



Deep Learning

Alberto Ezpondaburu



Analytical Index

1. Introduction to Deep Learning
2. Convolutional Neural Networks (CNN)
 - a. Introduction to Computer Vision
 - b. Introduction to CNN
 - c. Use Cases
3. Recurrent Neural Networks (RNN)
4. Natural Language Processing with Deep Learning
5. Deep Generative Modelling

2

Convolutional Neural Networks (CNN)



2.1

Introduction to Computer Vision



Computer Vision I



- 500 hours of video are uploaded to YouTube every minute.
- More than 50 billion photos have been uploaded to Instagram.
- 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 II

“Building artificial systems that process, perceive and reason about visual data”

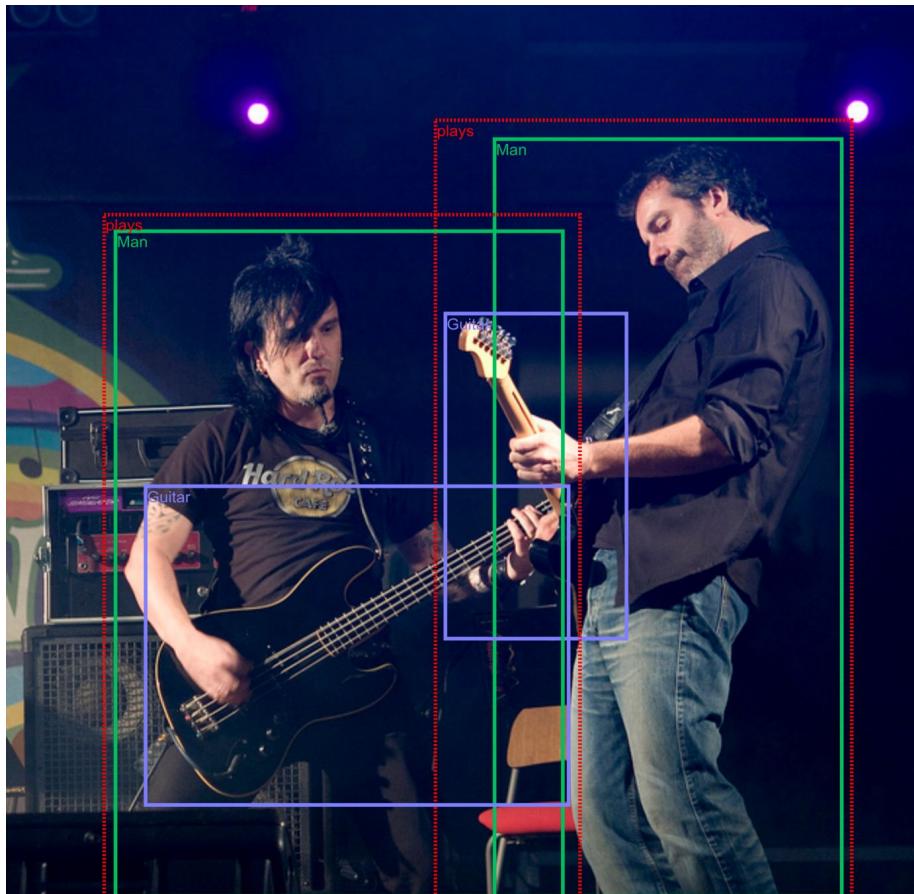
Visual Perception: 500 millions of years of evolution, involves both the evaluation of stimulus features and attention (differential processing of selected information that is relevant to behavior)



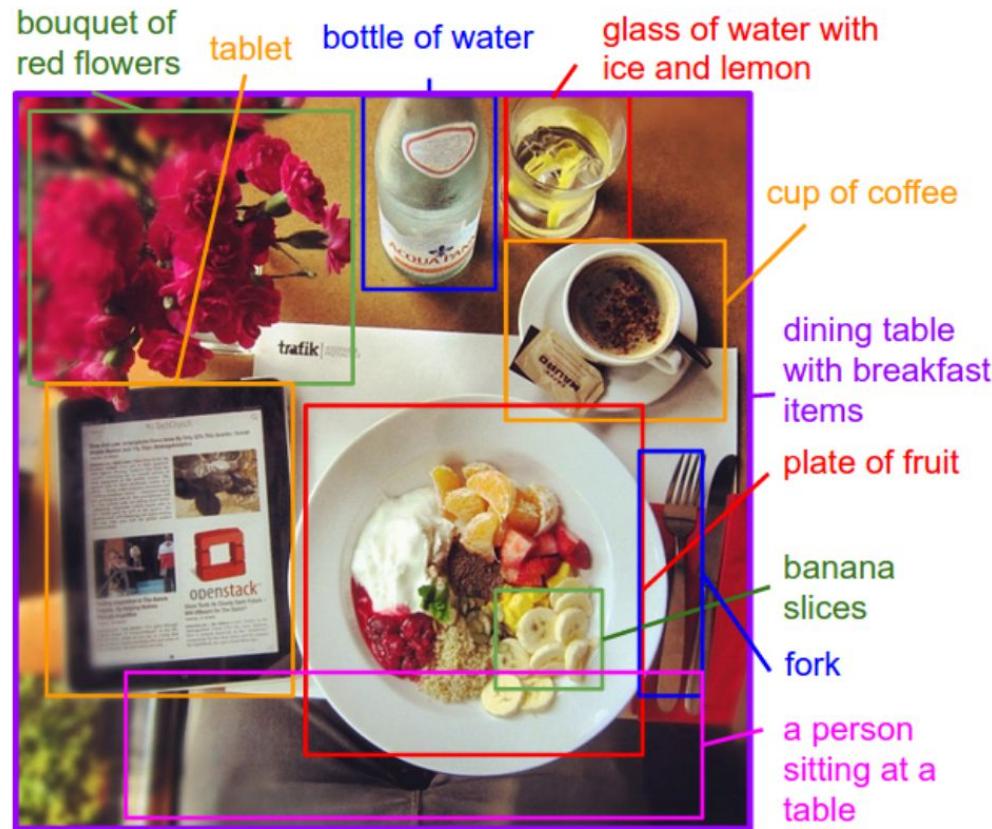
Computer Vision: Image Classification & Location



