ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE DE MADRID

Deep Learning

Álvaro Romo

Analytical Index

Convolutional Neural Networks (CNN)

a. Introduction to Computer Vision
b. Introduction to CNN
c. Transfer learning
d. Advanced use cases

2

Convolutional Neural Networks (CNN)

2.1

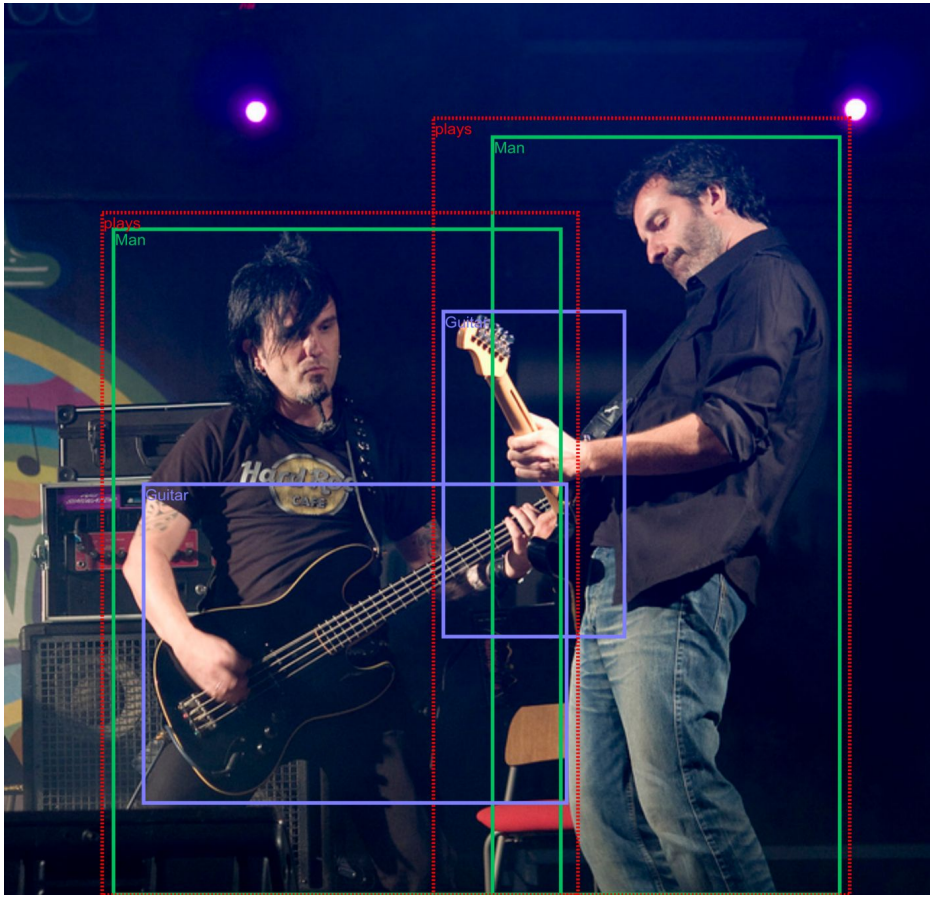
Introduction to Computer Vision

Computer Vision I

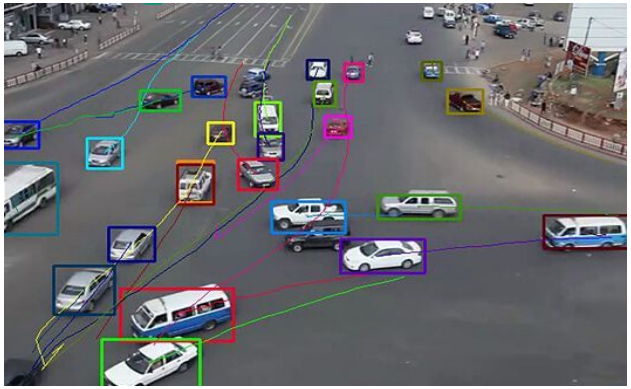
- Massive growth in visual data:
  - 500 hours of video are uploaded to YouTube every minute.
  - TikTok has over 1 billion monthly active users that spend 1.5 hours on average per day.
  - 100 million post and 1 billion stories are uploaded to Instagram every day.
  - There are cameras everywhere.
- Techniques and ideas can be used in other types of ordered data.
- People create and share visual data continuously.

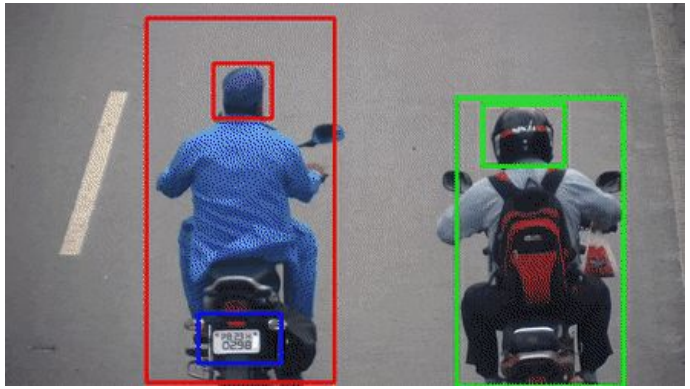
Computer Vision: Image Classification & Location

