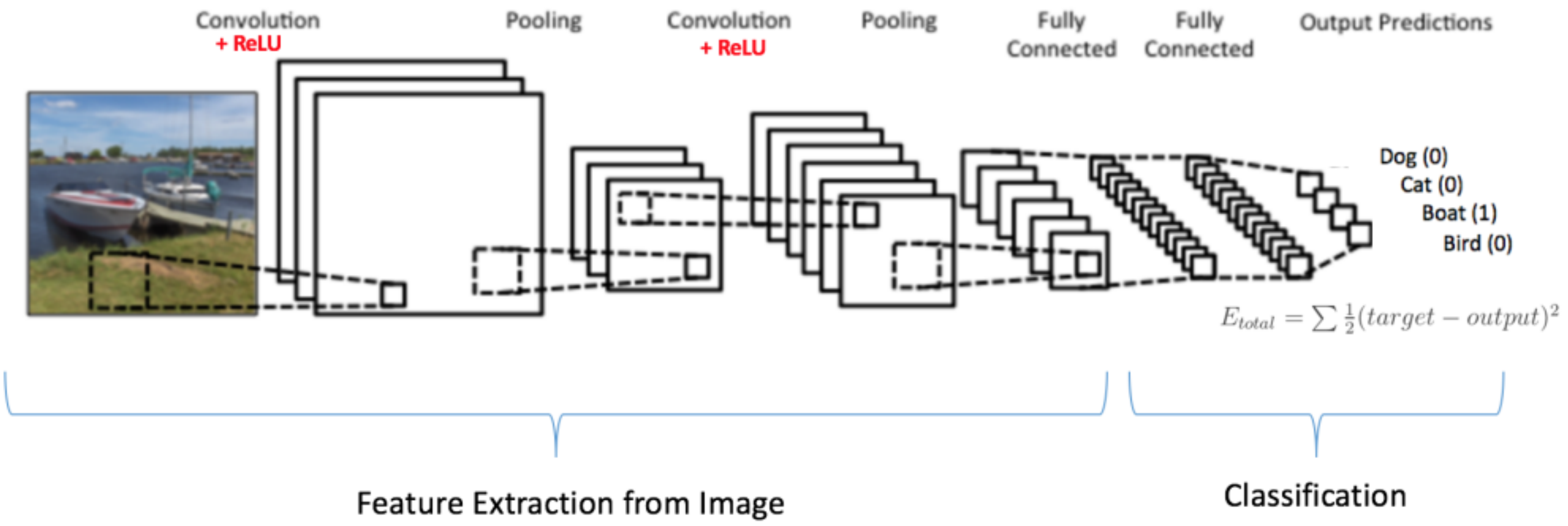
- Convolution
- Non Linearity (activation function)
- Pooling or Sub-sampling
- Classification (fully connected layer) and training



LeNet

Training and loss function

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



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

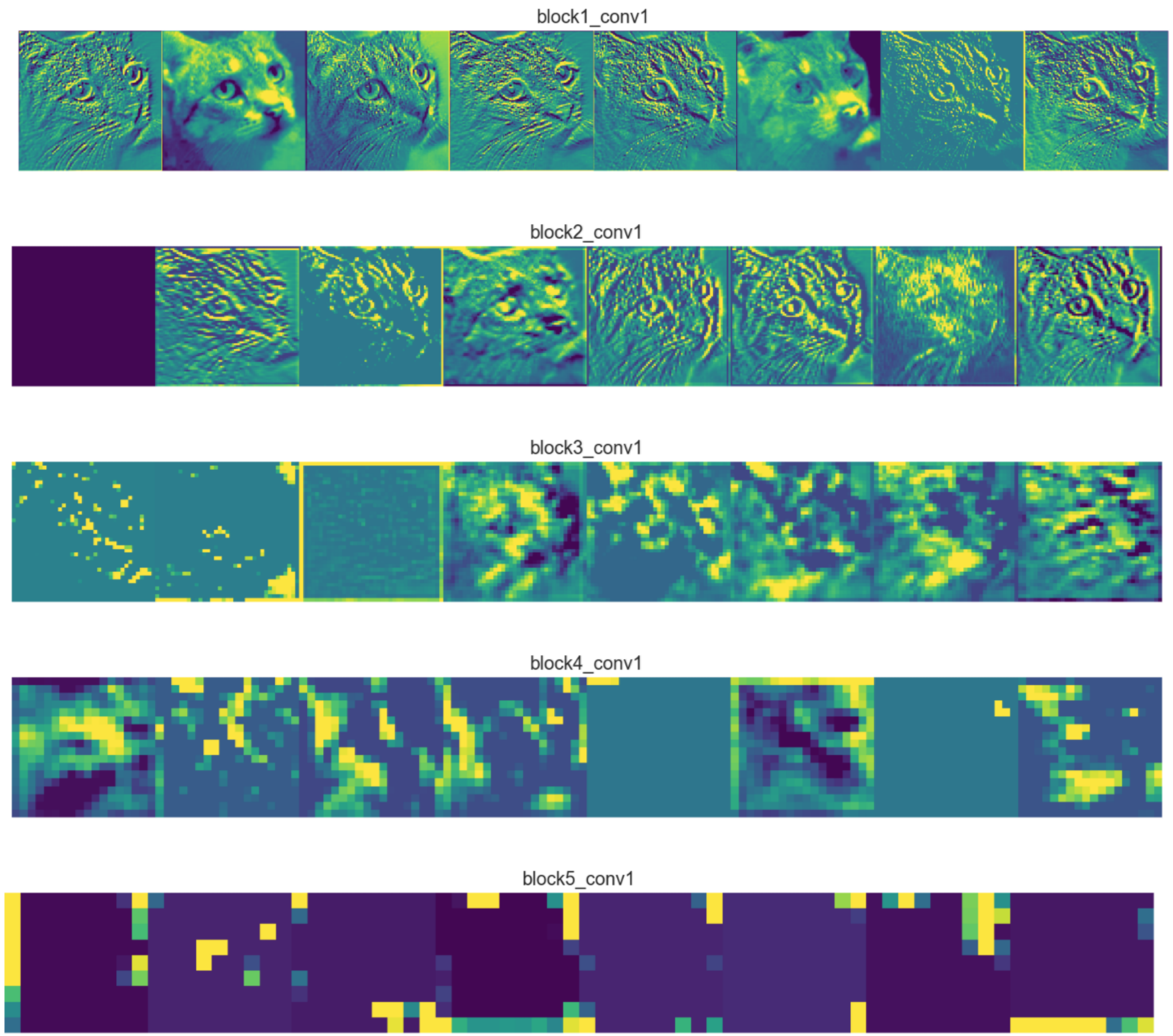
$$\frac{1}{n}\sum_{i=1}^{n}(TP_i - OP_i)^2$$

Cross-Entropy

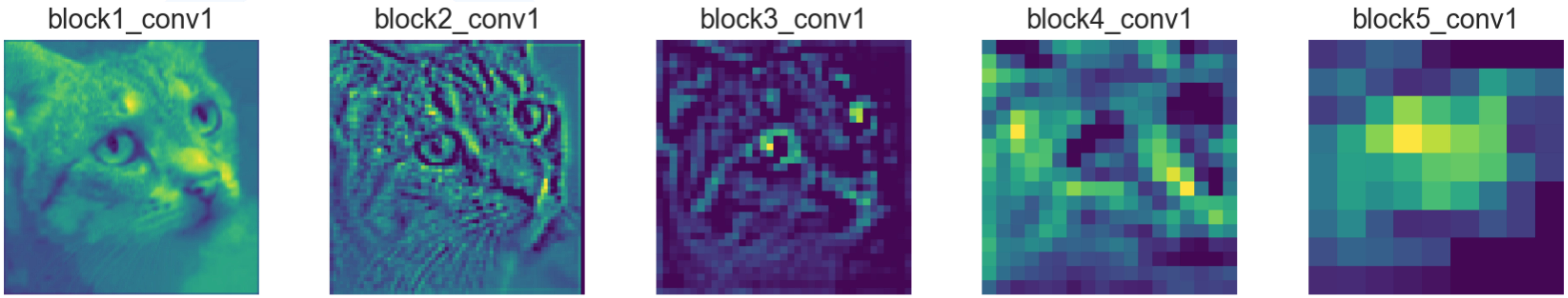
$$-\sum_{i=1}^{n} TP_i \log(OP_i)$$

1) 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
2) Repeat steps 2-4 with all images in the training set