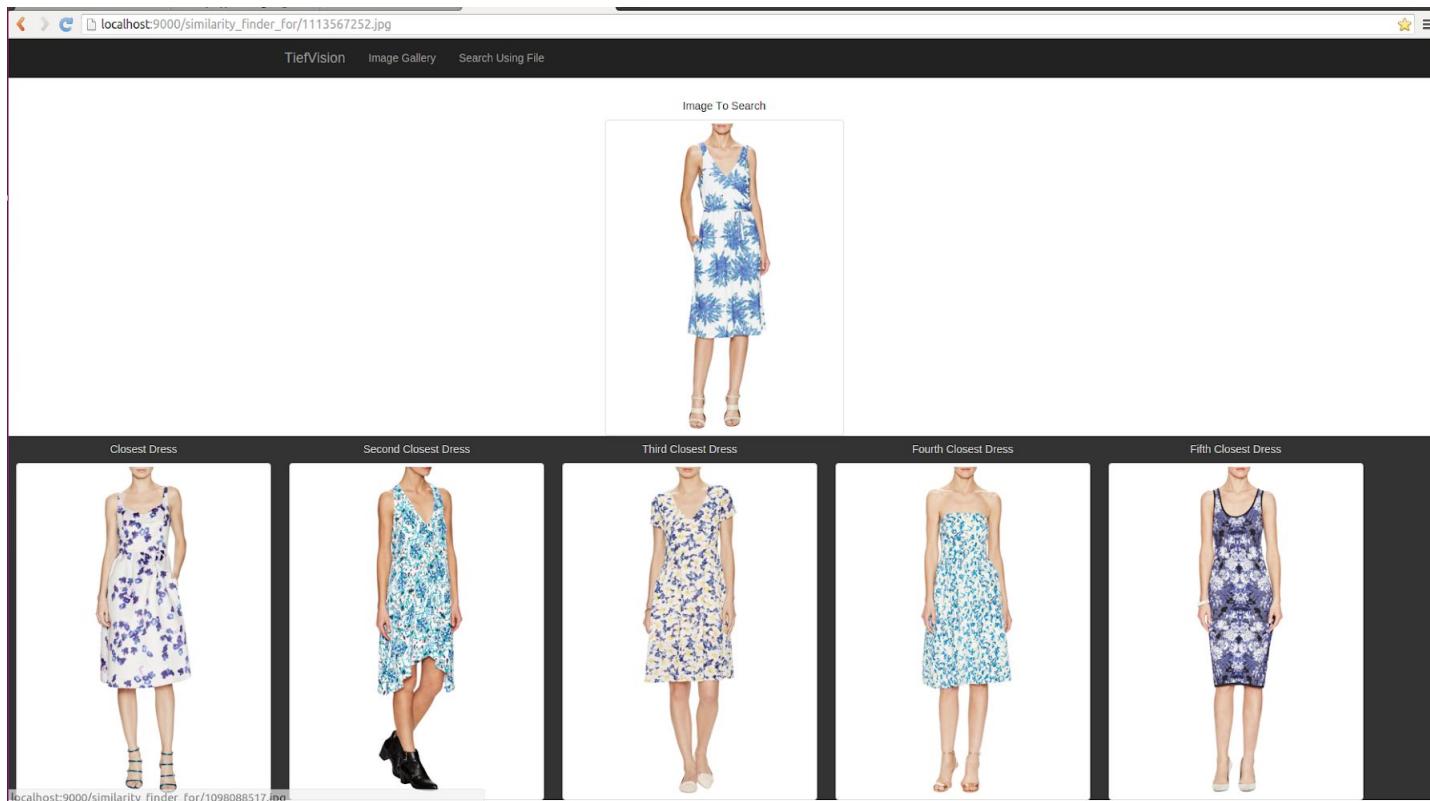
Computer Vision: Relationship Detection



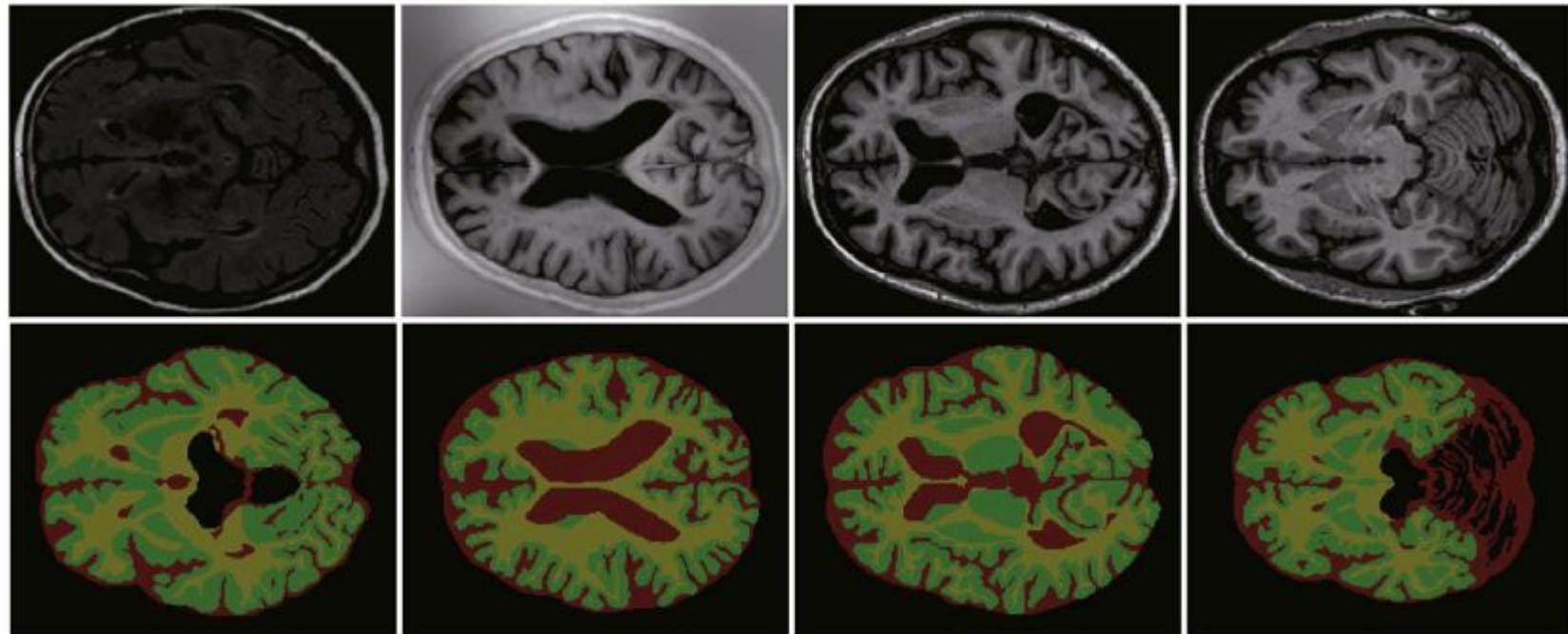
Computer Vision: Image annotation



Computer Vision: Image Similarity Search Engine



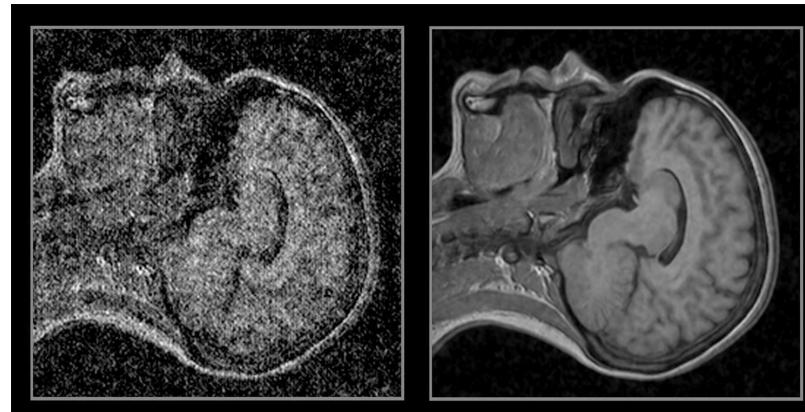
Computer Vision: Image segmentation



Computer Vision: Image Reconstruction



Computer Vision: Image Denoising



2.2

Introduction CNN:



Image Data



0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29	
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0	
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1	
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49	
13	217	243	255	155	35	226	52	2	0	10	13	232	255	255	36	
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62	
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0	
0	87	252	250	248	215	60	0	1	21	252	255	248	144	6	0	
0	13	112	255	255	245	255	182	181	248	252	242	208	36	0	19	
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0	
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0	
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4	
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0	
0	111	255	242	255	158	24	0	0	6	35	255	232	230	56	0	
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3	
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0	
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4	
0	18	146	250	255	247	255	255	249	255	240	255	129	0	5	0	
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0	
0	0	6	1	0	52	153	233	255	252	147	37	0	4	1	0	
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0	

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

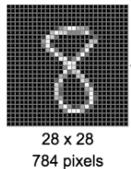


- Images are composed by pixels. 1 Megapixel can be a 1000x1000 matrix.
- Grayscale images can be seen as matrices of integers (0 -255) (black - white).