Computer Vision: Relationship Detection

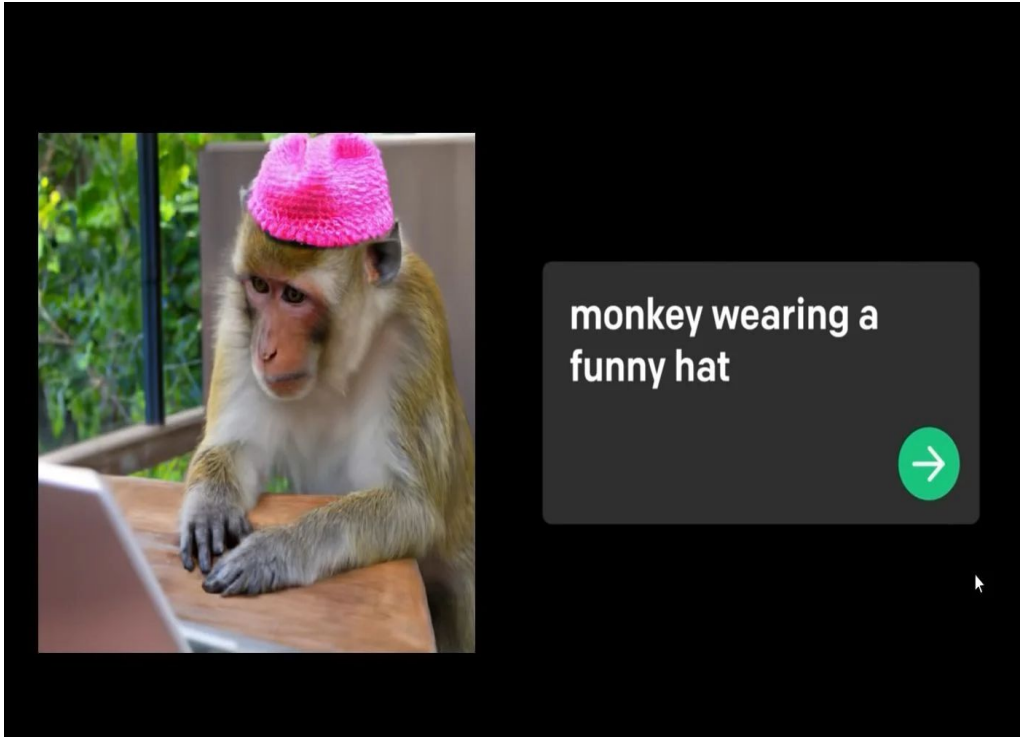


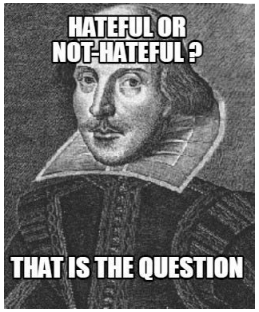
Computer Vision: ob

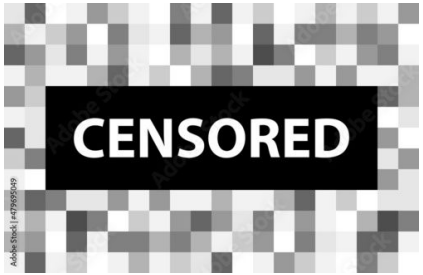




Computer Vision: Image annotation

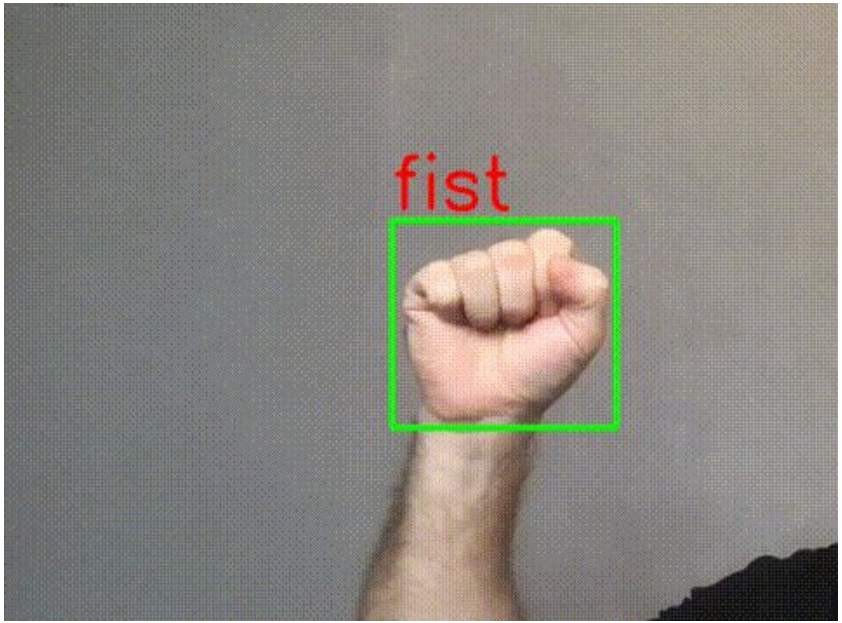




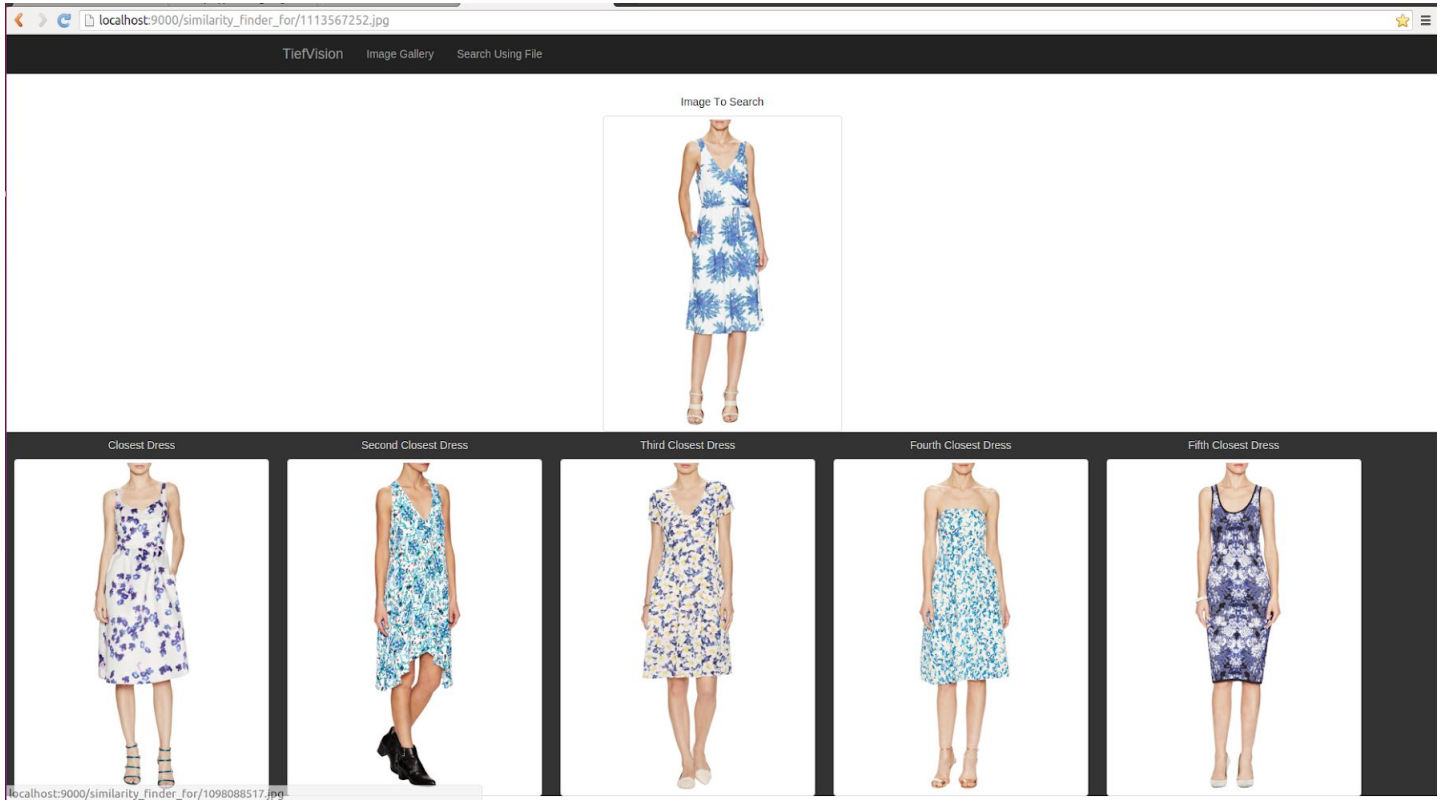


Computer Vision: Gesture Recognition

Interpreting human gestures to control devices or applications, used in gaming, healthcare, and smart home systems.

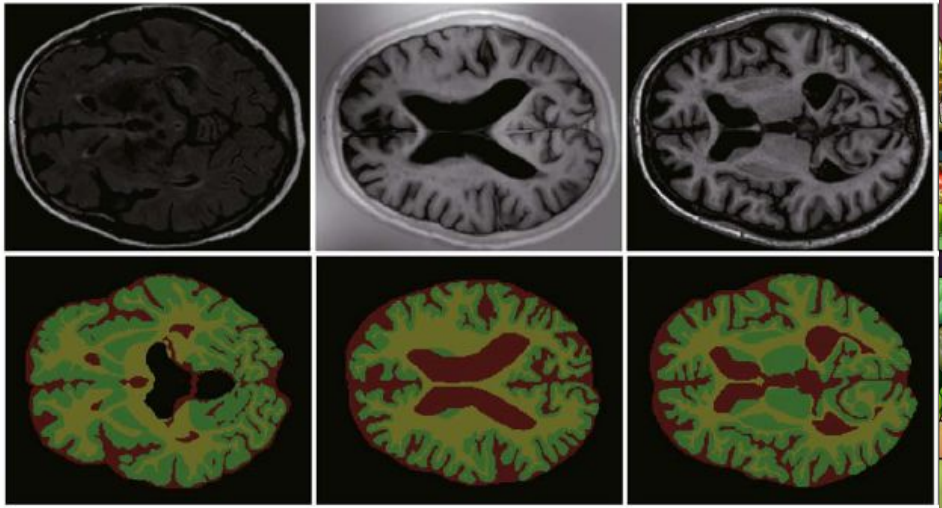


Computer Vision: Image Similarity Search Engine



https://github.com/paucarre/tiefvision

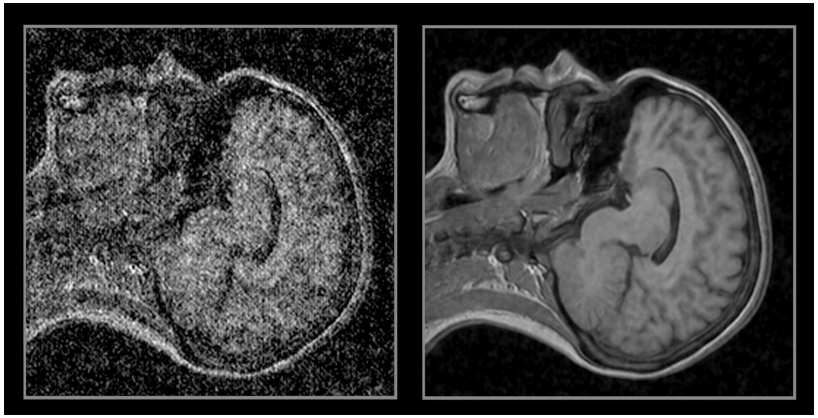
Computer Vision: Image segmentation



https://segment-anything.com/demo

Computer Vision: Image Reconstruction

Computer Vision: Image Denoising



Computer Vision: Image Generation

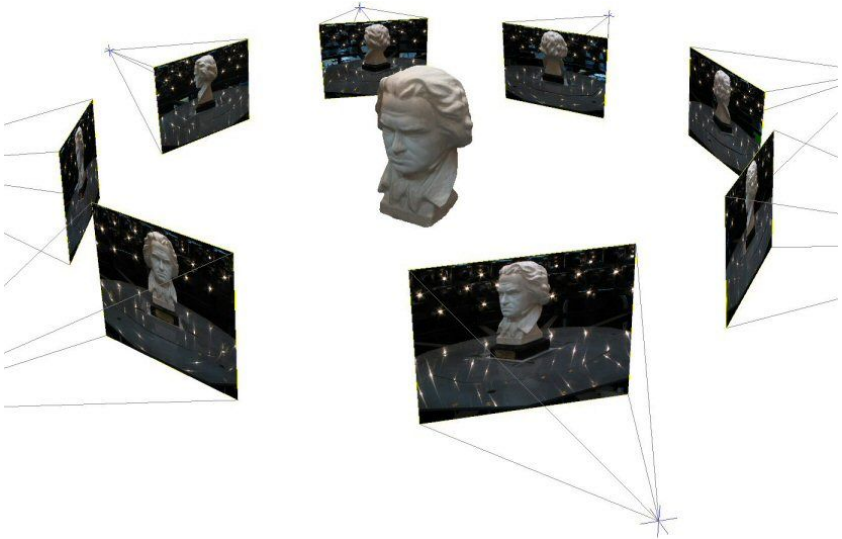
Computer Vision: Image Generation ad

https://emprendedor.com/coca-cola-primer-comercial-inteligencia-artificial-ia-generetiva-imagenes-stable-diffusion-masterpiece-obra-de-arte-video/

Computer Vision: social media

Computer Vision: 3D Reconstruction

Constructs 3D models from 2D images, used in virtual reality, architecture, and entertainment



Computer Vision: Super Resolution

Enhances the resolution of images, making them clearer, which is widely used in satellite imaging, medical imaging, and surveillance

Computer Vision: improve tool

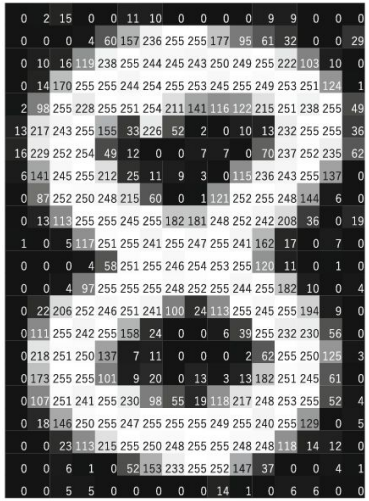
Computer Vision: assistant

CNN: image preprocessing

2.2

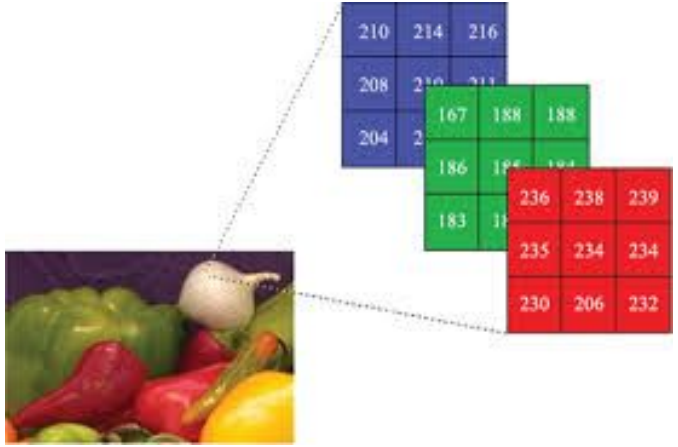
Introduction CNN

Image Data

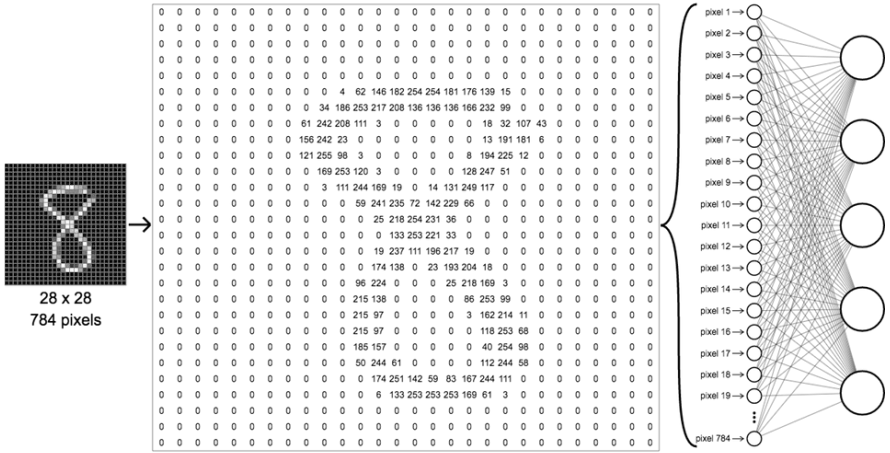


- Images are composed by pixels. 1 Megapixel can be a 1000x1000 matrix.
- Grayscale images can be seen as matrices of integers (0 -255) (black - white).