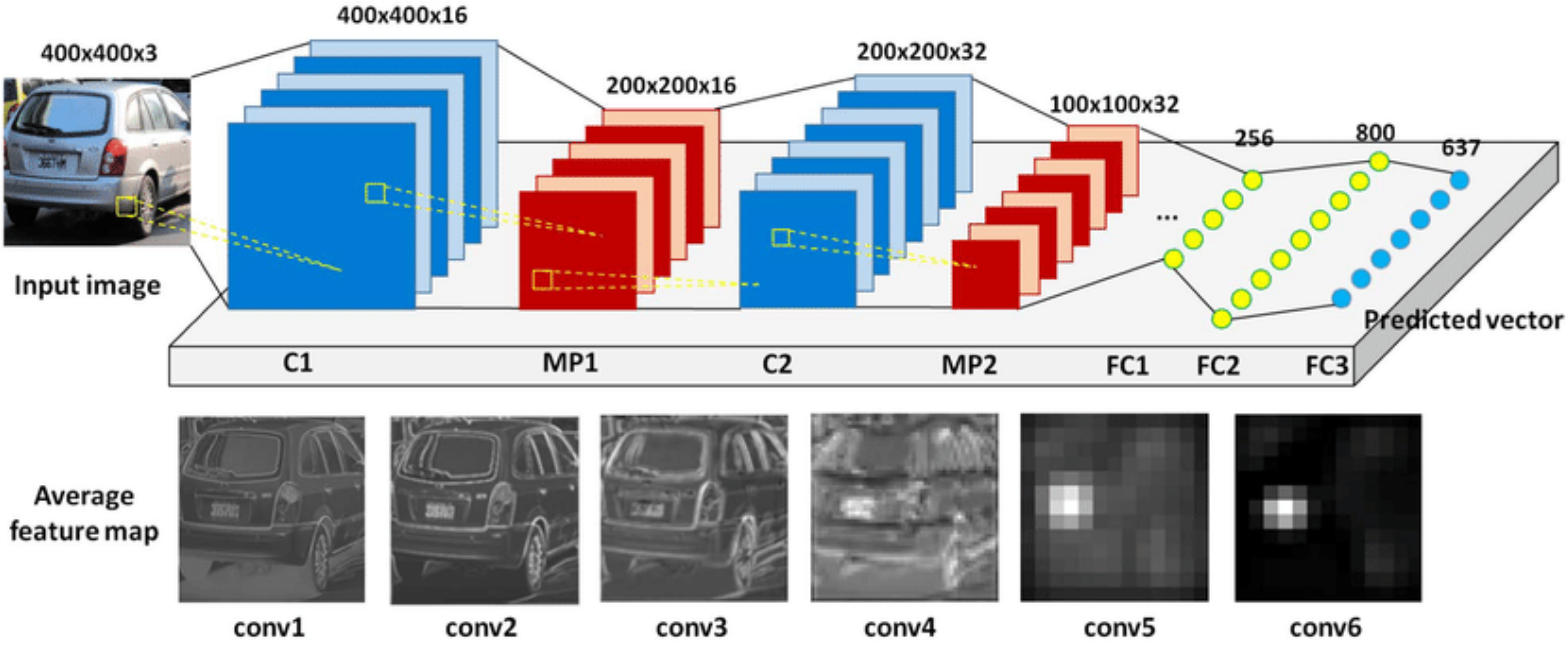
Visualizing CNN



Visualizing CNN



Visualizing CNN



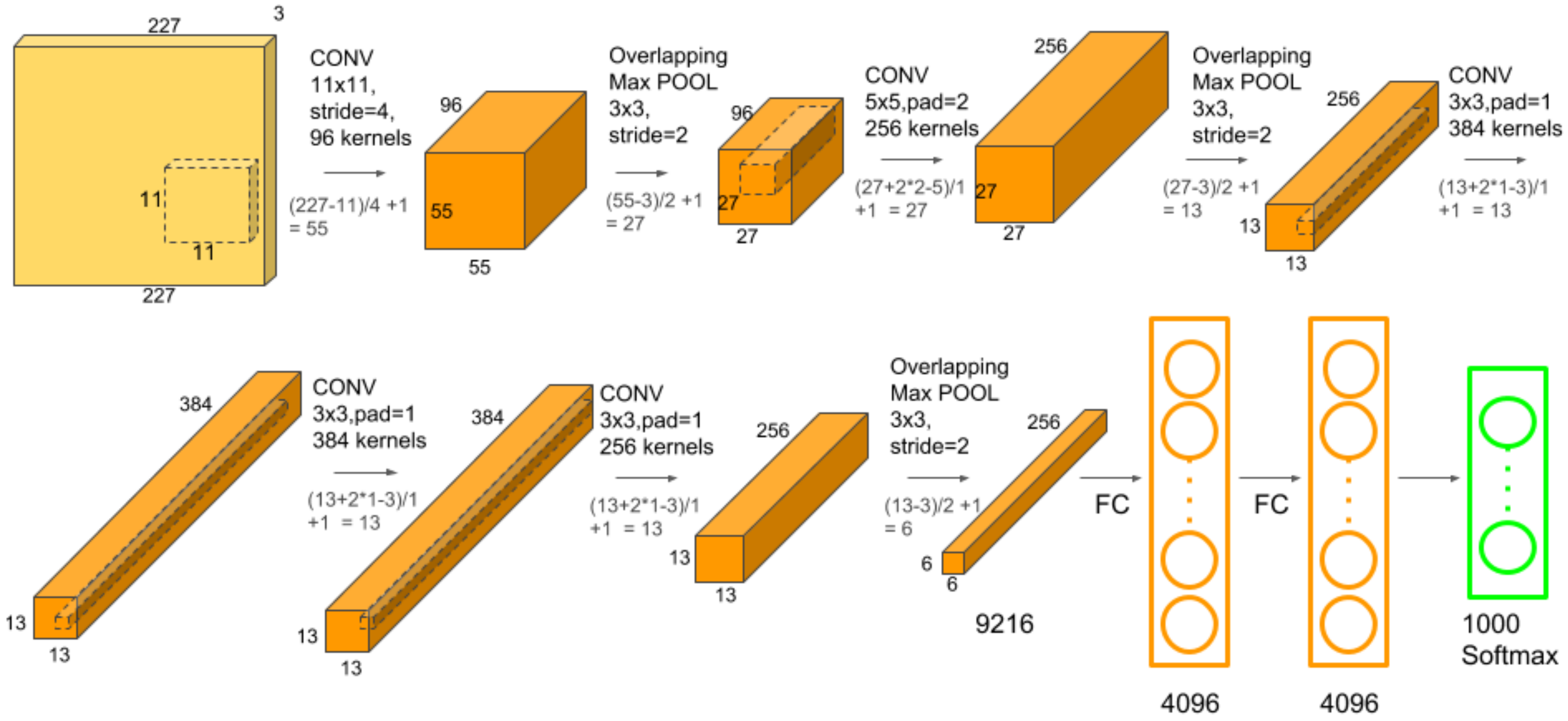
Useful implementation tips

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)

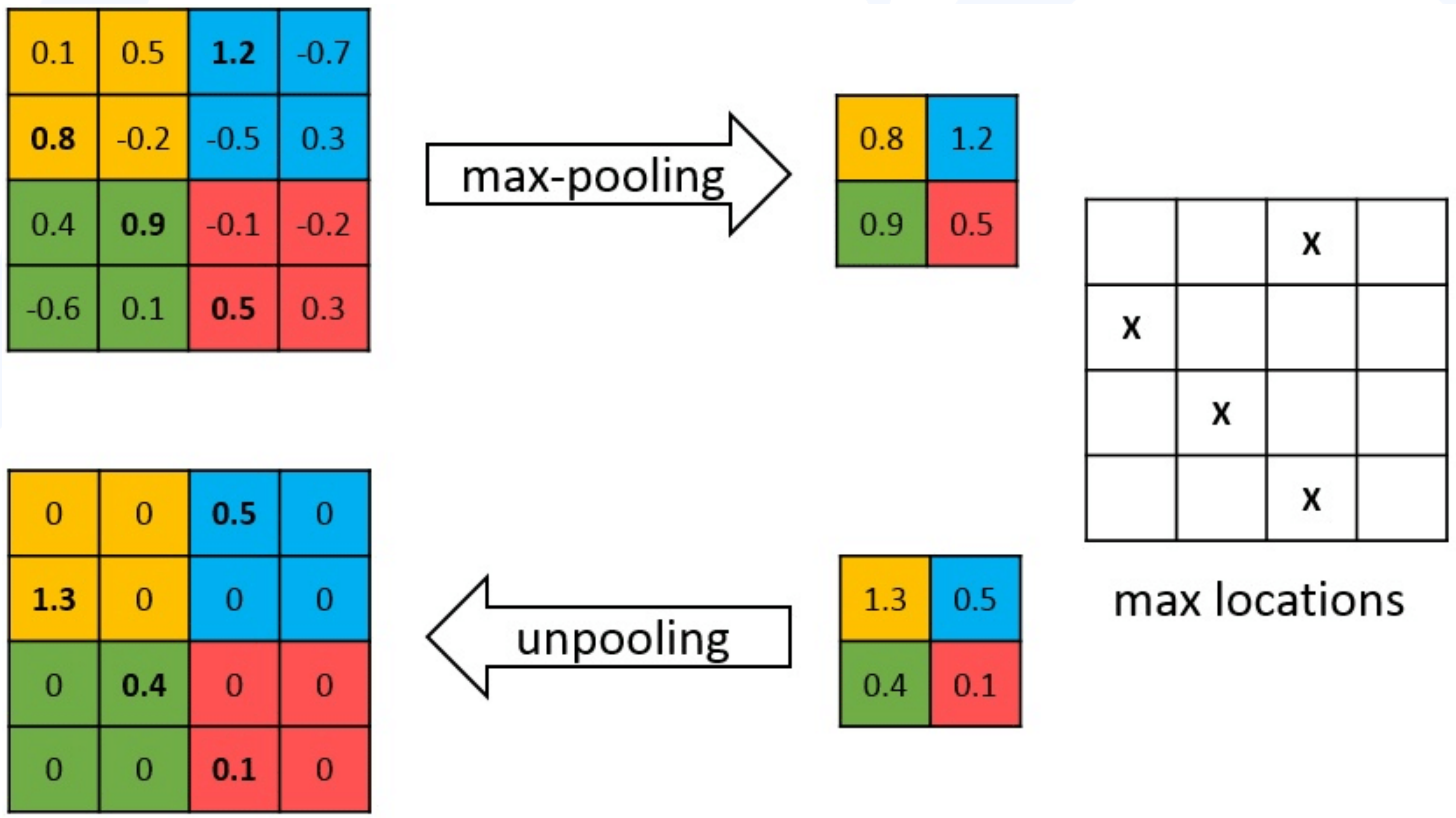
- AlexNet was much larger than previous CNNs. It has 60 million parameters and 650,000 neurons and took five to six days to train on two GTX 580 3GB GPUs.
- consists of **5 Convolutional Layers** and **3 Fully Connected Layers**