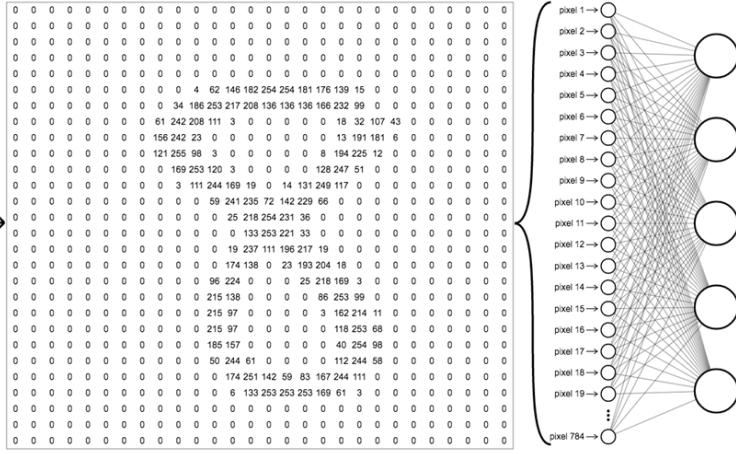
210	214	216
208	210	211
204	2	167
186	188	188
183	186	194
235	238	239
230	206	232



Problem with Dense Layers



28 x 28
784 pixels



- Spotify: 44.1kHz sample rate.
- Same problems.

- 1 Megapixel = $1000 \times 1000 = 10^6$ 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

$$\mathbf{X} \quad \begin{array}{|c|c|c|c|c|c|c|}\hline 3_1 & 0_0 & 1_{-1} & 2 & 7 & 4 \\ \hline 1_1 & 5_0 & 8_{-1} & 9 & 3 & 1 \\ \hline 2_1 & 7_0 & 2_{-1} & 5 & 1 & 3 \\ \hline 0 & 1 & 3 & 1 & 7 & 8 \\ \hline 4 & 2 & 1 & 6 & 2 & 8 \\ \hline 2 & 4 & 5 & 2 & 3 & 9 \\ \hline \end{array} \quad 6 \times 6$$
$$* \quad \mathbf{W} \quad \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \quad 3 \times 3 \quad = \quad \begin{array}{|c|c|c|c|c|c|c|c|}\hline & & & -5 & & & & \\ \hline & & & & & & & \\ \hline & & & & & & & \\ \hline & & & & & & & \\ \hline & & & & & & & \\ \hline & & & & & & & \\ \hline & & & & & & & \\ \hline & & & & & & & \\ \hline & & & & & & & \\ \hline \end{array} \quad 4 \times 4$$
$$\mathbf{Y} = \mathbf{X} \star \mathbf{W}$$
$$y_{p,q} = \sum_{i=0}^2 \sum_{j=0}^2 w_{ij} x_{i+p,j+q}$$
$$3 \times 1 + 1 \times 1 + 2 \times 1 + 0 \times 0 + 5 \times 0 + 7 \times 0 + 1 \times -1 + 8 \times -1 + 2 \times -1 = -5$$

- 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

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

$$\frac{1}{9} *$$

1	1	1
1	1	1
1	1	1

“box filter”

0	10	20	30	30	30	20	10	
0	20	40	60	60	60	40	20	
0	30	60	90	90	90	60	30	
0	30	50	80	80	90	60	30	
0	30	50	80	80	90	60	30	
0	20	30	50	50	60	40	20	
10	20	30	30	30	30	20	10	
10	10	10	0	0	0	0	0	



Convolution Operation: Example

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

6 x 6

*

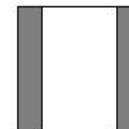
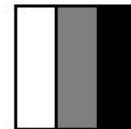
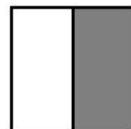
1	0	-1
1	0	-1
1	0	-1

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

8×6

*

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

3×3

6×4

=

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

$$\begin{array}{|c|c|c|c|c|}\hline 35 & 40 & 41 & 45 & 50 \\\hline 40 & 40 & 42 & 46 & 52 \\\hline 42 & 46 & 50 & 55 & 55 \\\hline 48 & 52 & 56 & 58 & 60 \\\hline 56 & 60 & 65 & 70 & 75 \\\hline\end{array} \times \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\\hline 0 & 0 & 0 \\\hline 0 & 0 & 0 \\\hline\end{array} = \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\\hline 0 & 0 & 0 \\\hline 0 & 42 & 0 \\\hline\end{array}$$

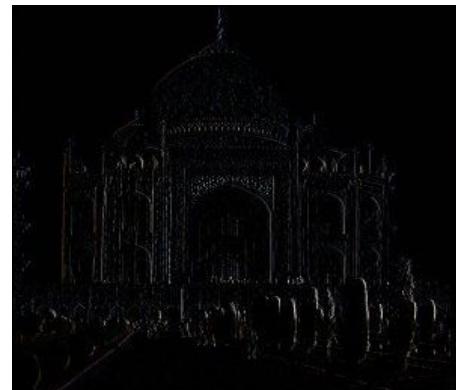
Edge Detection

$$\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\\hline 1 & -4 & 1 \\\hline 0 & 1 & 0 \\\hline\end{array}$$



Edge Detection
(right)

$$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\\hline -1 & 1 & 0 \\\hline 0 & 0 & 0 \\\hline\end{array}$$



Convolution Operation: Example



Blur

0	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	0

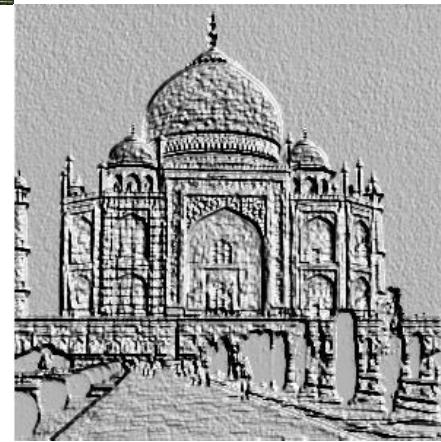


Sharpen

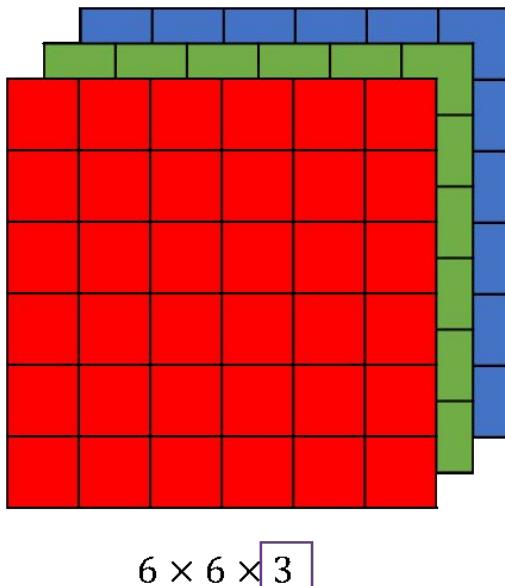
0	0	0	0	0	0
0	0	-1	0	0	
0	-1	5	-1	0	
0	0	-1	0	0	
0	0	0	0	0	0