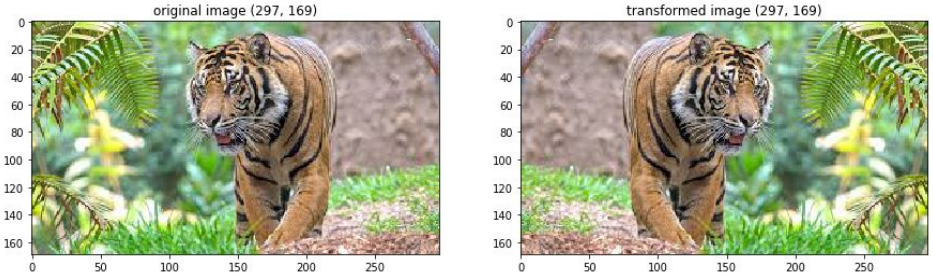
- Coloured images are usually represented as mixes of three colours: Red, Green and Blue (RGB)



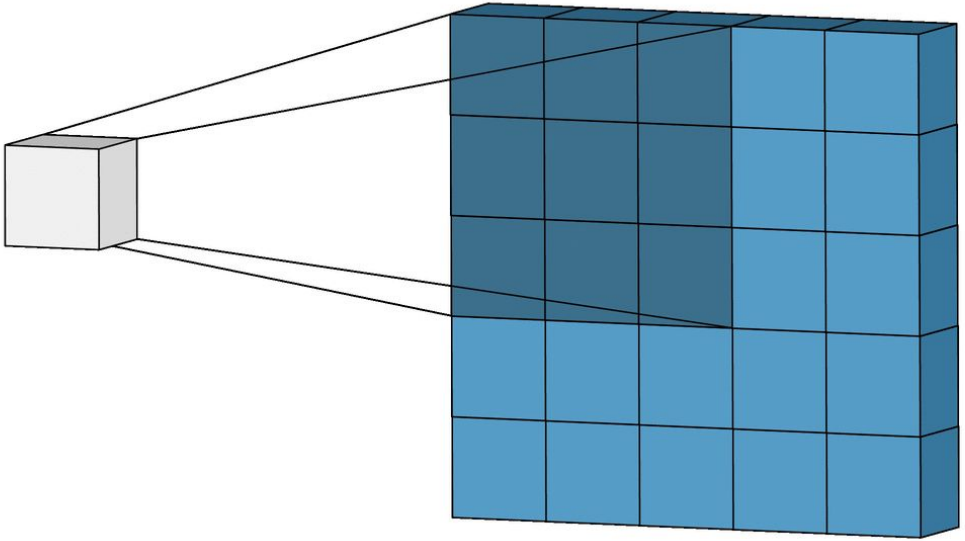
Problem with Dense Layers



- 1 Megapixel = 1000x1000 = 10^6 inputs.
- 1000 units hidden layer = 10^9 parameters!!
- No spatial information. Collapse 2D => 1D
- In most tasks, you can share the same operations in different parts of the image, In a fully-connected layer you learn pixel-by-pixel operations.

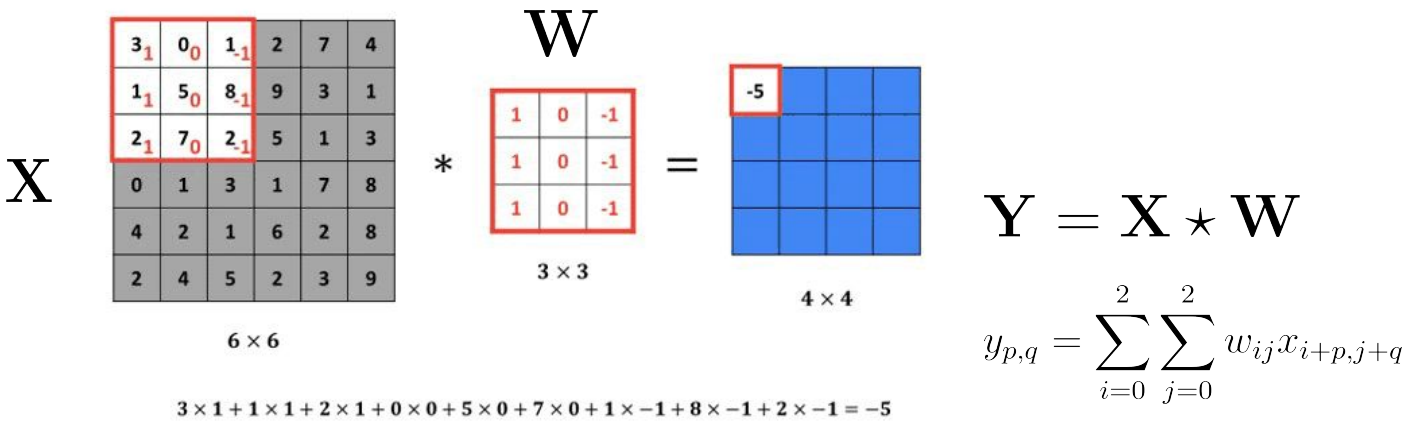


Convolution Operation: Intuition



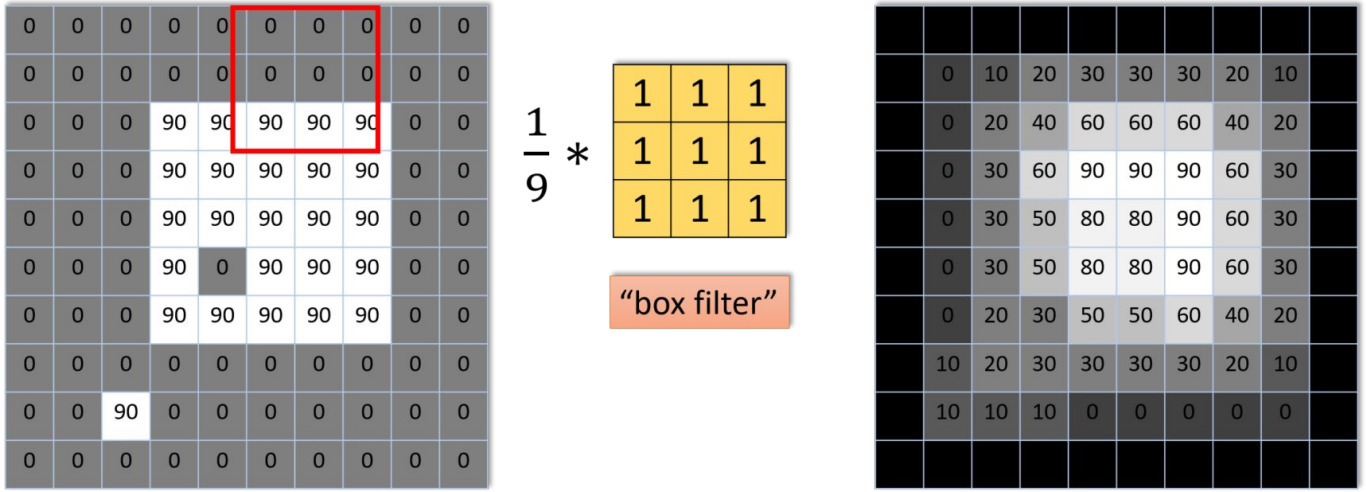
- Connect input patch to a single neuron to capture spatial information
- Use sliding window to order connections.
- 3x3 filter, stride = 1 (shift by 1 pixel the sliding window)

Convolution Operation: Filtering



- In mathematics it is called Cross-correlation, In Deep learning convolution.
- Operator: *
- Dimensions: $(n \times n) \star (f \times f) \equiv (n - f + 1) \times (n - f + 1)$

Convolution Operation: Example



Convolution Operation: Example

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

6 x 6

*

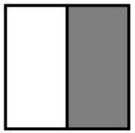
1	0	-1
1	0	-1
1	0	-1

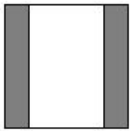
3 x 3

=

-0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

4 x 4





Convolution Operation: Example

255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255
255	255	255	255	255	255
10	10	10	10	10	10
10	10	10	10	10	10
10	10	10	10	10	10
10	10	10	10	10	10

8 X 6

*

1	1	1
0	0	0
-1	-1	-1

3 X 3

=

6 X 4

0	0	0	0
0	0	0	0
735	735	735	735
735	735	735	735
0	0	0	0
0	0	0	0

Convolution Operation: Example

35	40	41	45	50
40	40	42	46	52
42	46	50	55	55
48	52	56	58	60
56	60	65	70	75

×

0	1	0
0	0	0
0	0	0

=

		42		

Edge Detection

0	1	0	
1	-4	1	
0	1	0	

Edge Detection (right)

0	0	0	
-1	1	0	
0	0	0	

Convolution Operation: Example

Sharpen

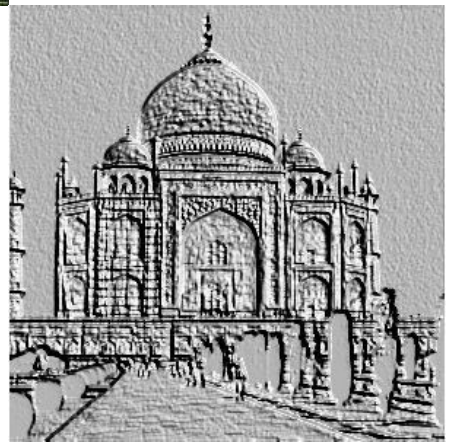
0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