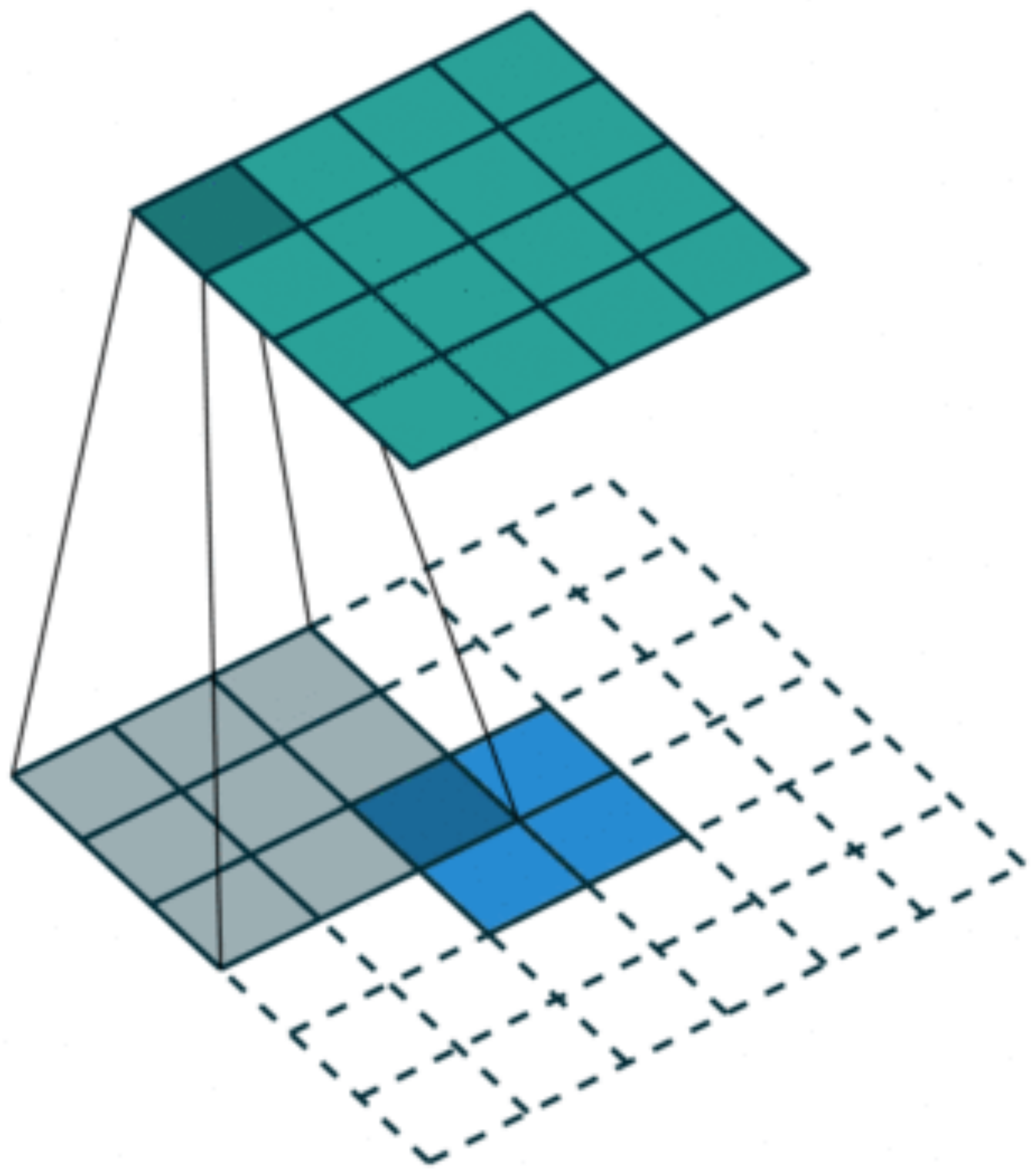
CNN Architectures: ZFnet (Zeiler & Fergus - 2013)