Emboss

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

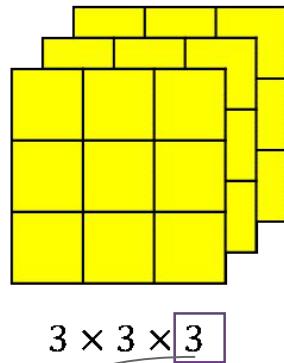


Convolution Operation: RGB

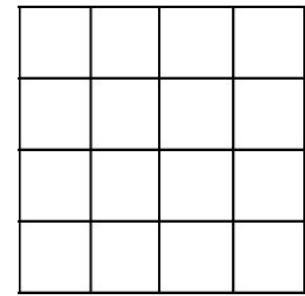


*

3 separate convolutions and then sum all 3

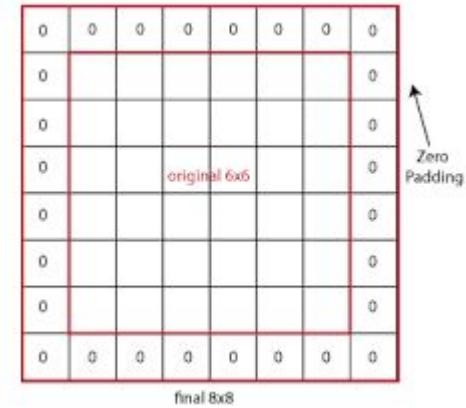
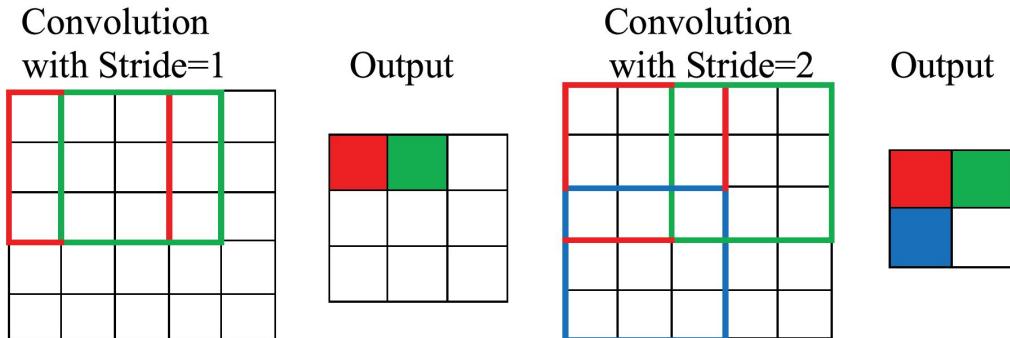