Blur

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

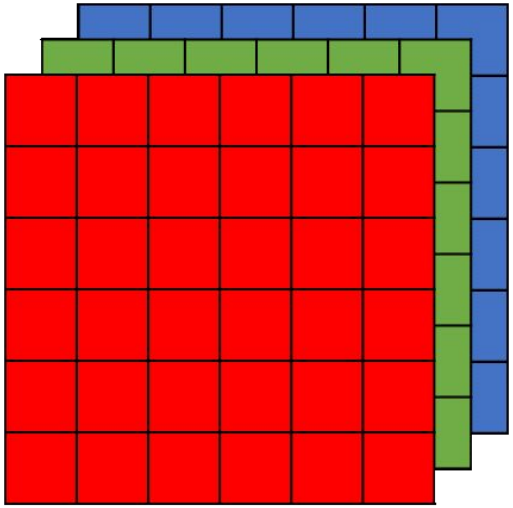
Emboss

-2	-1	0	
-1	1	1	
0	1	2	



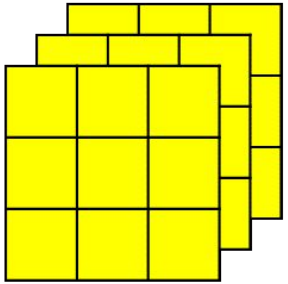
Convolution Operation: RGB

3 separate convolutions and then sum all 3



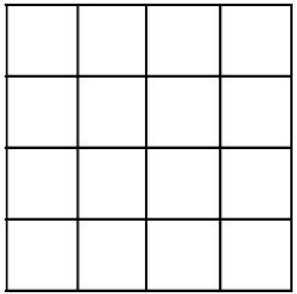
$6 \times 6 \times 3$

*



$3 \times 3 \times 3$

=



4×4

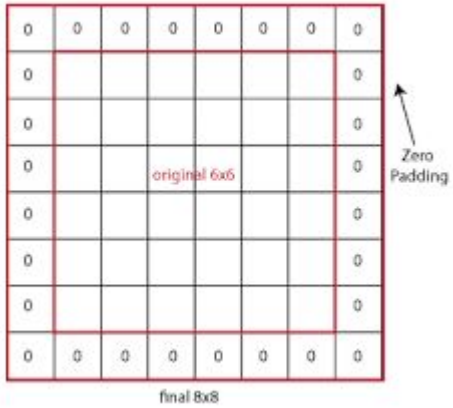
Convolution: Padding & Stride

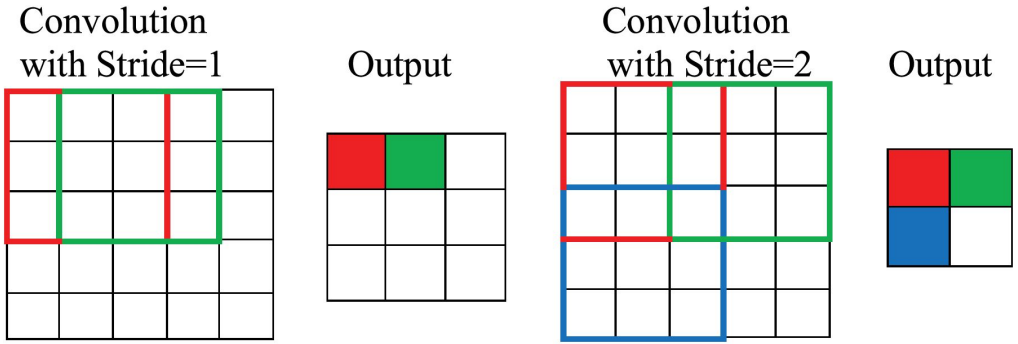
- The image shrinks every time a convolution operation is perform $(n \times n) * (f \times f) = (n - f + 1) \times (n - f + 1)$
- Padding (p): amount of zero-pixels added to an image.

$$(n - f + 2p + 1) \times (n - f + 2p + 1)$$

- Types convolution:
  - **“valid”**: No padding (p =0)
  - **“same”**: Output size = Input size

$$p = \frac{f - 1}{2}$$





- Stride: (s) number of rows and columns traversed per slide

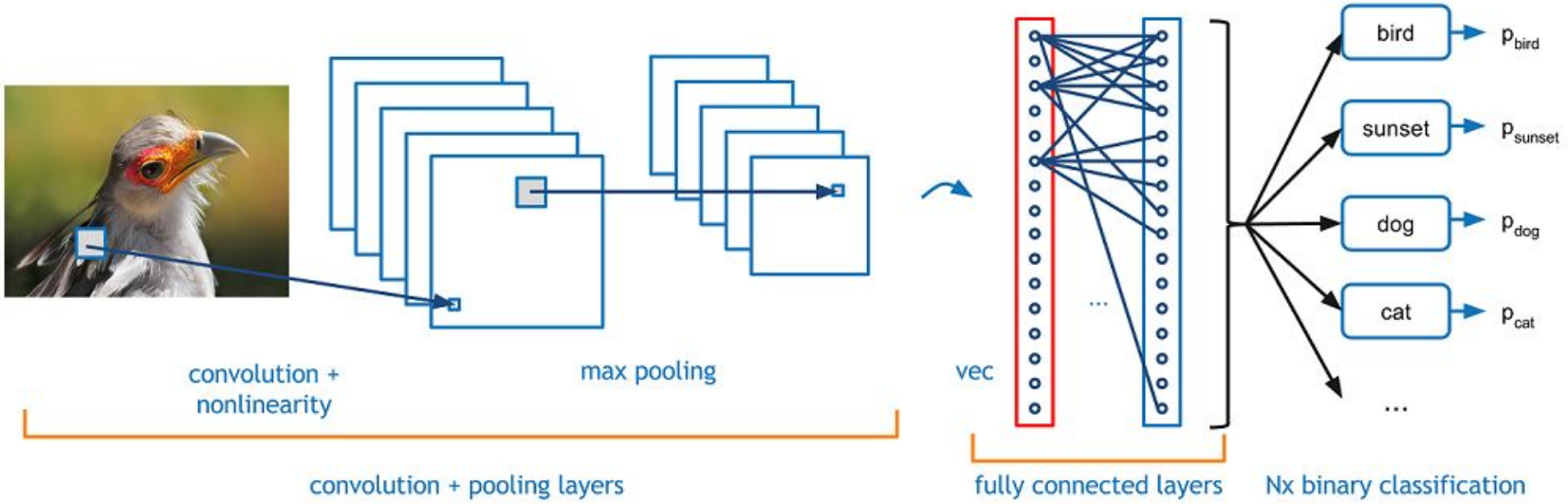
$$\left\lfloor \frac{n - f + 2p}{s} + 1 \right\rfloor$$

CNNs Advantages

- **Sparse Connections:** we need to store fewer parameters, which both reduces the memory requirements of the model and improves its statistical efficiency.

- **Shareable Weights:** use the same parameter for more than one function in a model. Rather than learning a separate set of parameters for every location, we learn only one set. **Equivariance to translation**, if the input changes, the output changes in the same way.

CNN Layers



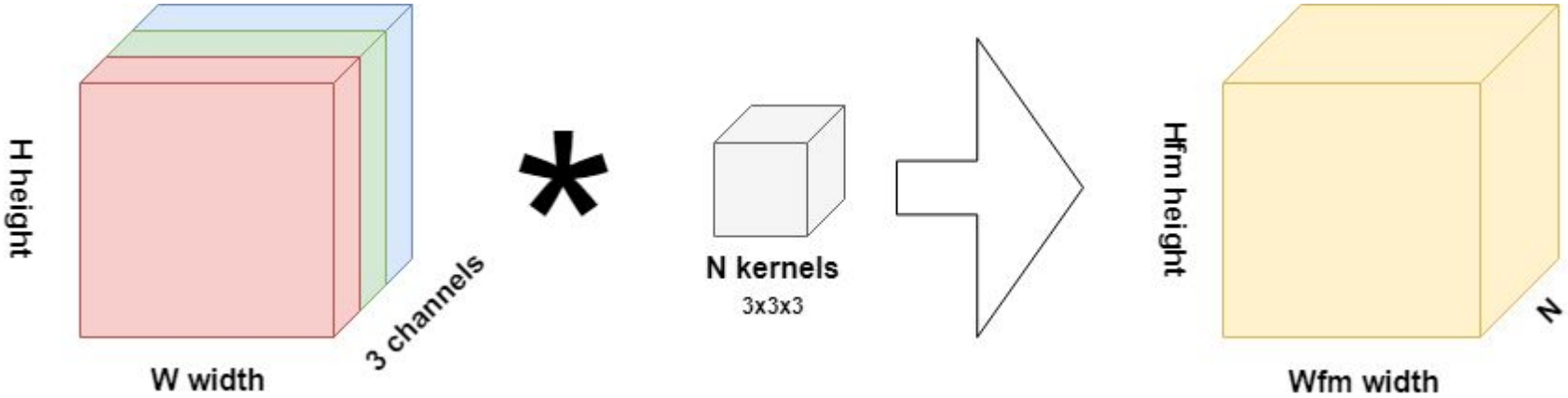
- Convolution + non-linearity (ReLu)
- Pooling: Downsampling
- Fully-connected

```
tf.keras.layers.Conv2D
```

```
tf.keras.layers.MaxPool2D
```

```
tf.keras.layers.Dense
```

CNN Layers: Convolution

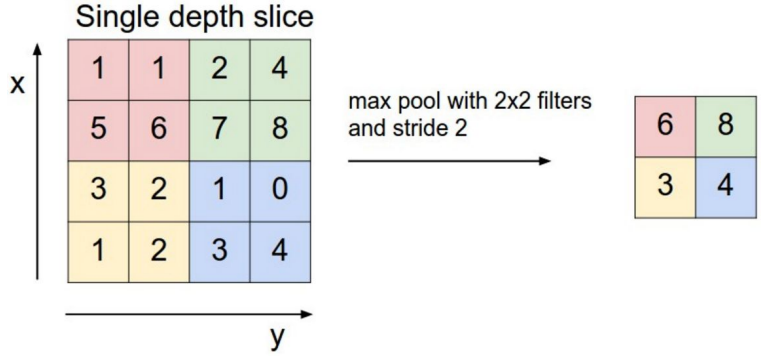


Layer dimensions: (h x w x N) Relu(convolution + b) for every filter (N different filters)

```
tf.keras.layers.Conv2D(
    filters=N,
    kernel_size=(h,w),
    strides=(1, 1),
    padding="valid"
)
```

CNN Layers: Pooling

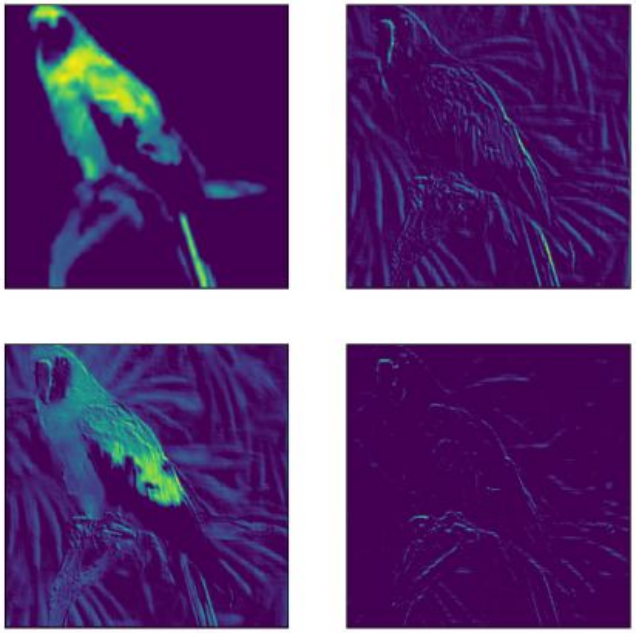
- **Dimensionality reduction:** representations turn smaller and more manageable.
- Reduces the number of parameters contributing to a smaller complexity.
- **Spatial invariance.**
- MaxPooling and average pooling are the most common.
- Backprop is not affected. **No parameters to tune**.