- 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



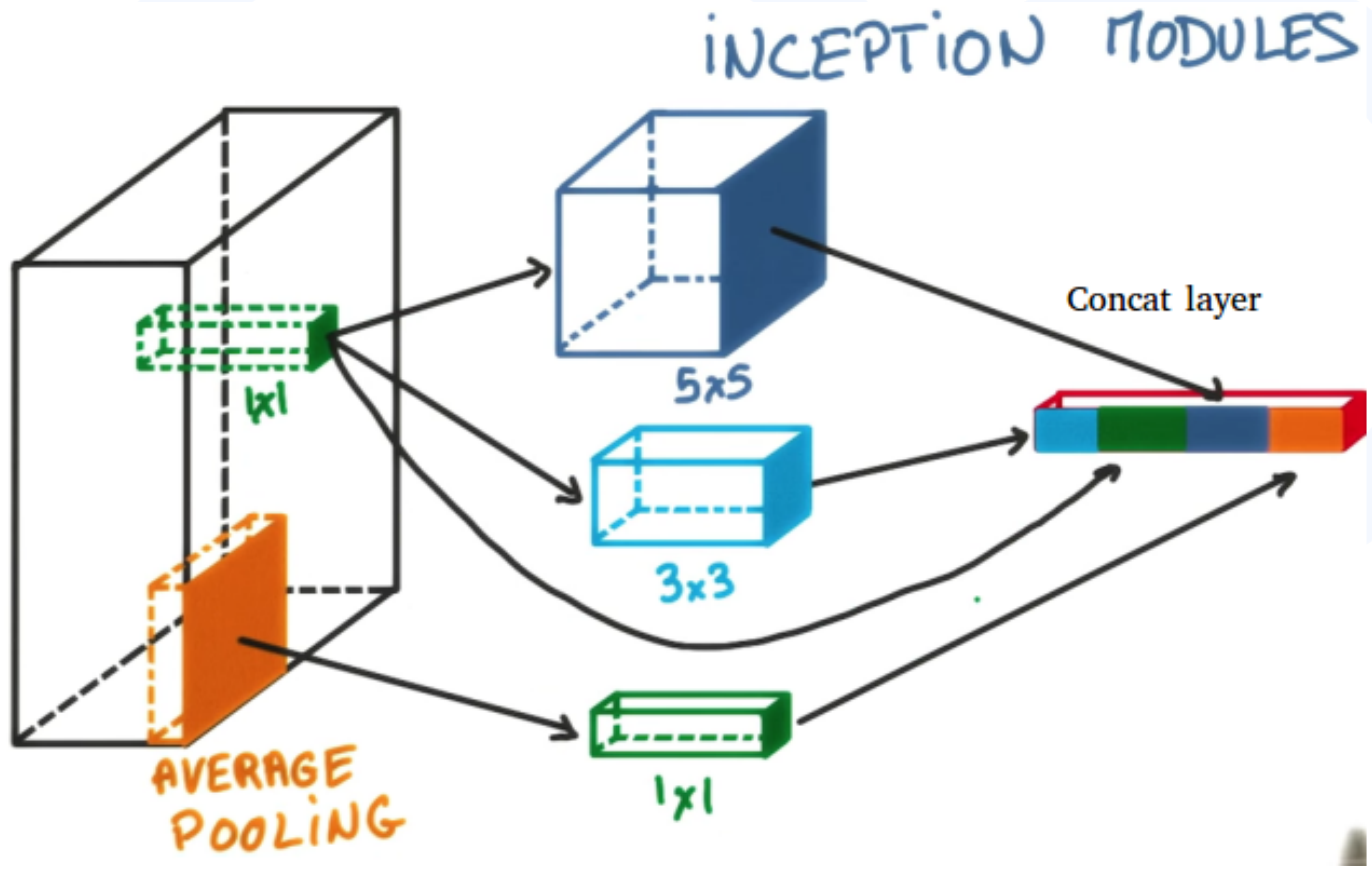
Deconvolution

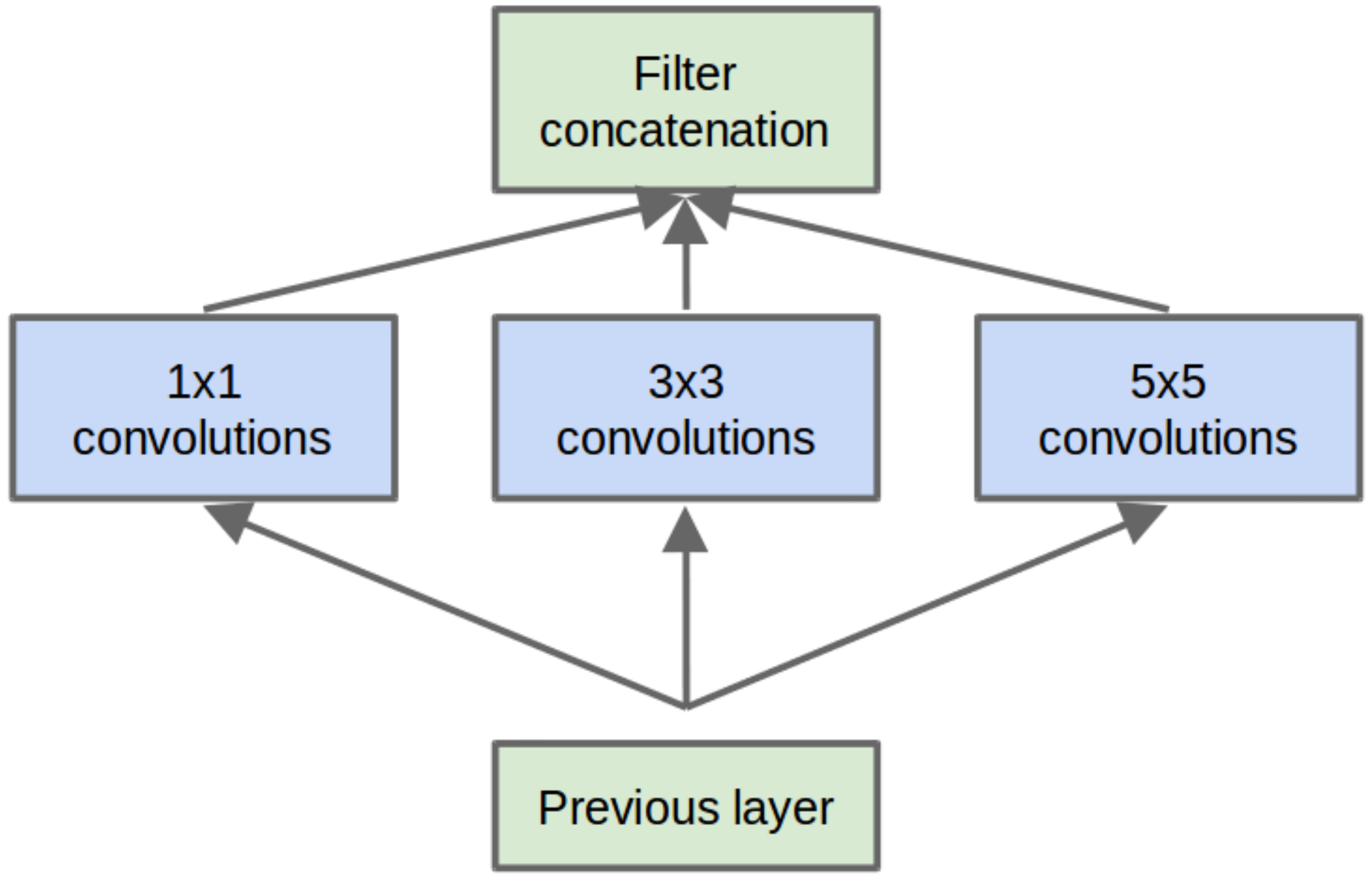


Blue is input, cyan is output

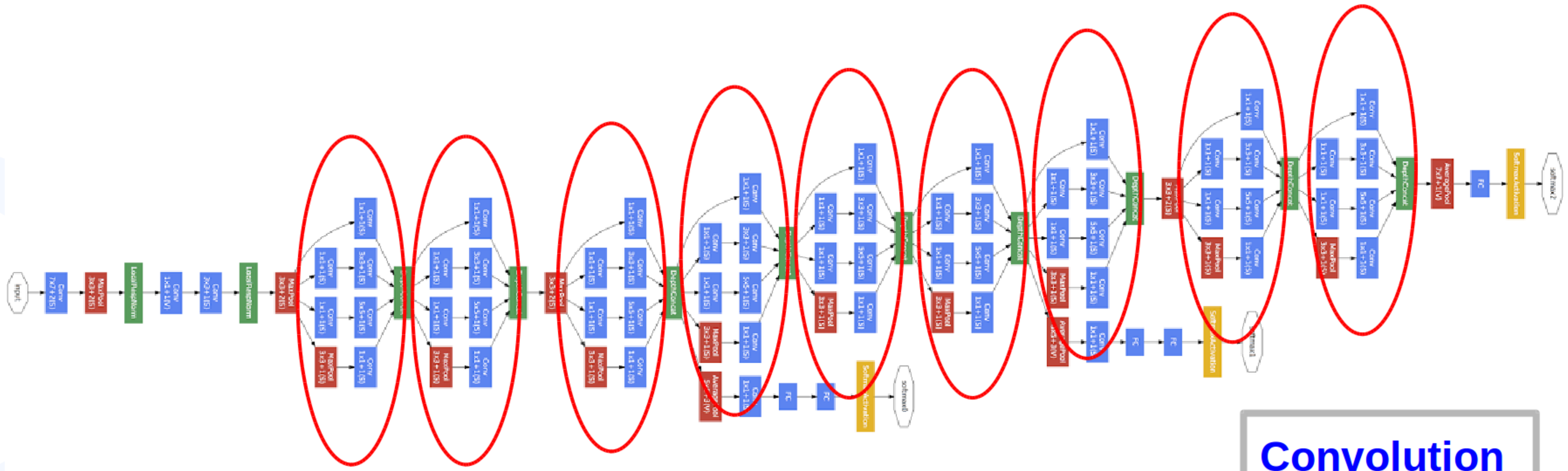
CNN Architectures: GoogLeNet (2014)

- 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.





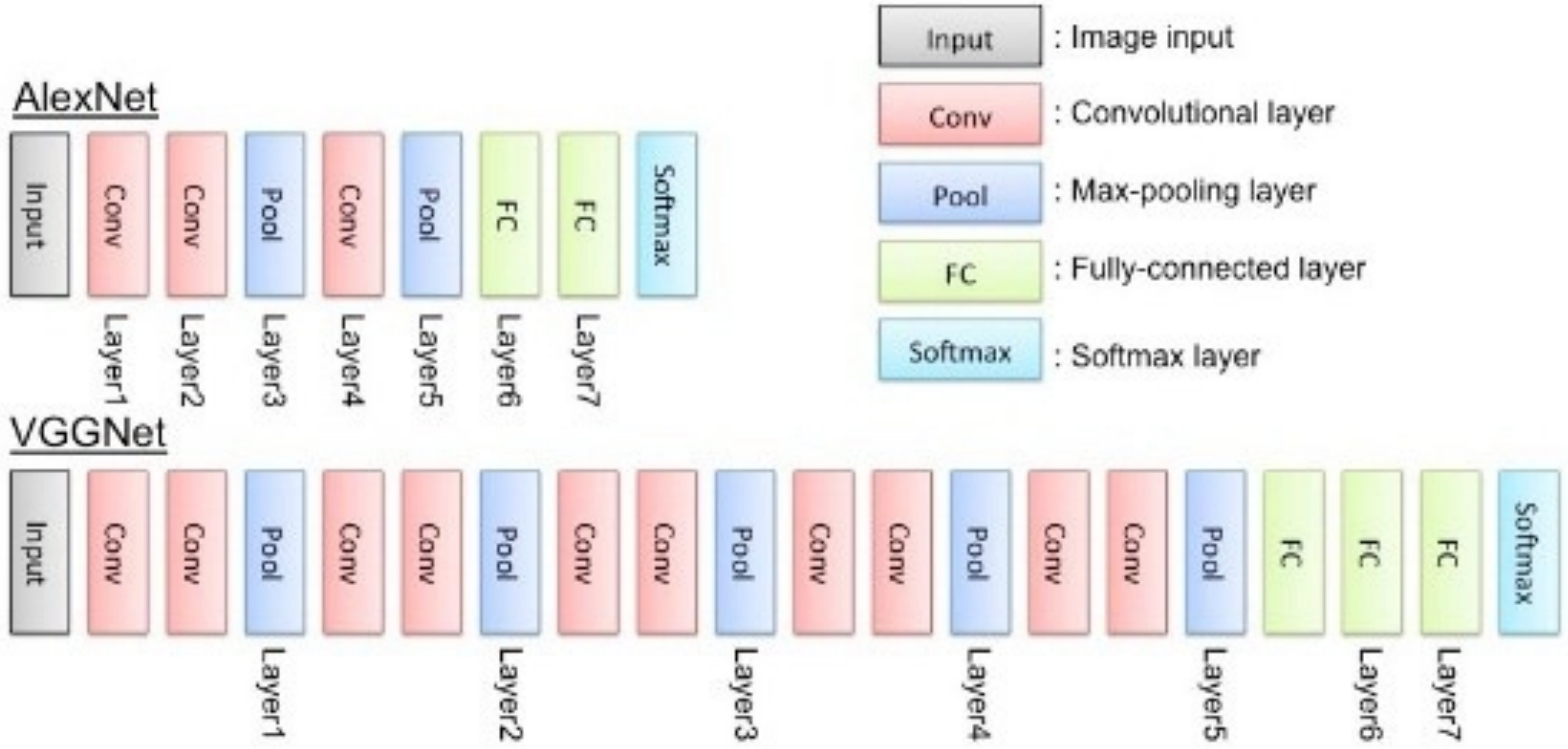
CNN Architectures: GoogLeNet (2014)