=



Convolution: Padding & Stride

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

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

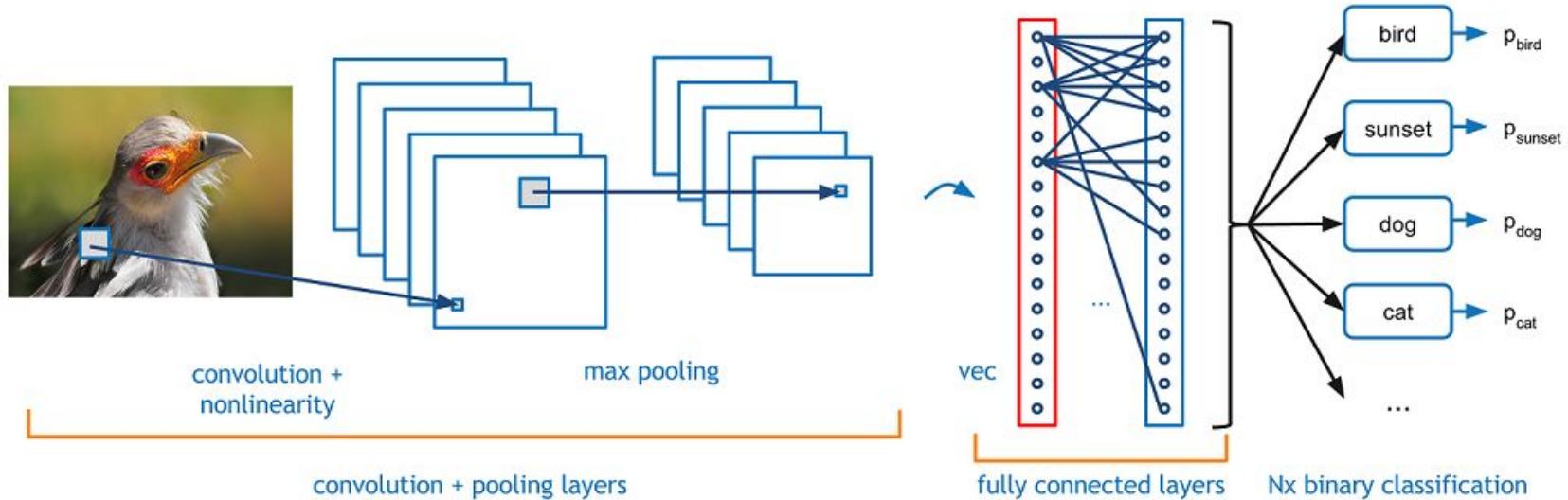


CNNs Advantages

- **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

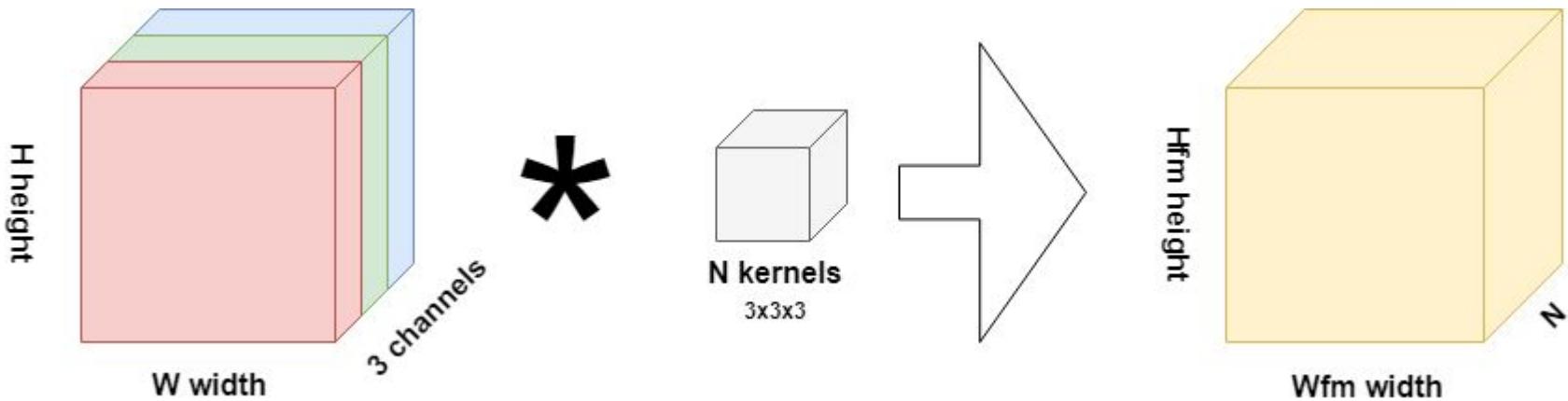
```
tf.keras.layers.Conv2D
```

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

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



CNN Layers: Convolution



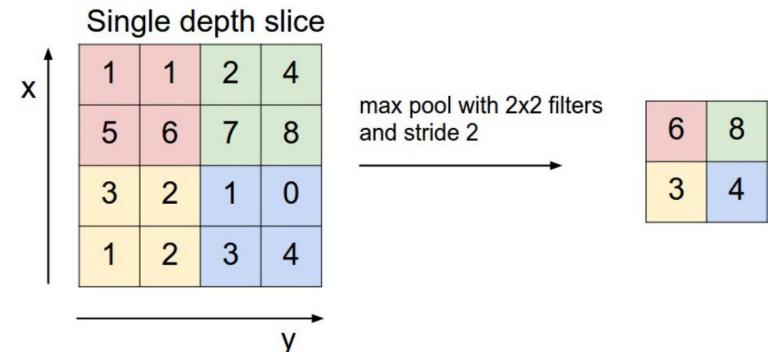
Layer dimensions: $(h \times w \times N)$ $\text{Relu}(\text{convolution} + b)$ for every filter (N different filters)

```
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.**

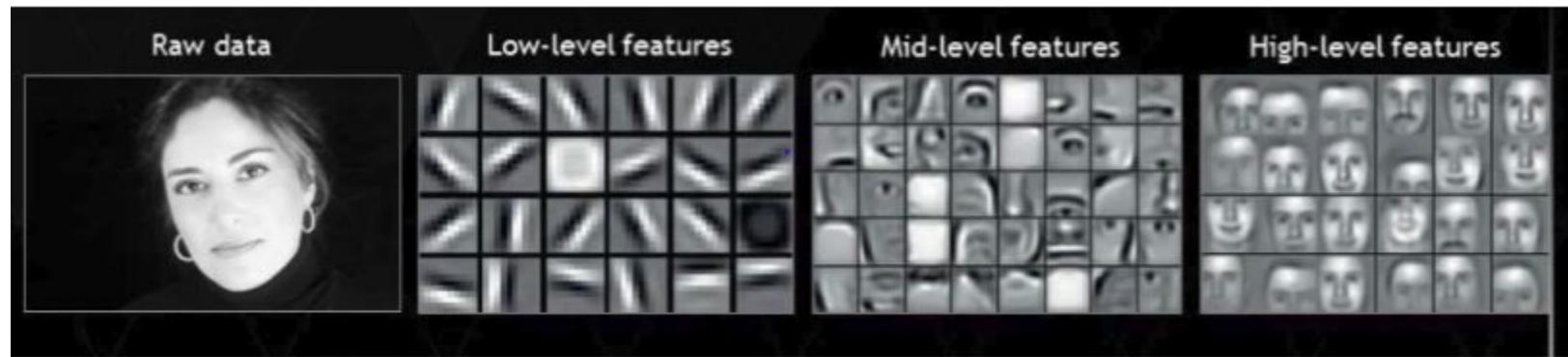