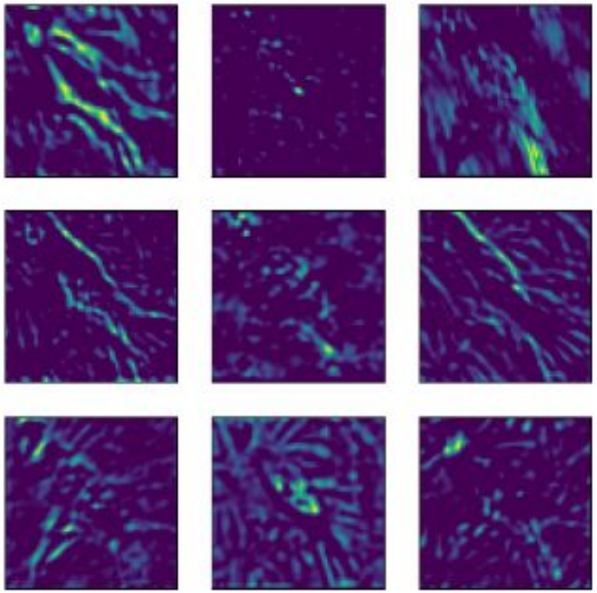

```
tf.keras.layers.MaxPooling2D(
    pool_size=(2, 2), strides=(1,1)
)
```

```
tf.keras.layers.AveragePooling2D(
    pool_size=(2, 2), strides=(1,1)
)
```

CNN Features



Layer 1



Layer 3

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

CNN: Data Augmentation

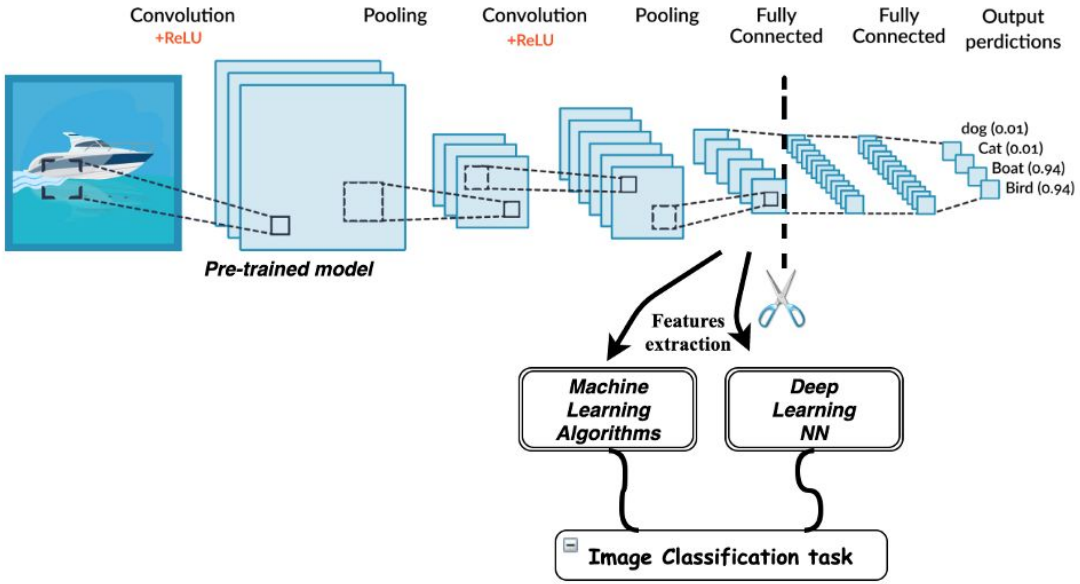
- Flip
- Rotation
- Scale
- Translation
- Crop
- Gaussian Noise
- ...

CNN: introduction to CNN + XAI

2.3

Transfer learning

CNN: Transfer learning

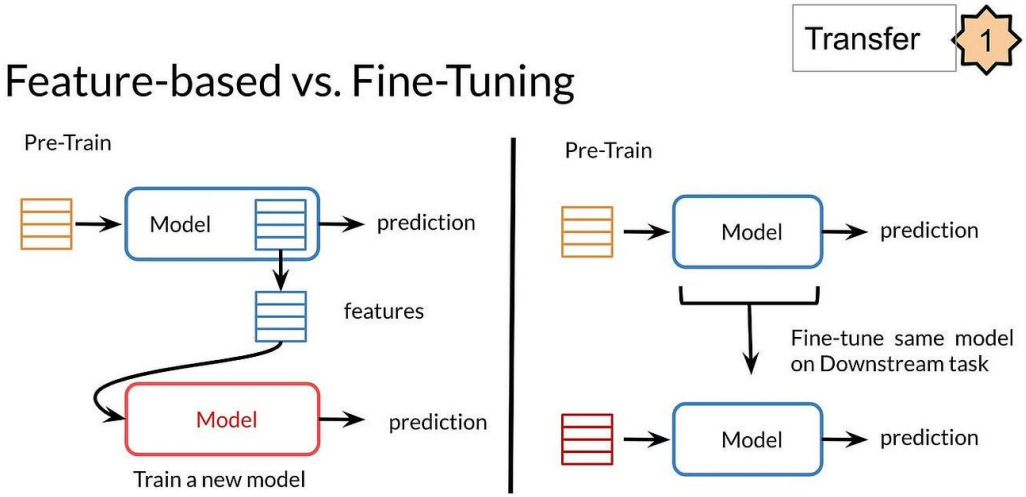


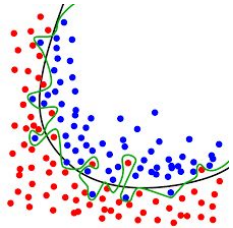
Instead of training a Deep Neural Net from scratch for your task:

- Take a network trained on a different domain for a different task
- Adapt it for your domain and your task

CNN: Feature-based vs finetuning

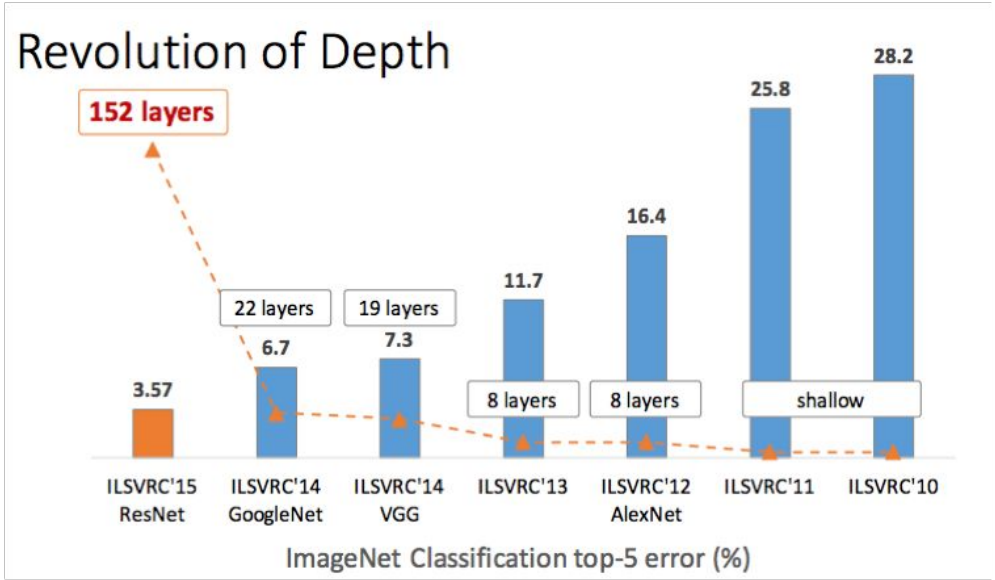
- **Feature-based**: the weights of the pre-trained model are not updated (freeze weights). A new model is trained by taking the model output as input.
- **Fine-tuning**: we update the weights of the model by changing the last layers.





CNN: transfer learning

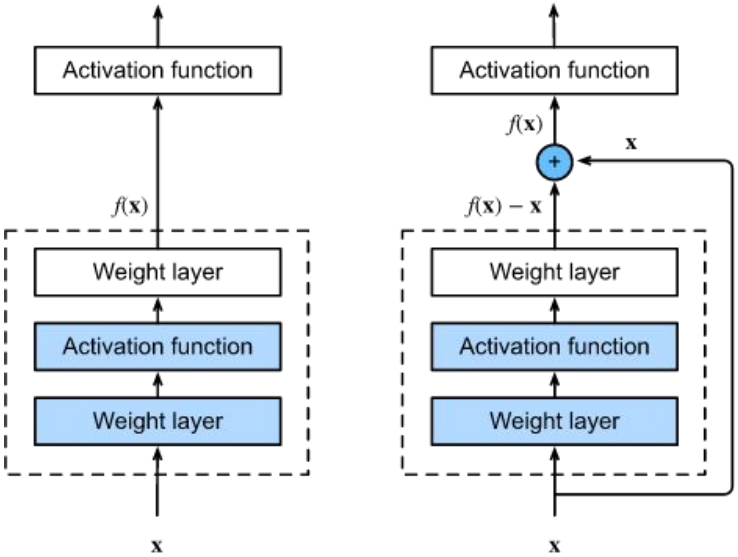
Modern Architectures: Revolution on Depth



Human Beings : Top-5 Error Rate – 5.1%

Source:
https://medium.com/@Lidinwise/the-revolution-of-depth-facf174924f5

Modern Architectures: ResNet



Source: d2l.ai

- Increasing network depth does not work by simply stacking layers together.
- You can skip the training of few layers using skip-connections or residual connections.
- The residual block can learn the identity function more easily.