Convolution
Pooling
Softmax
Concat/Normalize

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



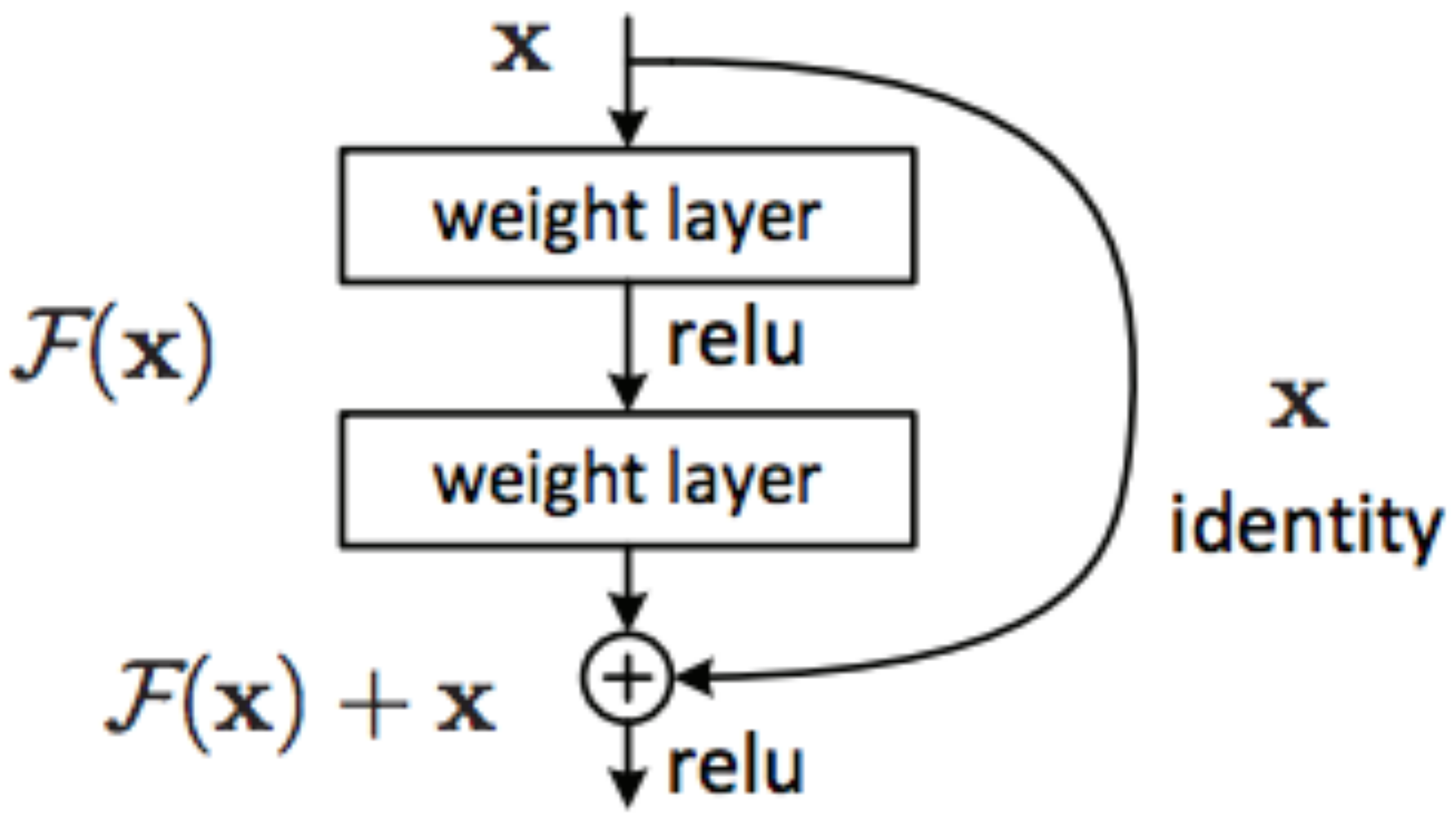
CNN Architectures: ResNets (2015)

- 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

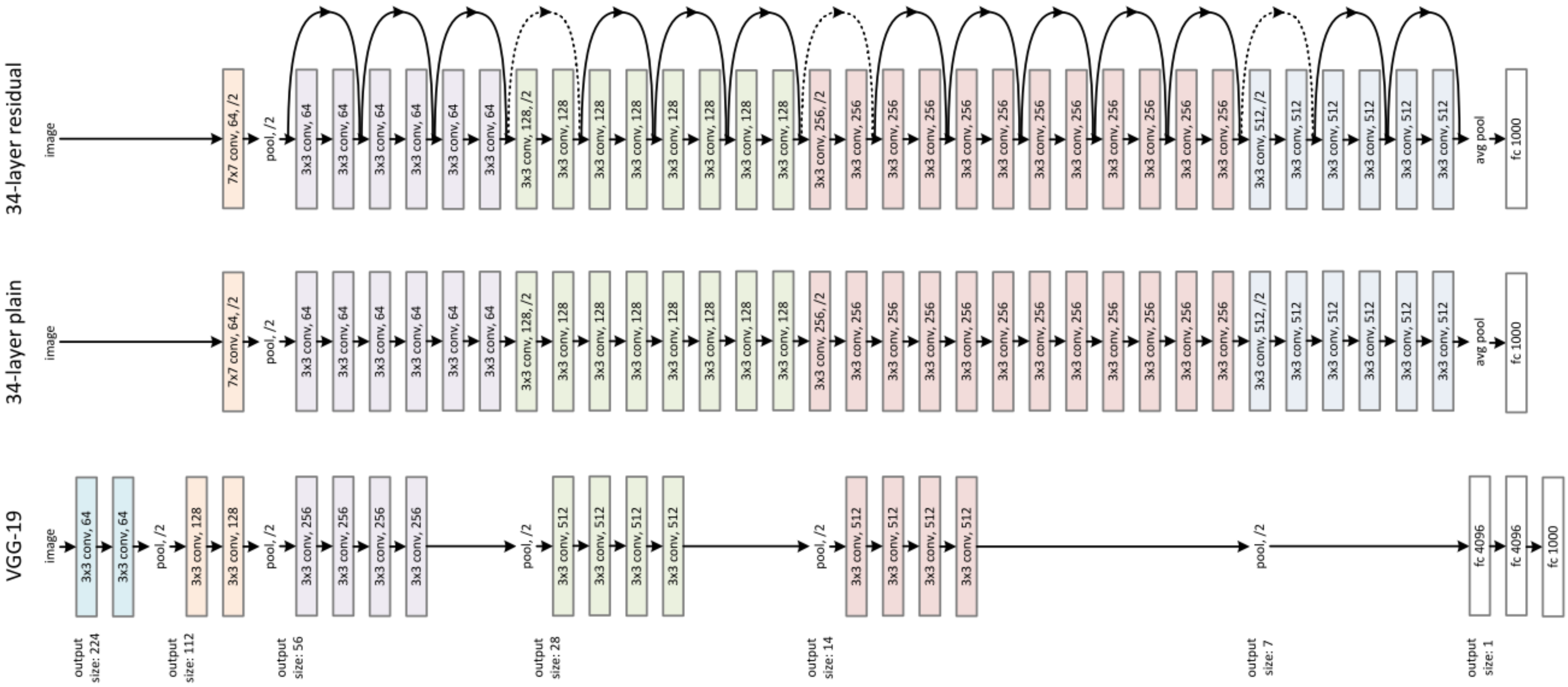
$$f(x) = x$$

- 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

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



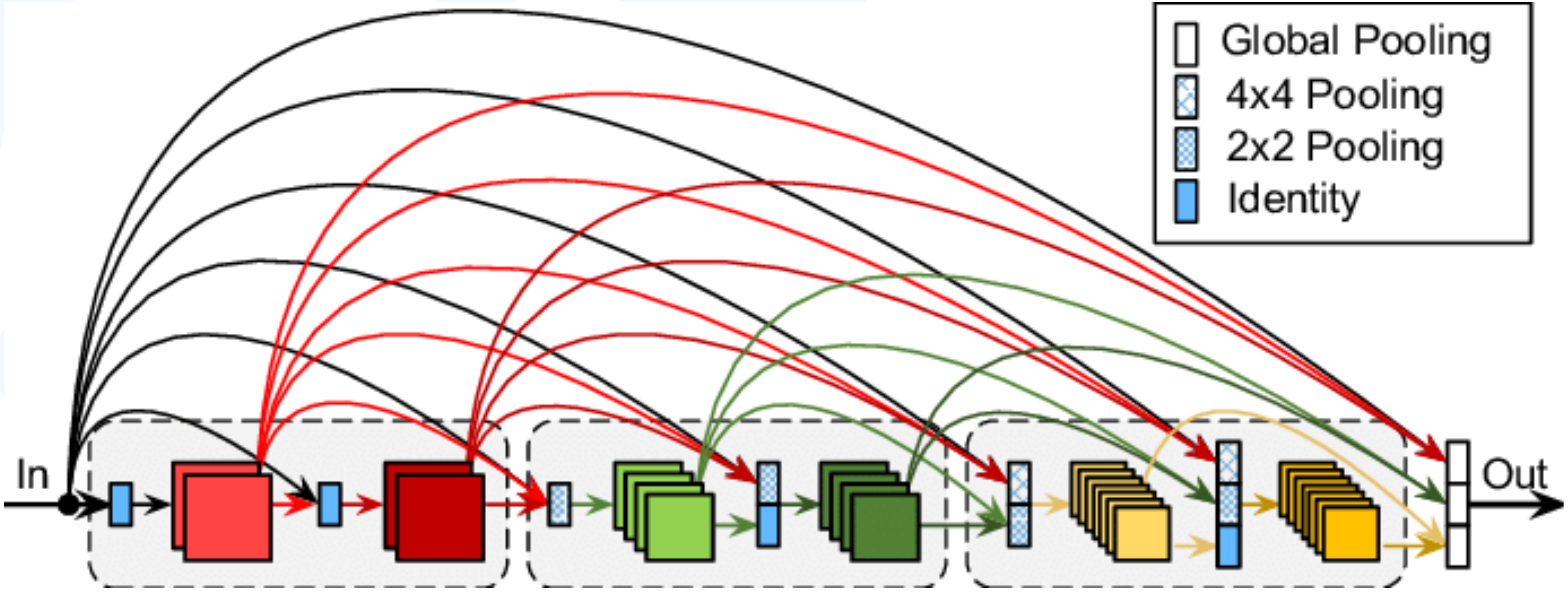
CNN Architectures: ResNets (2015)