```
tf.keras.layers.MaxPooling2D(  
    pool_size=(2, 2), strides=(1, 1)  
)
```

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



CNN Features



Conv 1

Conv 2

Conv 3



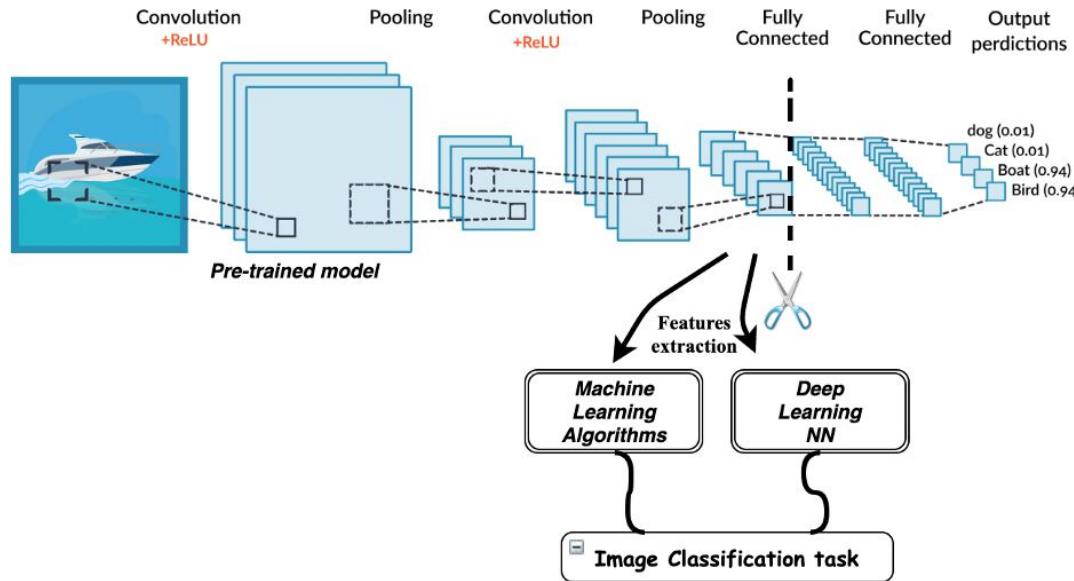
CNN: Data Augmentation



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



CNN: Transfer learning

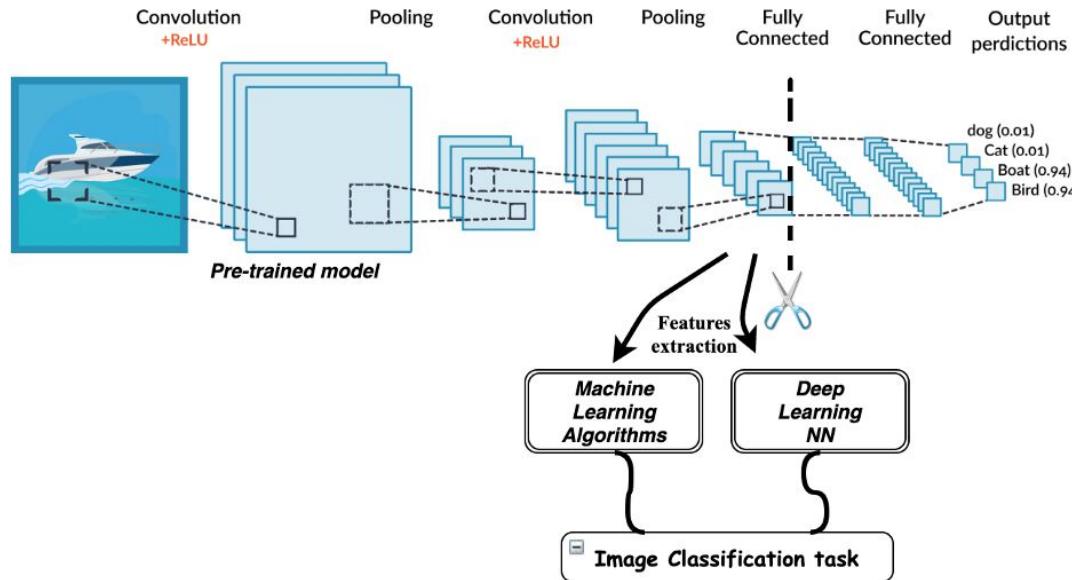


Instead of training a Deep Neural Net from scratch for your task:

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



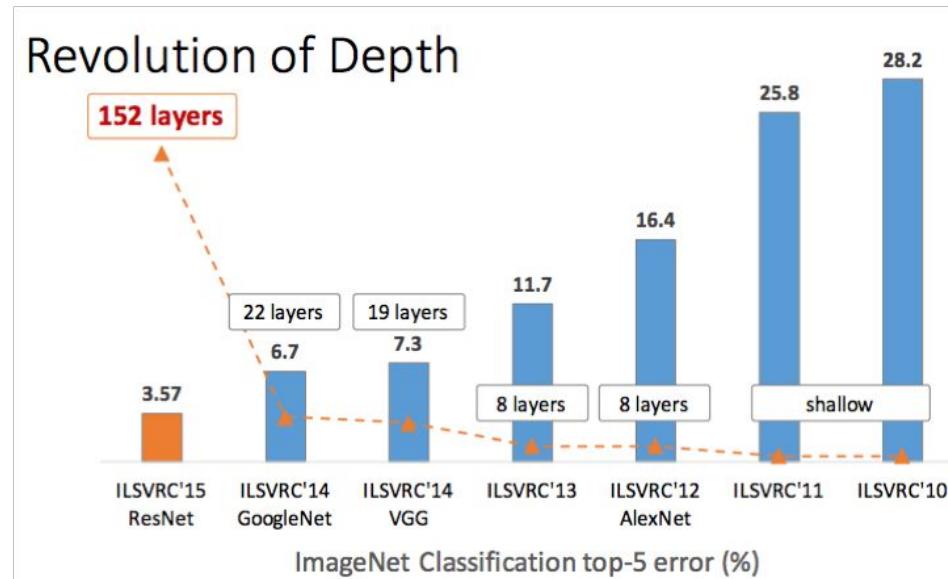
CNN: Fine-tuning



We not only update the CNN architecture but also re-train it to learn new object classes.



Modern Architectures: Revolution on Depth



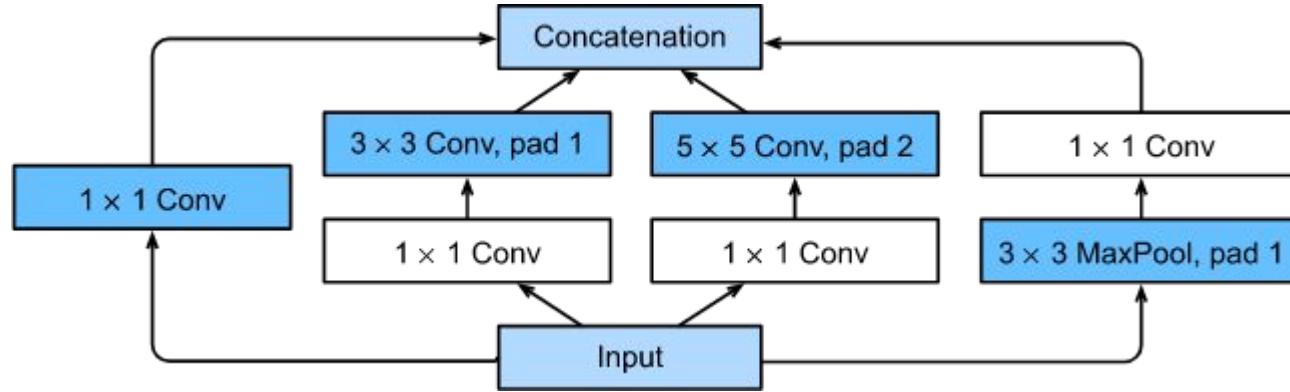
Human Beings : Top-5 Error Rate – 5.1%

Source:

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



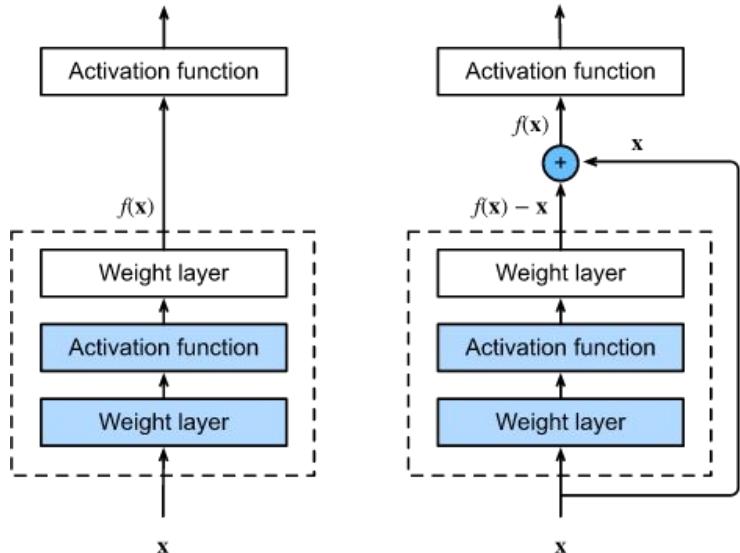
Modern Architectures: GoogleNet, Inception blocks



- Networks with Parallel Concatenations
- The Inception block is equivalent to a subnetwork with four paths



Modern Architectures: ResNet



- 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:

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	-

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



2.3

Use Cases:



Neural Style Transfer

content image



(C)

style image



(S)

generated image



(G)



Paper : A Neural Algorithm of Artistic Style



Neural Style Transfer: Visualizing Deep Layer

1. Use pre-trained model for extracting content and style (VGG-19).
2. Inputs: a pair of images, style and content.
 - a. Extract style from image S
 - b. Extract content from image C
3. Generate a new imagen with a training loop (gradient descend)
 - a. Optimize Generated image G



(C)



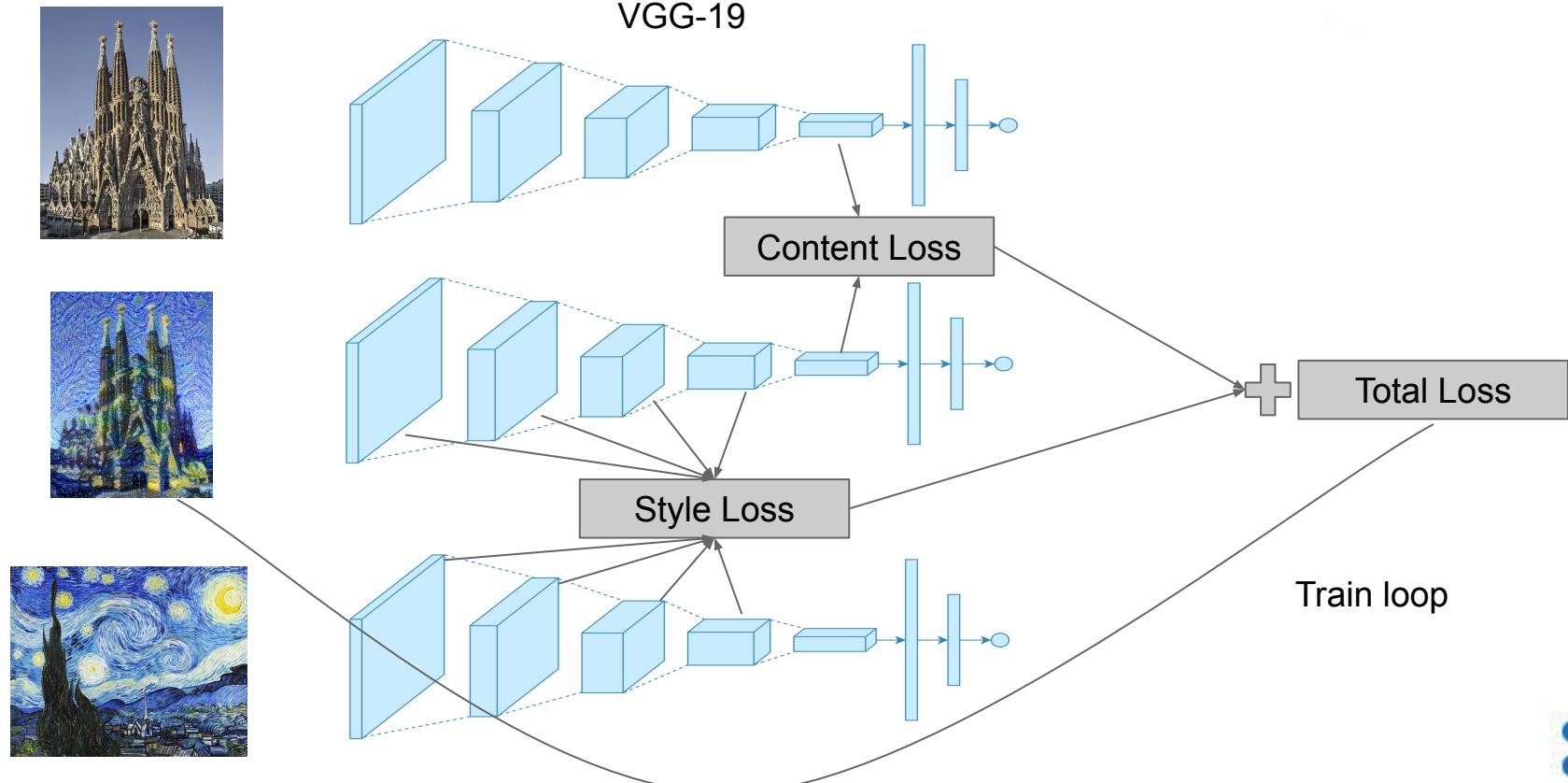
(S)



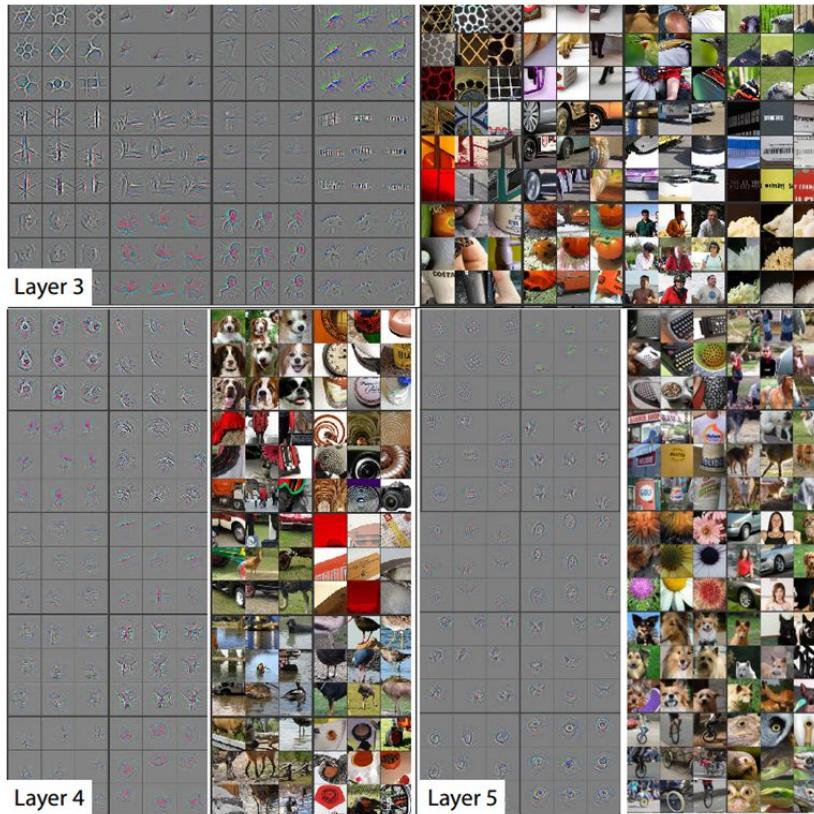
(G)



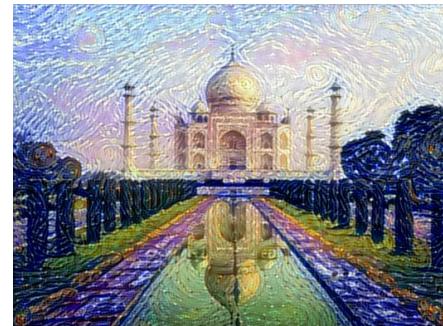
Neural Style Transfer: Visualizing Deep Layer



Neural Style Transfer: Visualizing Layers



Neural Style Transfer

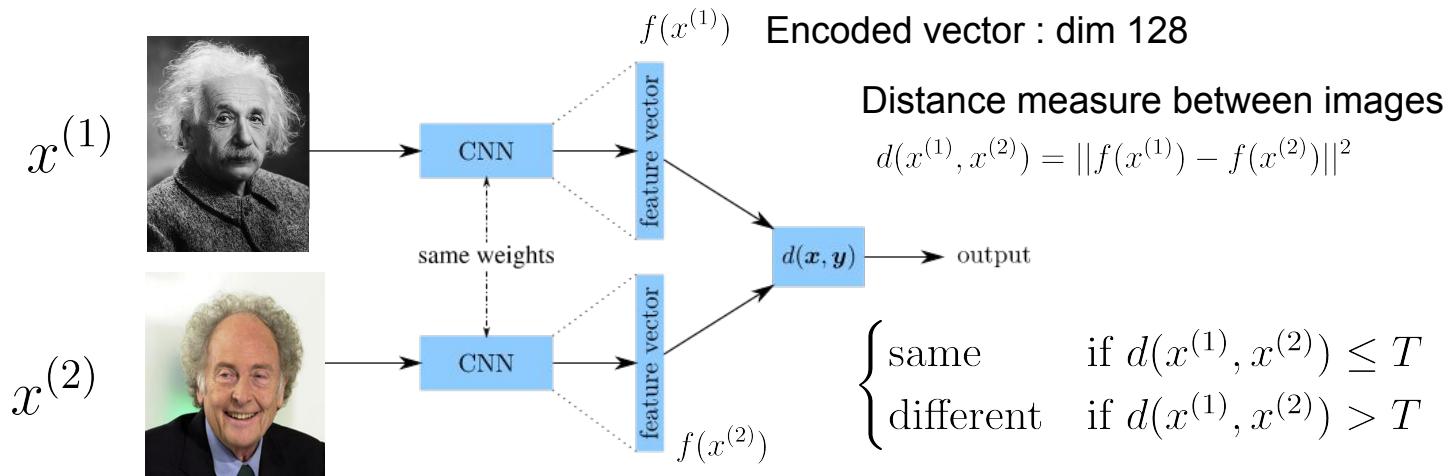


Face Recognition/Verification: Siamese Net

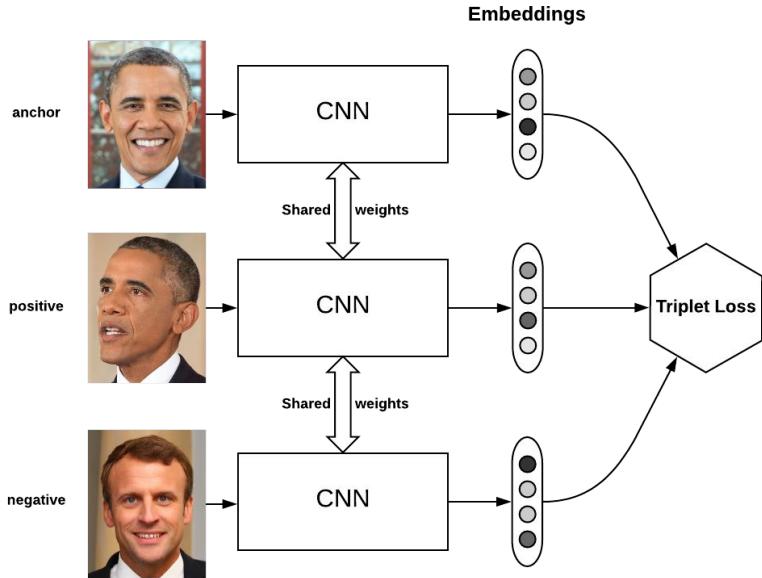
Verification: Input: Image and name => Correct/Wrong

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.



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.