Modern Architectures: pretrained models

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

```
from tensorflow.keras.applications.resnet50
import ResNet50

model = ResNet50()
model.summary()
```

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-
EfficientNetB0	29 MB	-	-	5,330,571	-

TensorFlow Hub

https://tfhub.dev/

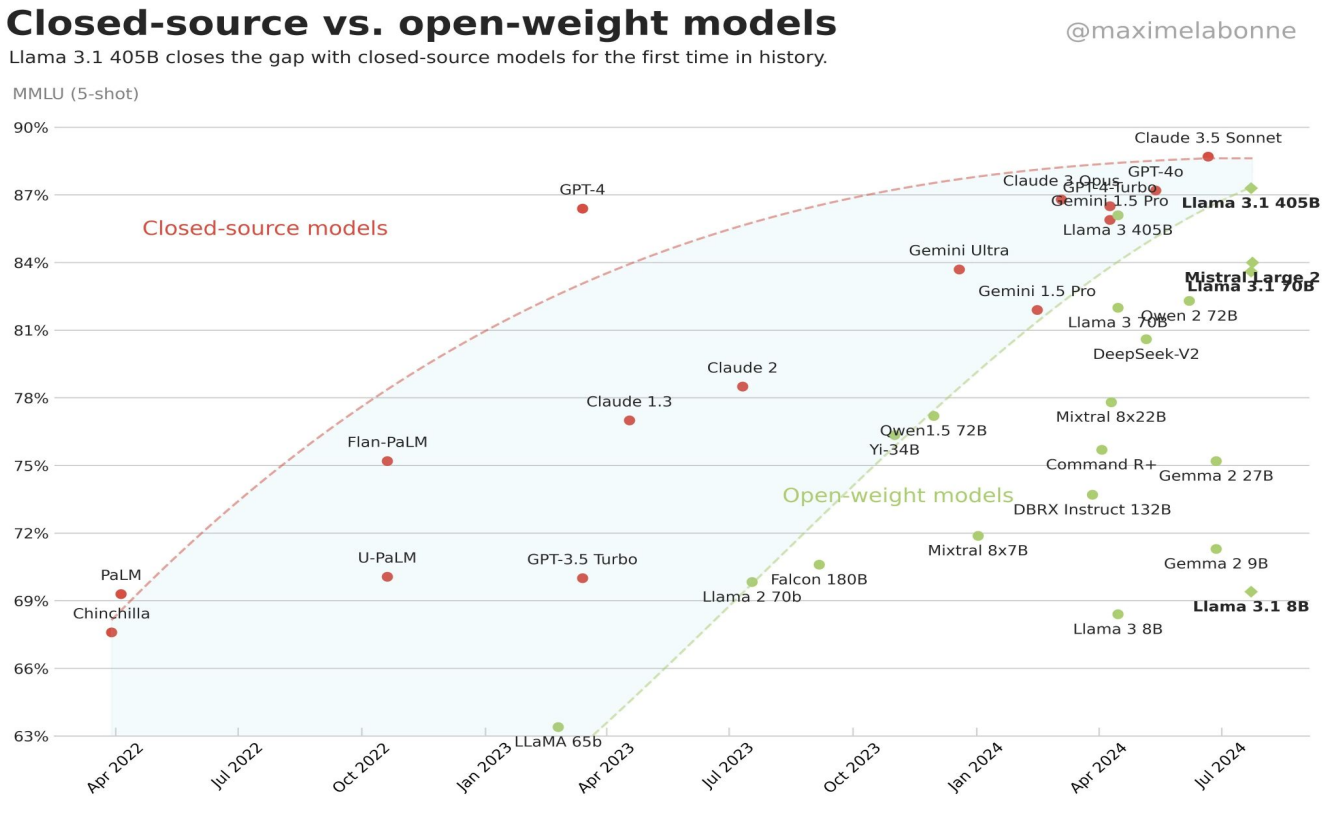
huggingface_hub

https://huggingface.co/models

Comparatives open source y close source

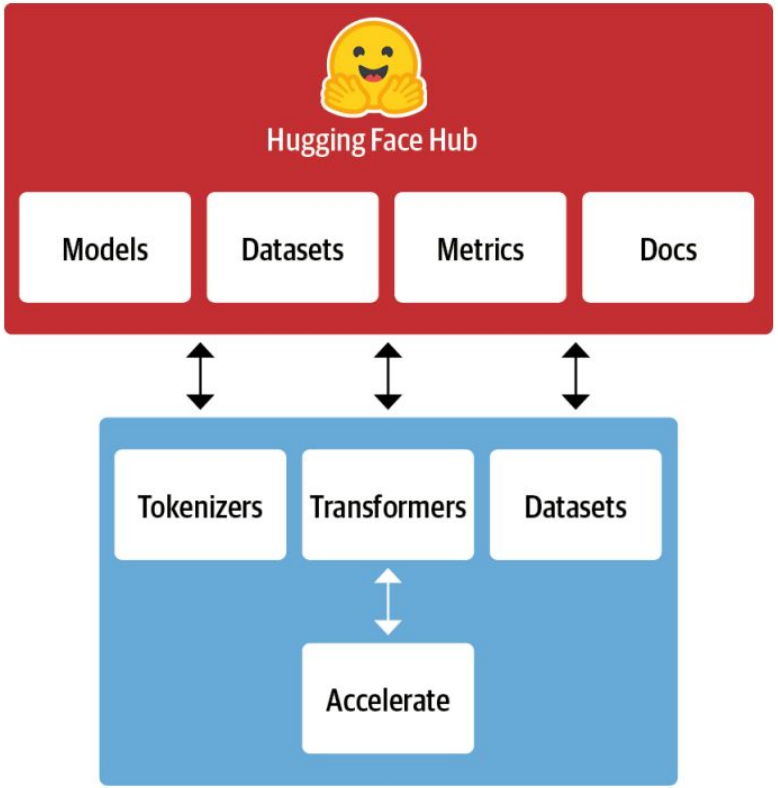
	Open source	Close source
Costs	Low initial cost but higher maintenance	Higher cost but less maintenance
Innovation	Collaborative community with rapid innovation	Slower innovation subject to a single entity
Support	Community-based and may be inconsistent	Consistent and administered by the entities through official channels
Adaptation	Highly customizable with access to model code and weights	Limited adaptation to platform limits
Performance	High performance in tailored domains, poorer performance in generic domains	High performance in generic domains, poorer performance in specific domains

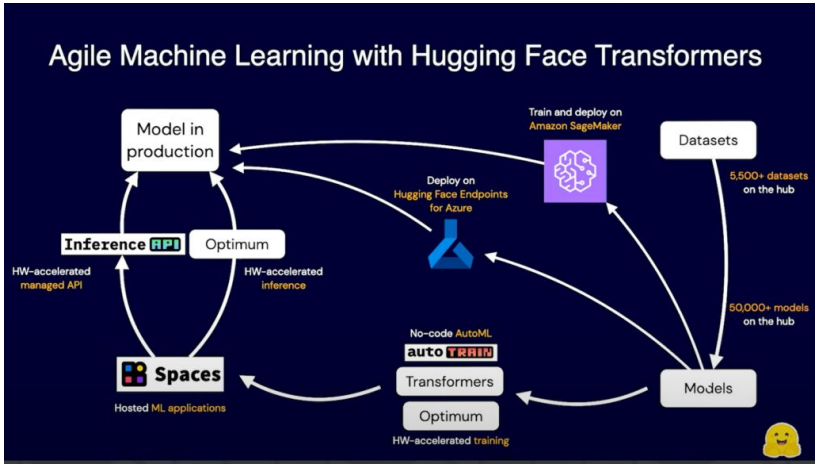
Performance of historical models



CNN: hugging face

Introduction to Hugging Face





Walkthrough → `huggingface_hub`

https://huggingface.co/

Introduction to Hugging Face

Spaces

hf.co/spaces

https://huggingface.co/spaces/akhaliq/ArcaneGAN

https://huggingface.co/spaces/google/sdxl

https://huggingface.co/spaces/prodia/fast-stable-diffusion

https://huggingface.co/spaces/huggingface-projects/QR-code-AI-art-generator

https://huggingface.co/spaces/radames/Enhance-This-DemoFusion-SDXL

https://huggingface.co/spaces/InstantX/InstantID

CNN: Hugging Face and CV models

2.3

Advanced Use Cases:

