AI Applications – NLP, Computer Vision, IoT
UCS655

Unit 3

Fundamentals of Computer Vision and its applications

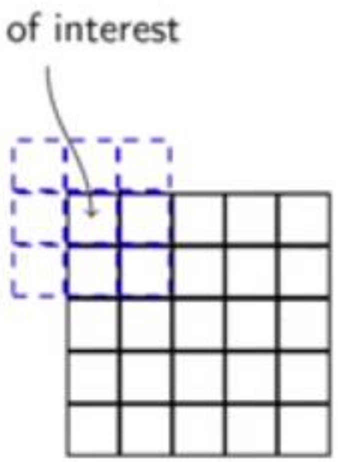
Content

- Introduction and goal of computer vision
- Basics of image processing and formation
- Introduction of ANN
- **Convolutional neural network**
- Application of computer vision in face recognition

Convolution

- Convolution is a mathematical way of combining two signals to form a third signal

$$G(i,j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} \underbrace{H(u,v)}_{\text{Non-uniform weights}} I(i-u, j-v)$$

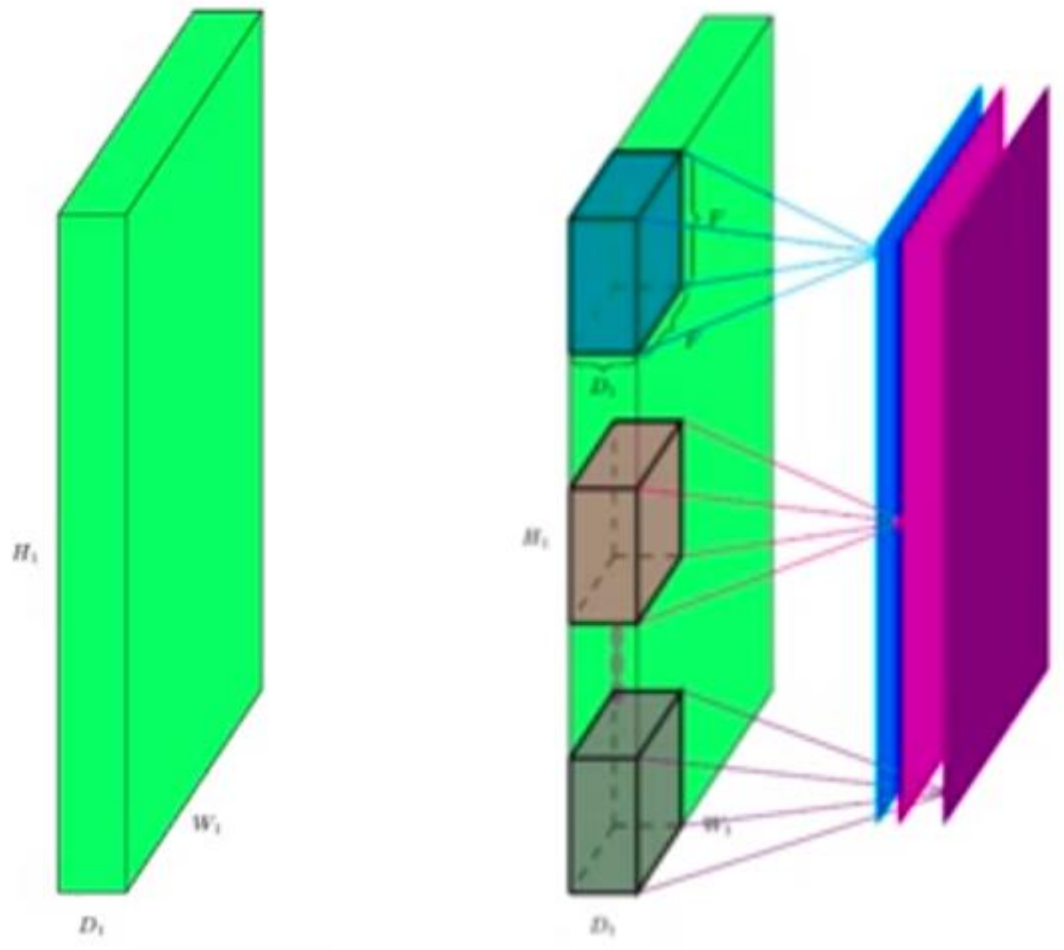


- General Representation: $K_1 \times K_2$

$$Y(i,j) = \sum_{a=\lfloor -\frac{K_1}{2} \rfloor}^{\lfloor \frac{K_1}{2} \rfloor} \sum_{b=\lfloor -\frac{K_2}{2} \rfloor}^{\lfloor -\frac{K_2}{2} \rfloor} X(i-a, j-b) W\left(\frac{K_1}{2}+a, \frac{K_2}{2}+b\right)$$