$$d(A, P) < d(A, N) + \alpha$$
- Train with more similars negative examples.

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



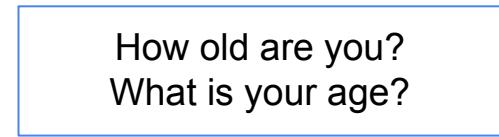
Siamese Net: Other Applications



Handwritten Signature



Voice check



How old are you?
What is your age?

Question duplicates



Image Classification Advances

Classification



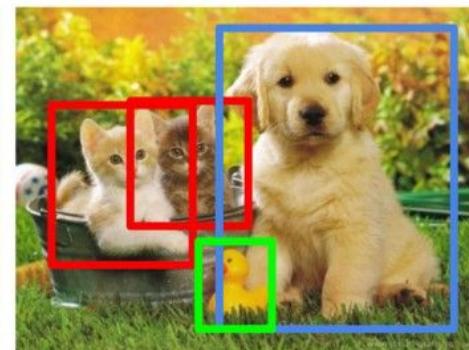
CAT

Classification + Localization



CAT

Object Detection



CAT, DOG, DUCK

Instance Segmentation



CAT, DOG, DUCK

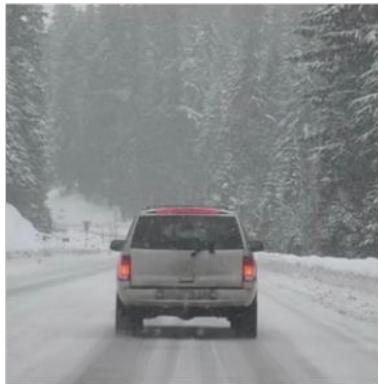
Single object

Multiple objects



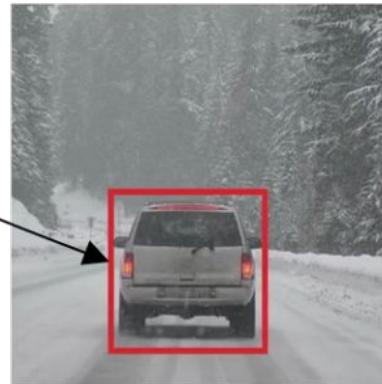
Object Localization I

Image classification



“Car”

Classification with
localization



One object

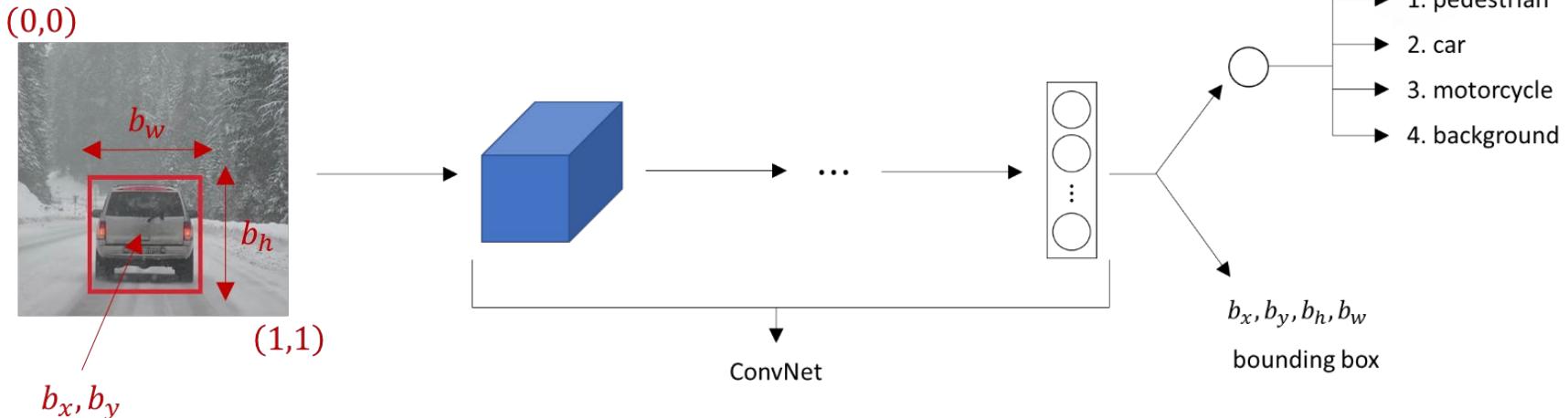
Detection



Multiple objects



Object Localization II



Target:

$$y = \begin{bmatrix} c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad \left. \begin{array}{l} \text{Sigmoid: Is there an object?} \\ \text{Bounding box, ReLu} \\ \text{Classes of objects, softmax} \end{array} \right\}$$





UNIVERSIDAD
COMPLUTENSE
DE MADRID