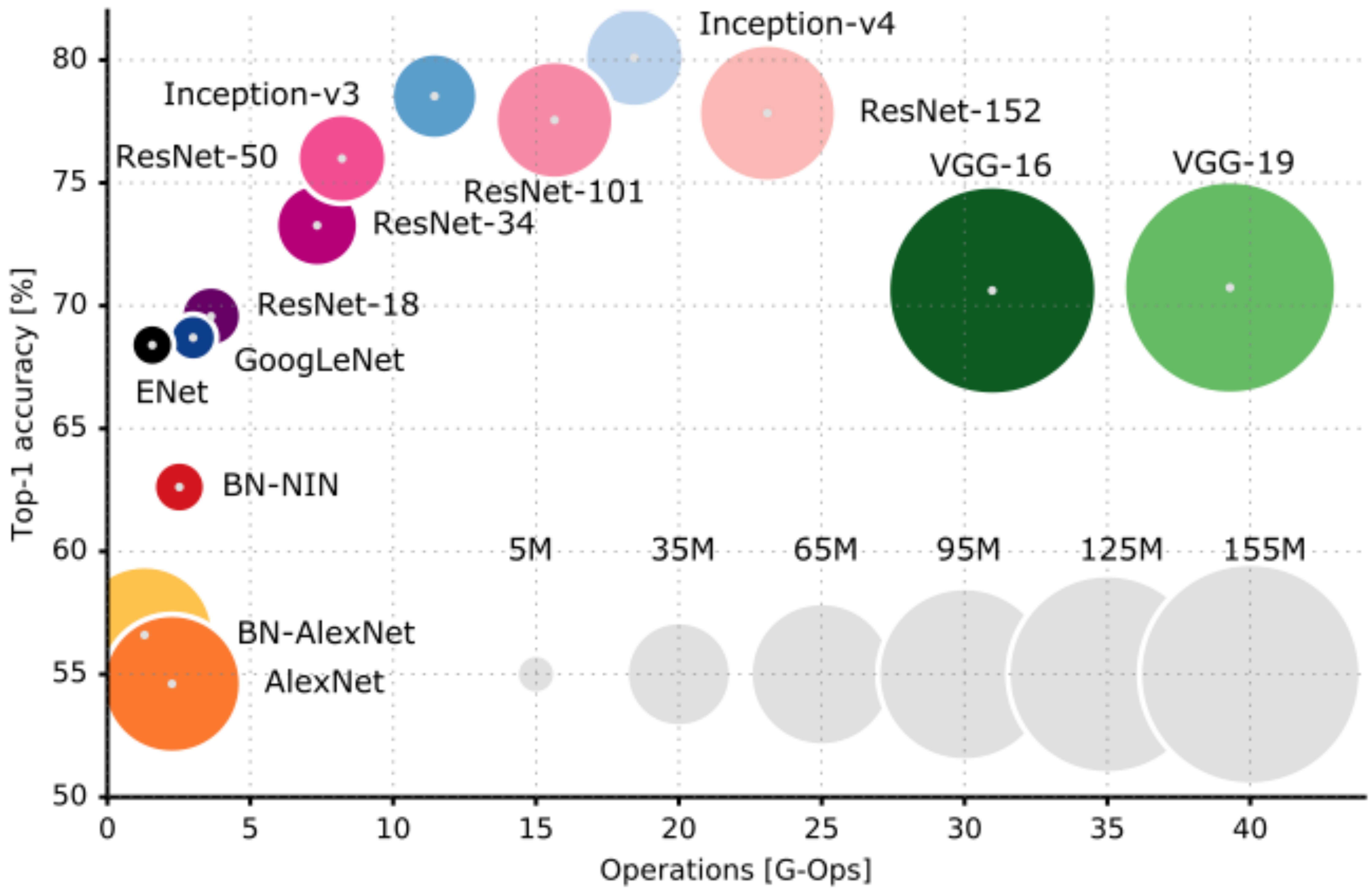
Neosperience

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

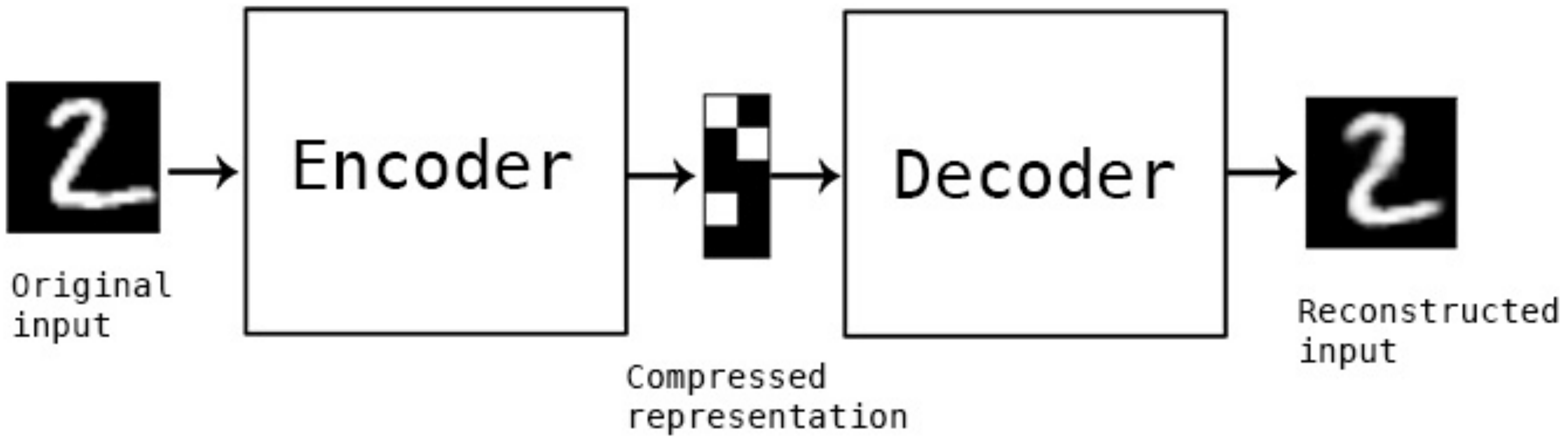


CNN Architectures: Complexity vs Accuracy



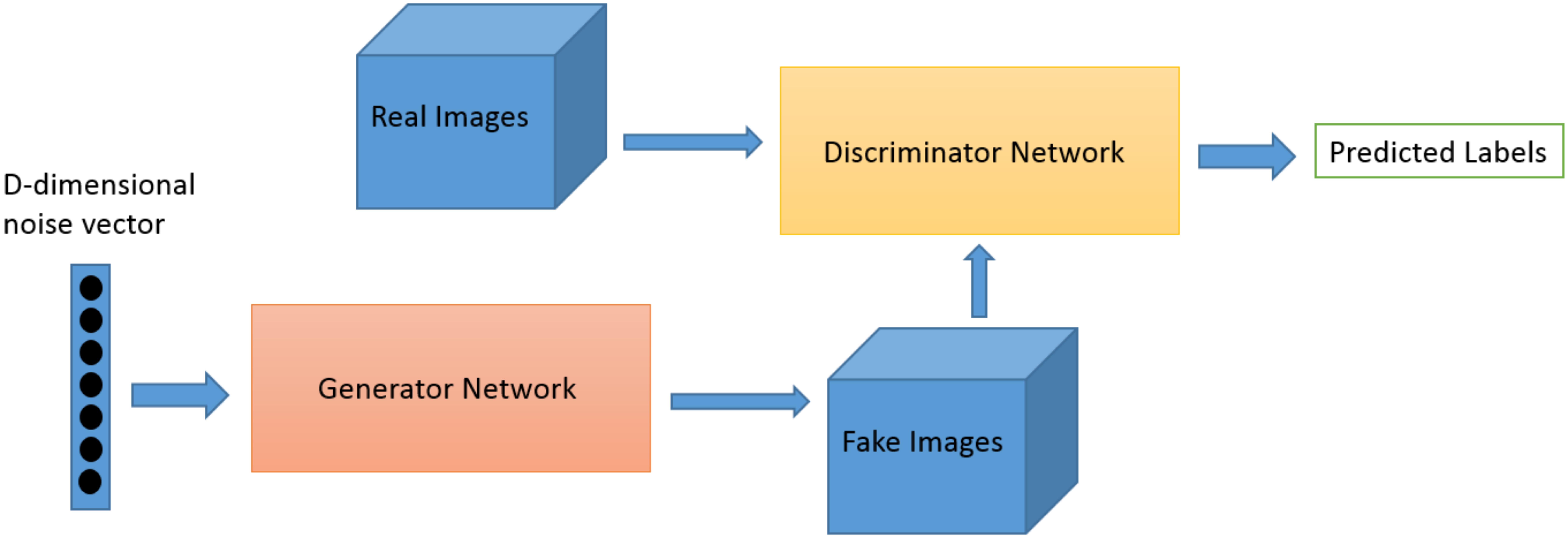
Autoencoders

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.