Image Classification Advances

Multi-class classification

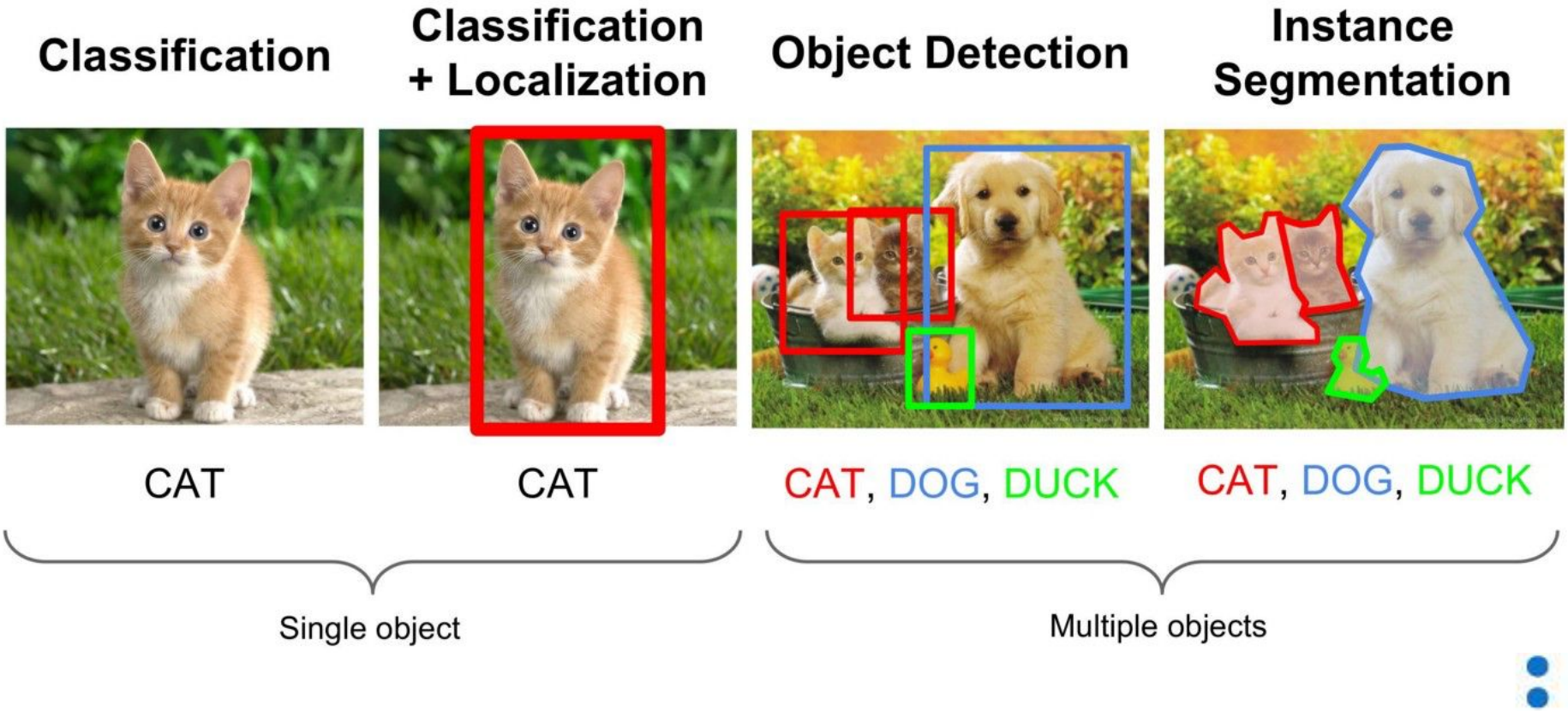
Dog

Cat

Multi-Label classification

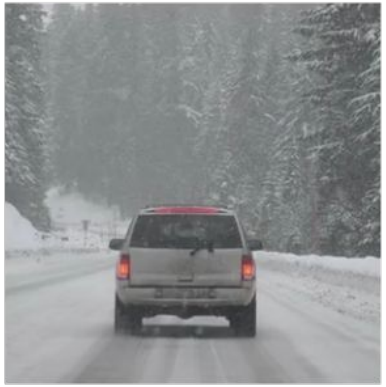
Dog, cat

Image Classification Advances



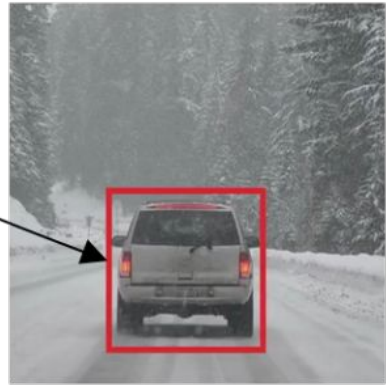
Object Localization I

Image classification



"Car"

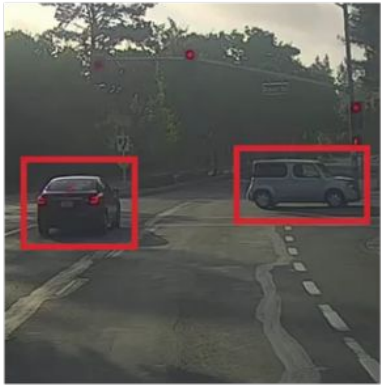
Classification with localization



"Car"

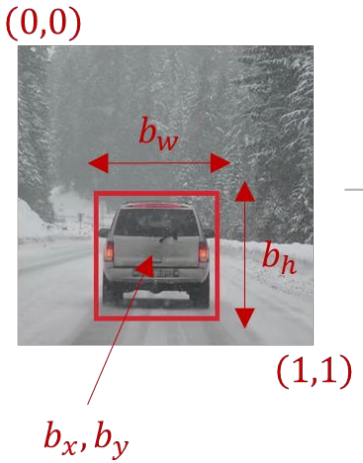
One object

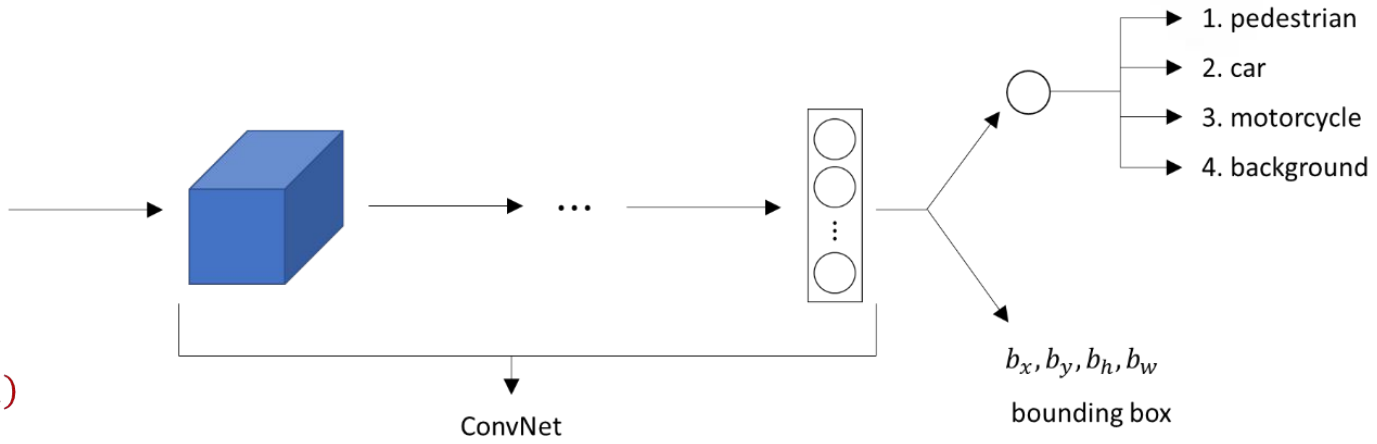
Detection



Multiple objects

Object Localization II





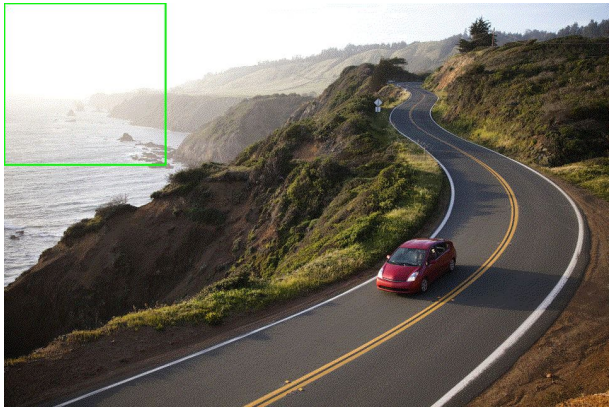
Target: $y = \begin{bmatrix} c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$

Sigmoid: Is there an object?

Bounding box, ReLu

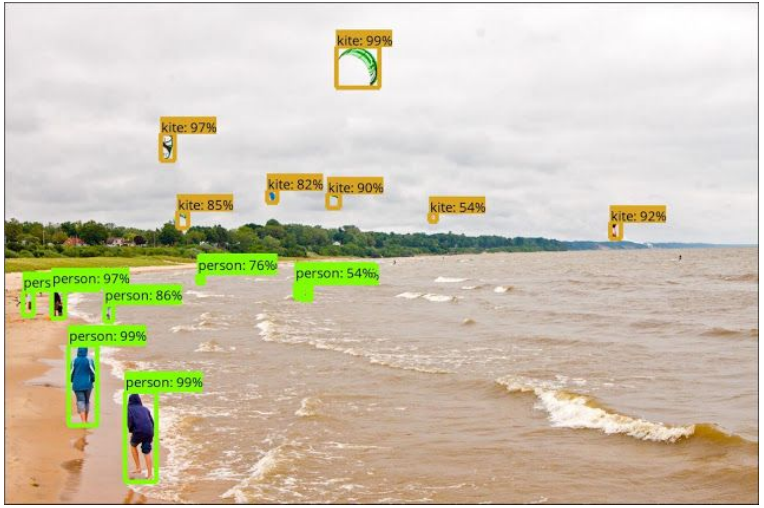
Classes of objects, softmax

Object Detection: Sliding Windows



Object Detection: Popular Algorithms

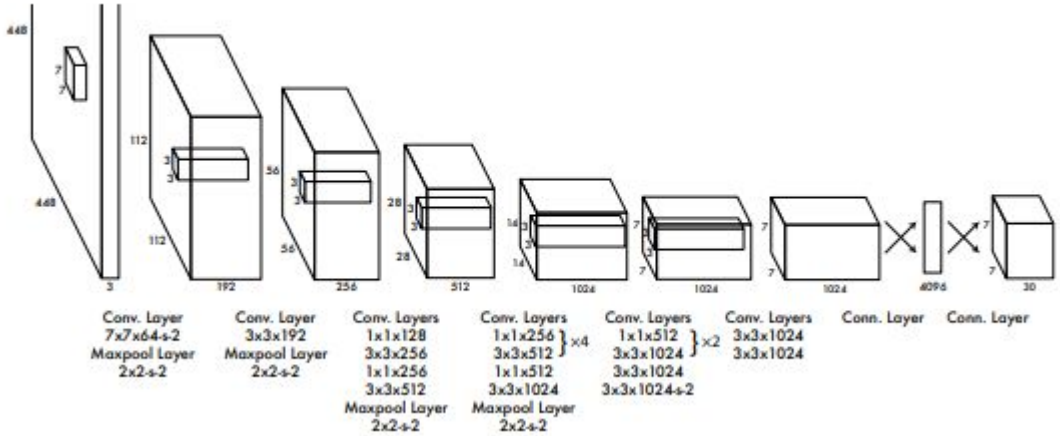
- R-CNN
- Faster -RCNN
- YOLO
- SSD



YOLO: You Only Look Once

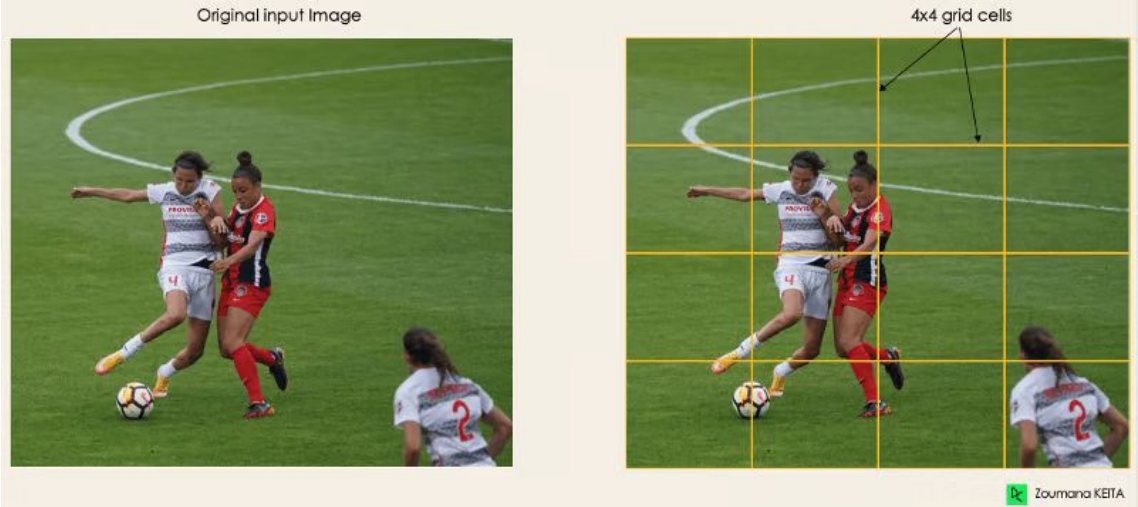
Steps:

- Residual blocks
- Bounding box regression
- Intersection Over Unions or IOU for short
- Non-Maximum Suppression.