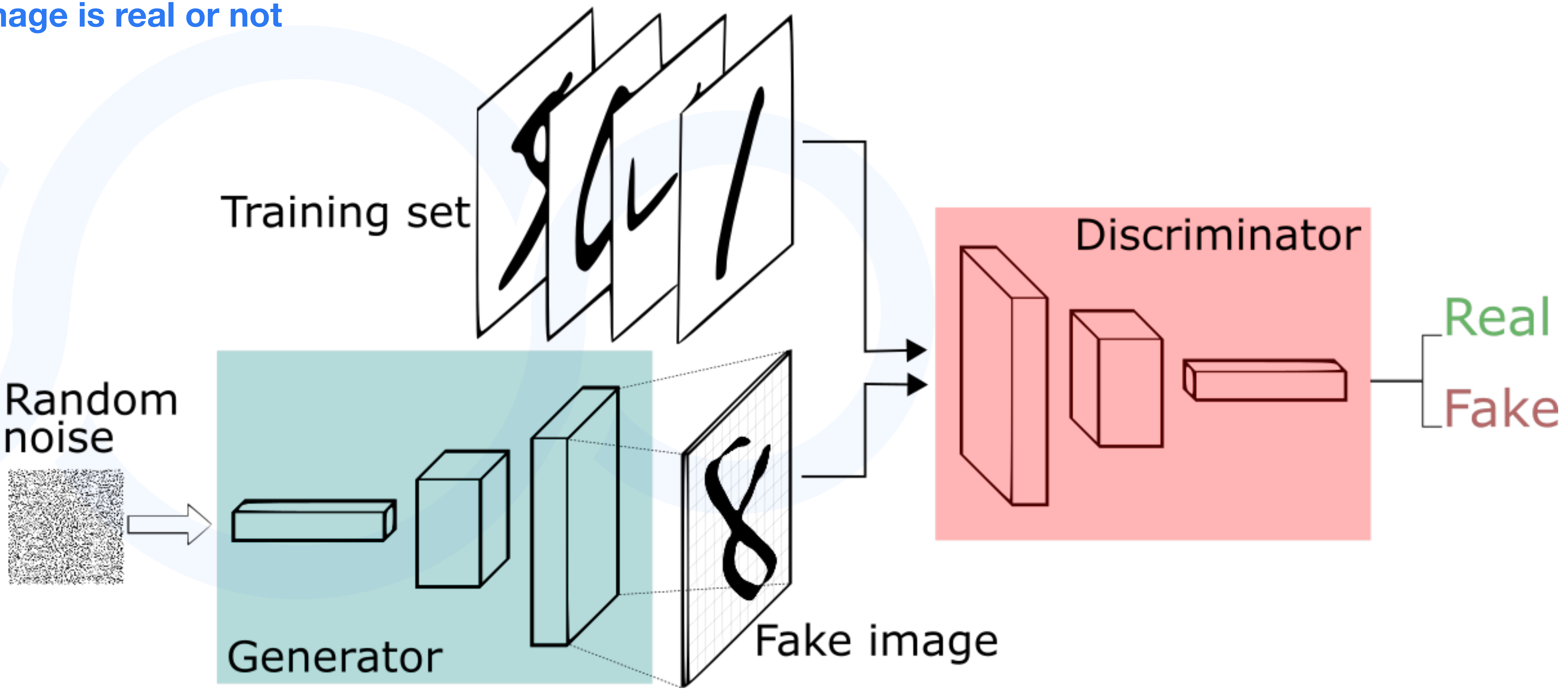
Generative Adversal Networks (GANs)

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



MNIST

Breed detector

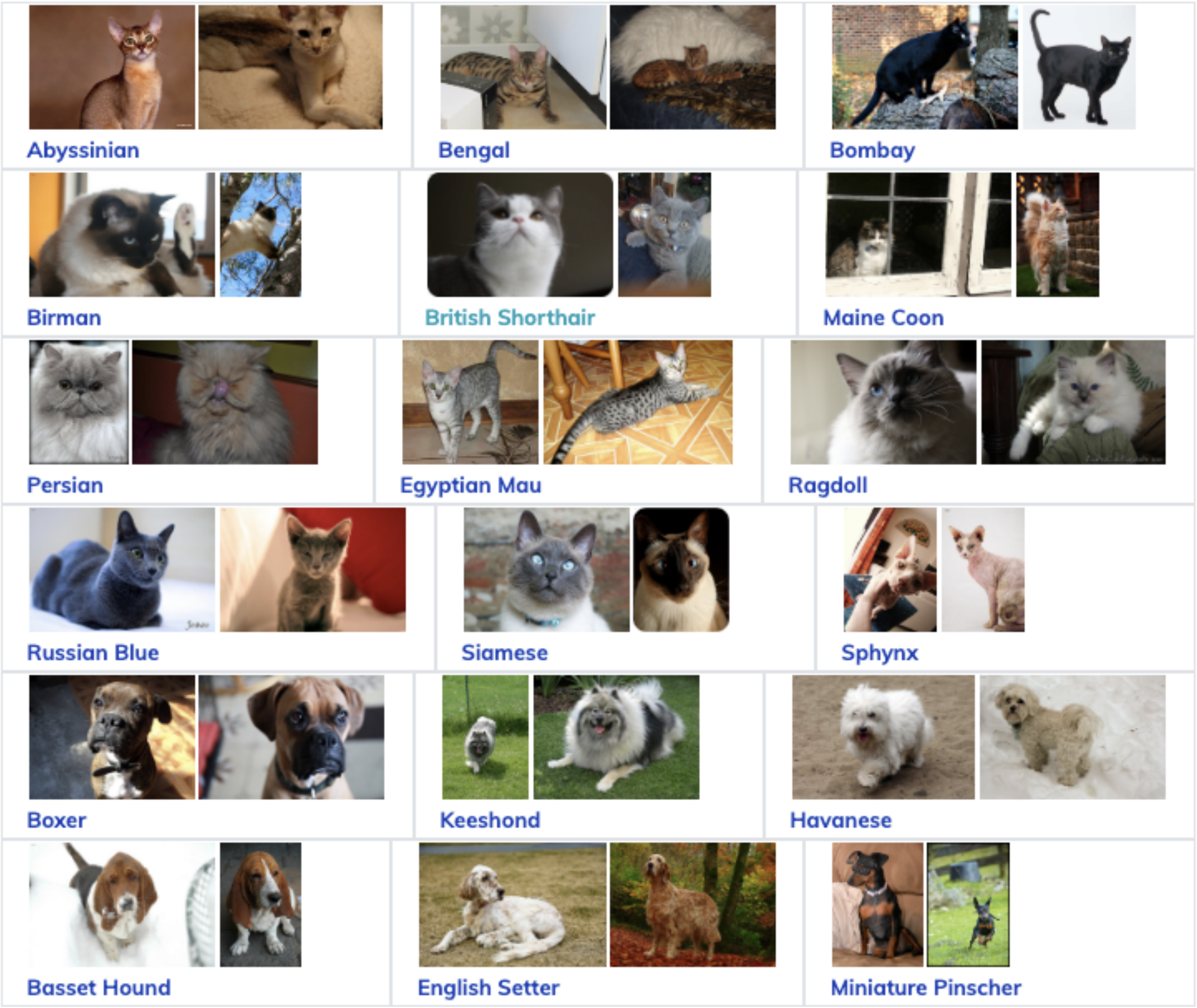
Problem: build a breed detector

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.

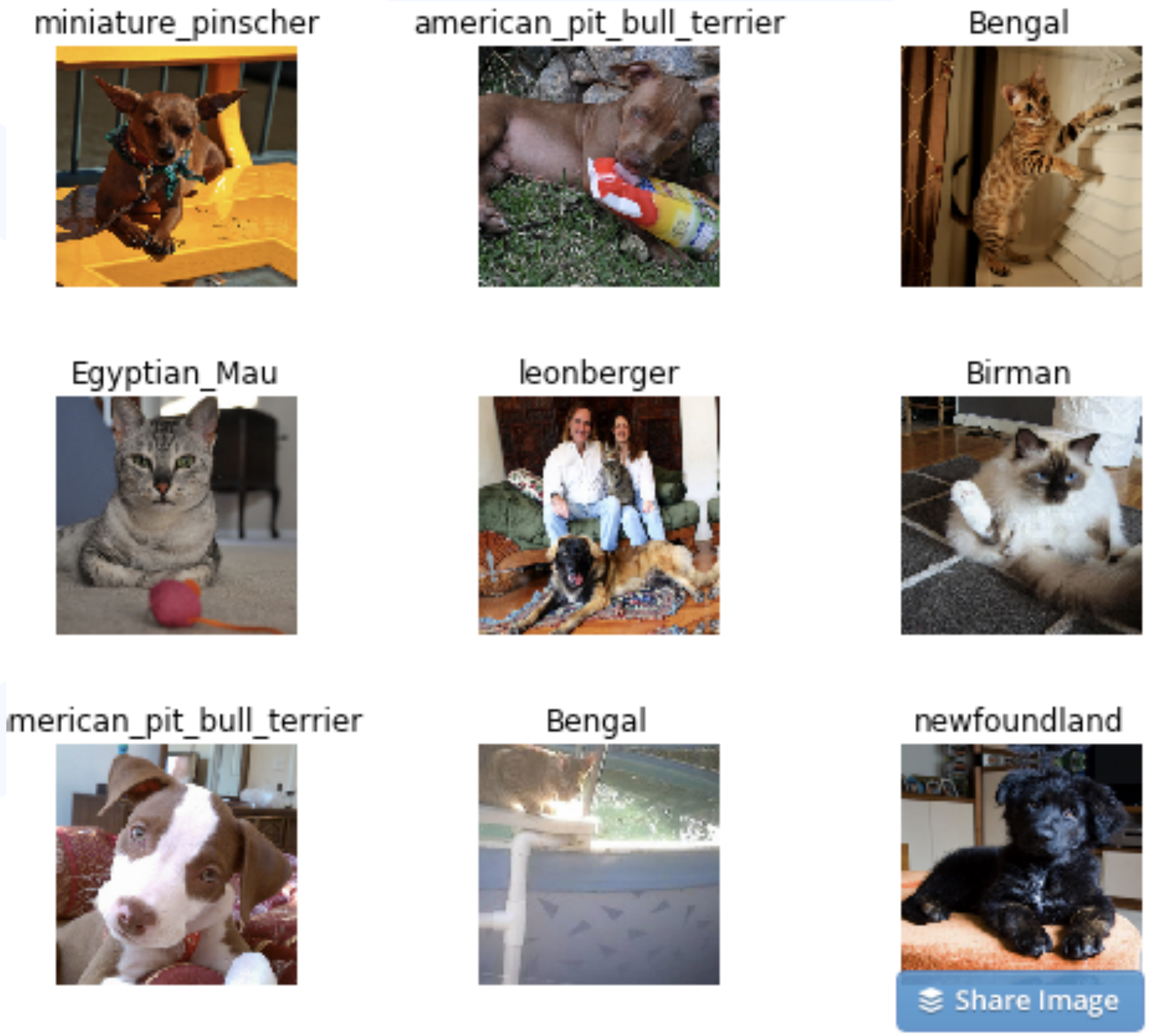
Since 2016 the computing power of GPUs makes this problem too naive to be used as benchmark, so we moved to detecting the breed of the pet in a picture