[Original paper]

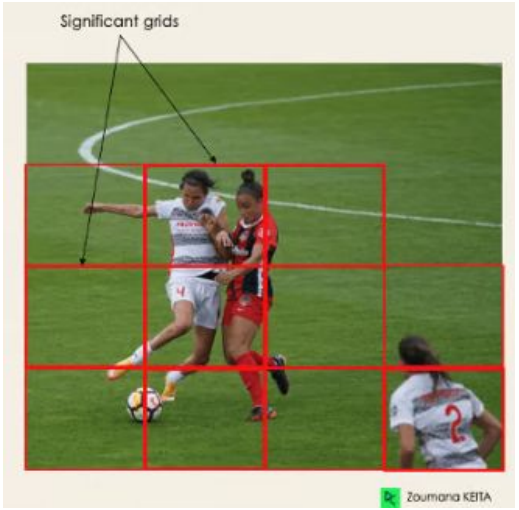
YOLO: Residual blocks

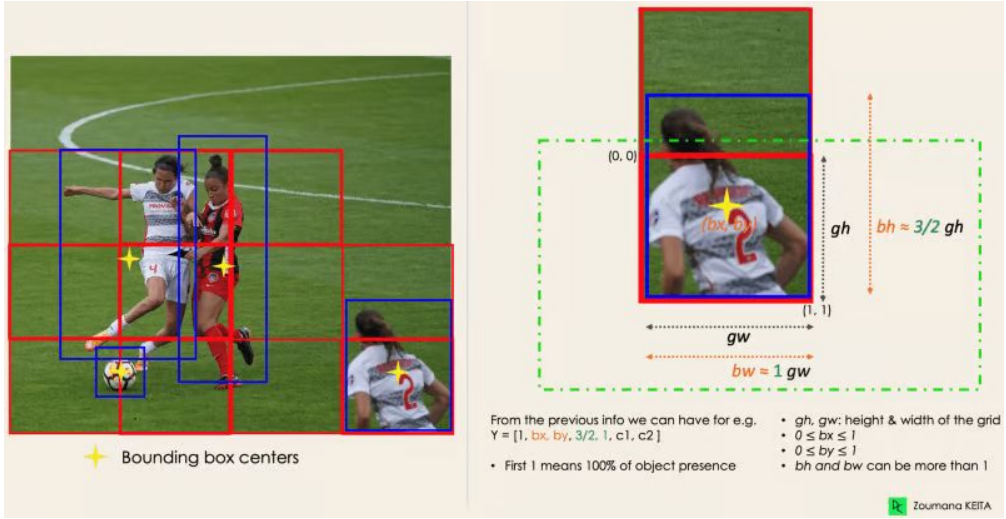


YOLO explained

YOLO: Bounding box regressor

- **pc**: probability score of the grid containing an object
- **bx, by:** coordinates of the center of the bounding box with respect to the enveloping grid cell
- **bh, bw:** height and the width of the bounding box with respect to the enveloping grid cell
- **cn**: classifier layers



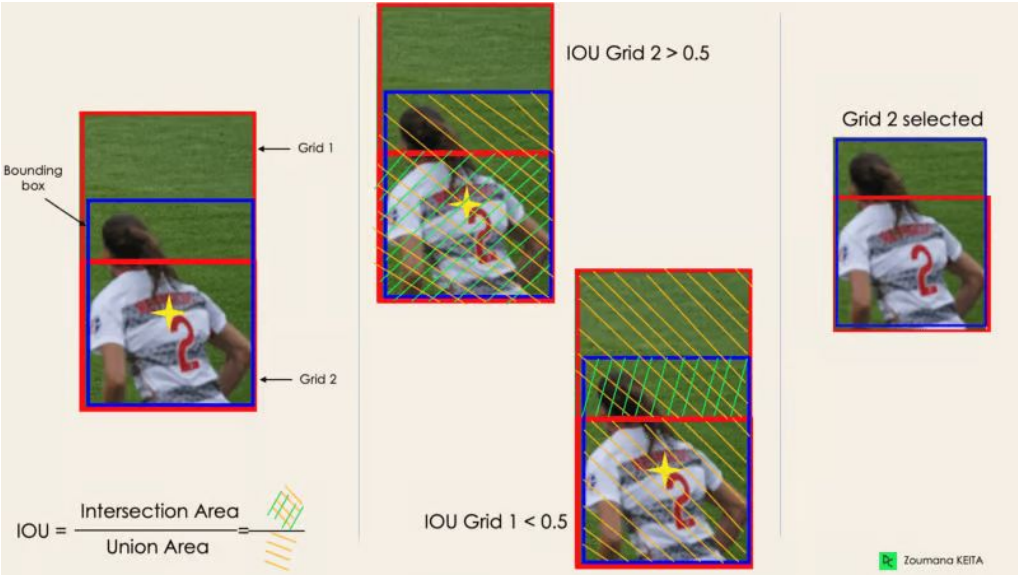


Y = [pc, bx, by, bh, bw, c1, c2]

YOLO explained

YOLO: IOU and Non-Max Suppression

- **IOU:** YOLO computes the IOU of each grid cell, which is the Intersection area divided by the Union Area. Considers those with an IOU > threshold
- **Non-Max Suppression (NMS):** keep only the boxes with probability detection score greater than NMS value.

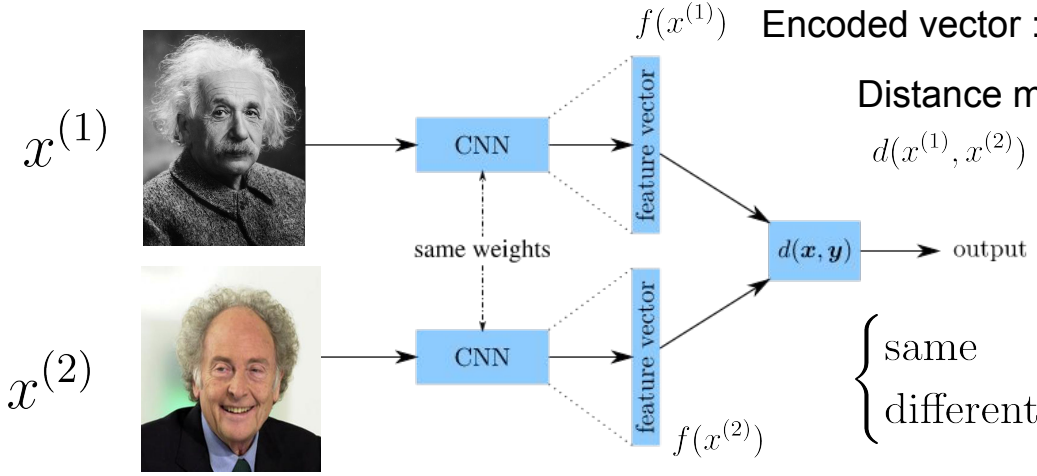


[YOLO explained]

YOLO and object detection

Face Recognition/Verification: Siamese Net

Verification: Input: Image and name => Correct/Wrong
Recognition: Input: Image => name (K possibilities) (much harder)
One-shot learning: Recognize a person given a single image. Small training set is not enough for training a CNN. Not need a classifier, use similarity function.

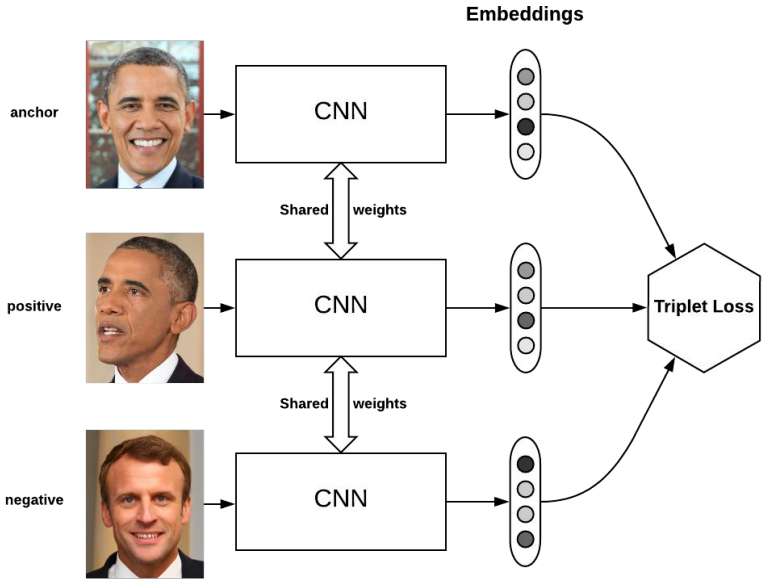


Distance measure between images

$$d(x^{(1)}, x^{(2)}) = ||f(x^{(1)}) - f(x^{(2)})||^2$$

$$\begin{cases} \text{same} & \text{if } d(x^{(1)}, x^{(2)}) \leq T \\ \text{different} & \text{if } d(x^{(1)}, x^{(2)}) > T \end{cases}$$

Face Recognition/Verification: Triplet Loss



- Train triplets: Anchor, Negative and Positive.
- We want the distance between anchor and positive be lower than the distance between the anchor and the negative image.

$$d(A, P) + \alpha < d(A, N)$$

- Train with more similars negative examples.

$$\sum_i \max\left(||f(A^{(i)}) - f(P^{(i)})||^2 - ||f(A^{(i)}) - f(N^{(i)})||^2 + \alpha, 0\right)$$

ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE DE MADRID