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



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.

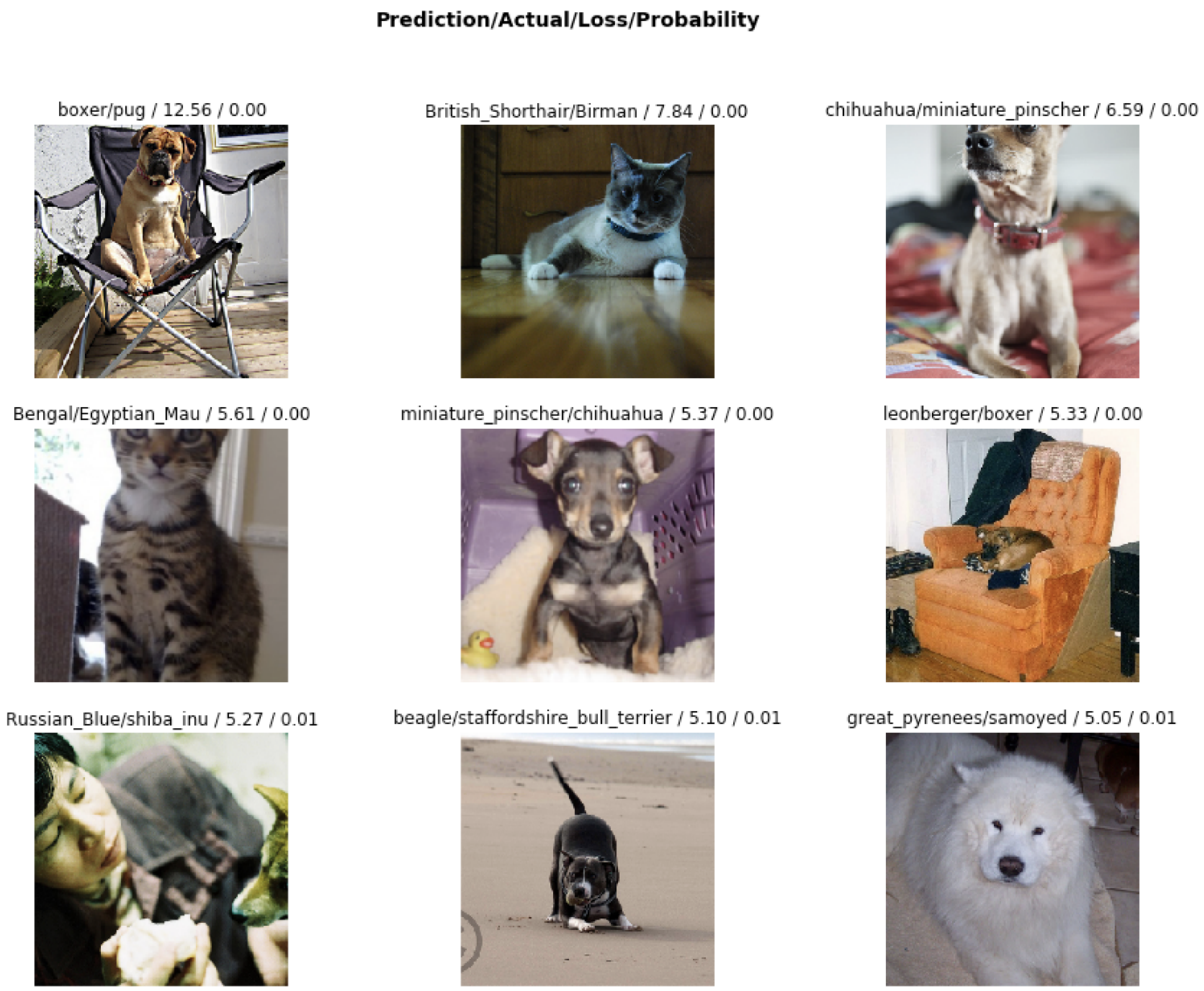
```
learn = cnn_learner(data, models.resnet34, metrics=error_rate)
```

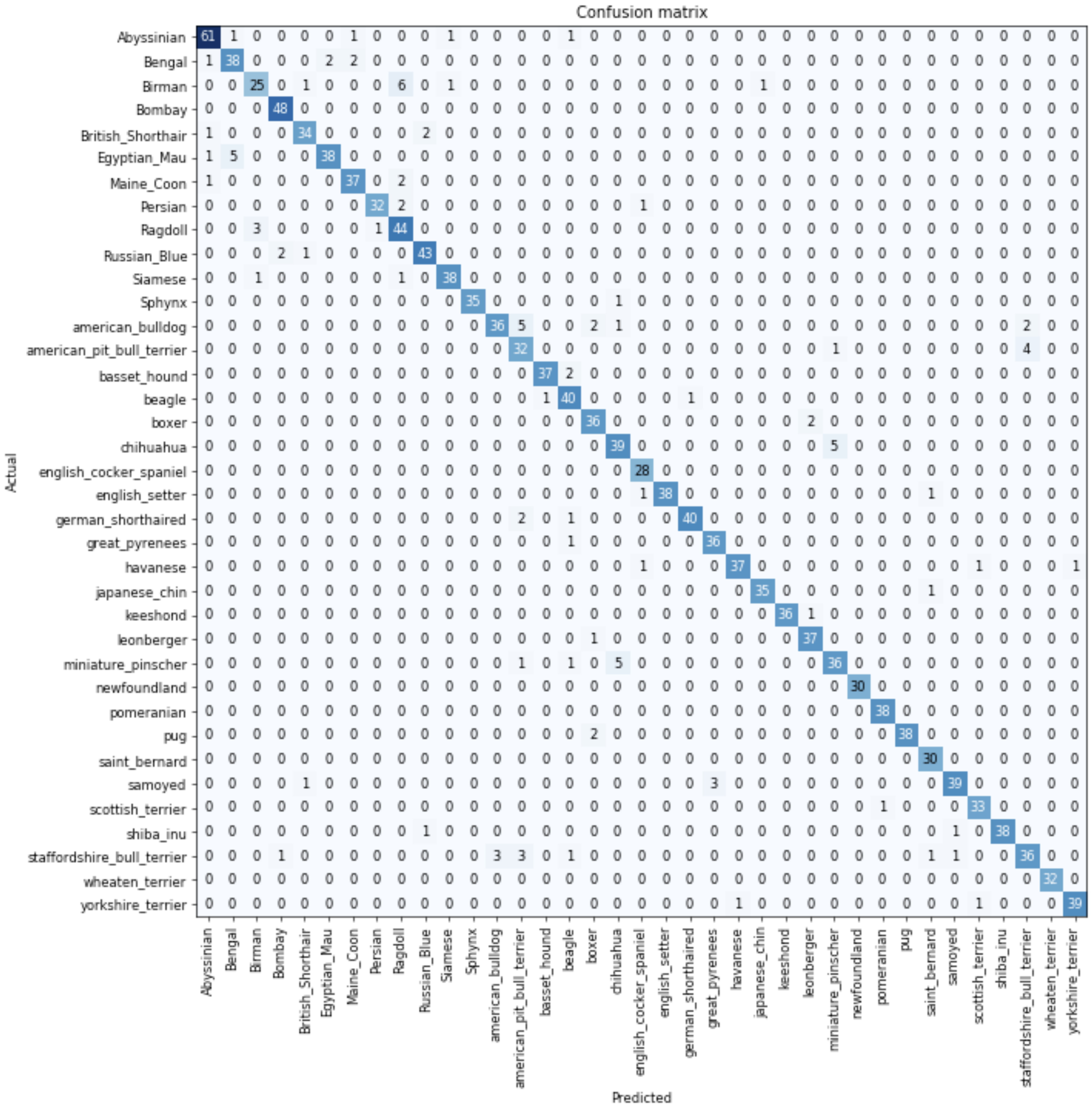
```
learn.fit_one_cycle(epocs)
```

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.





thank you.