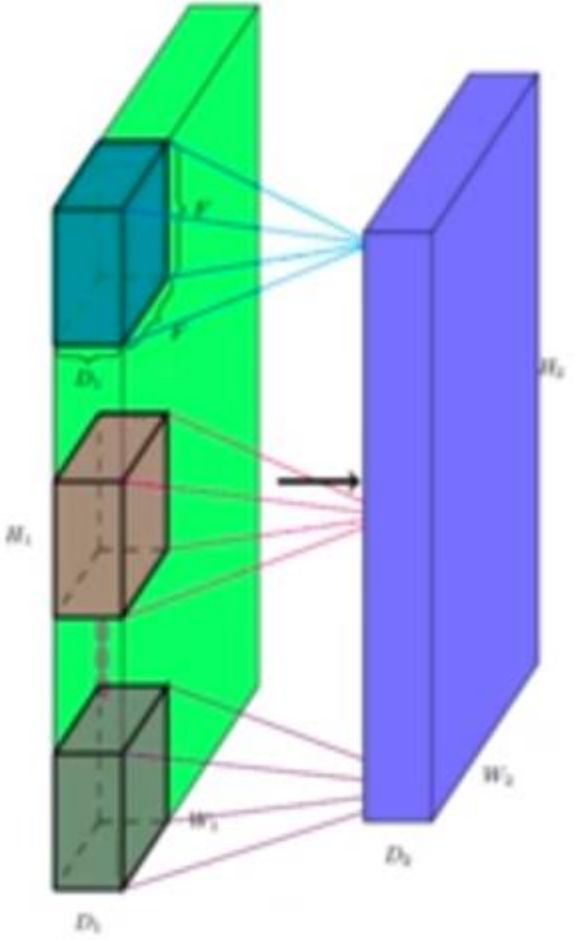
Convolution (Hyper)Parameters

- Input dimensions: $W_1 \times H_1 \times D_1$
- Spatial extent of a filter is $F \times F$
- Output dimensions: $W_2 \times H_2 \times D_2$

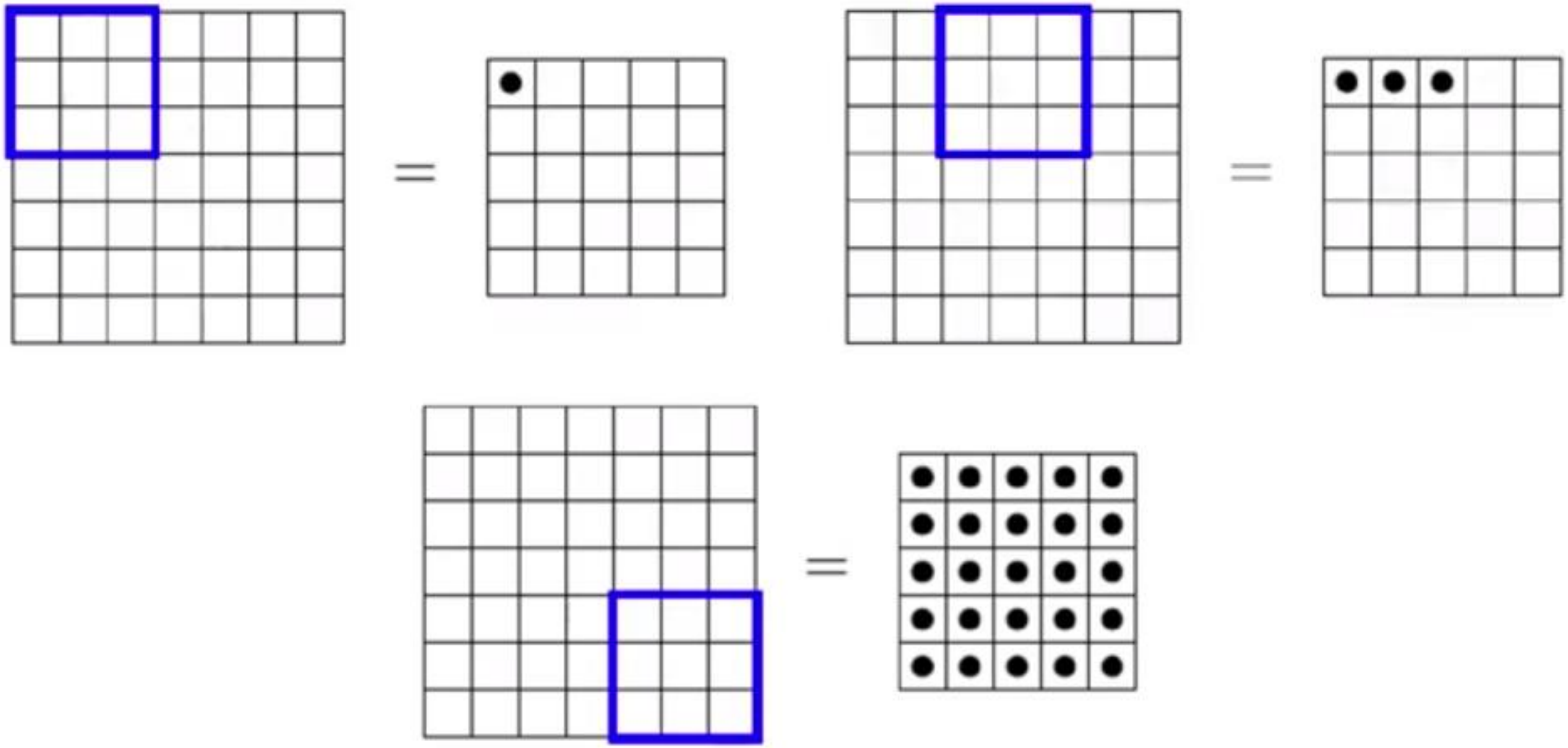


Convolution (Hyper)Parameters

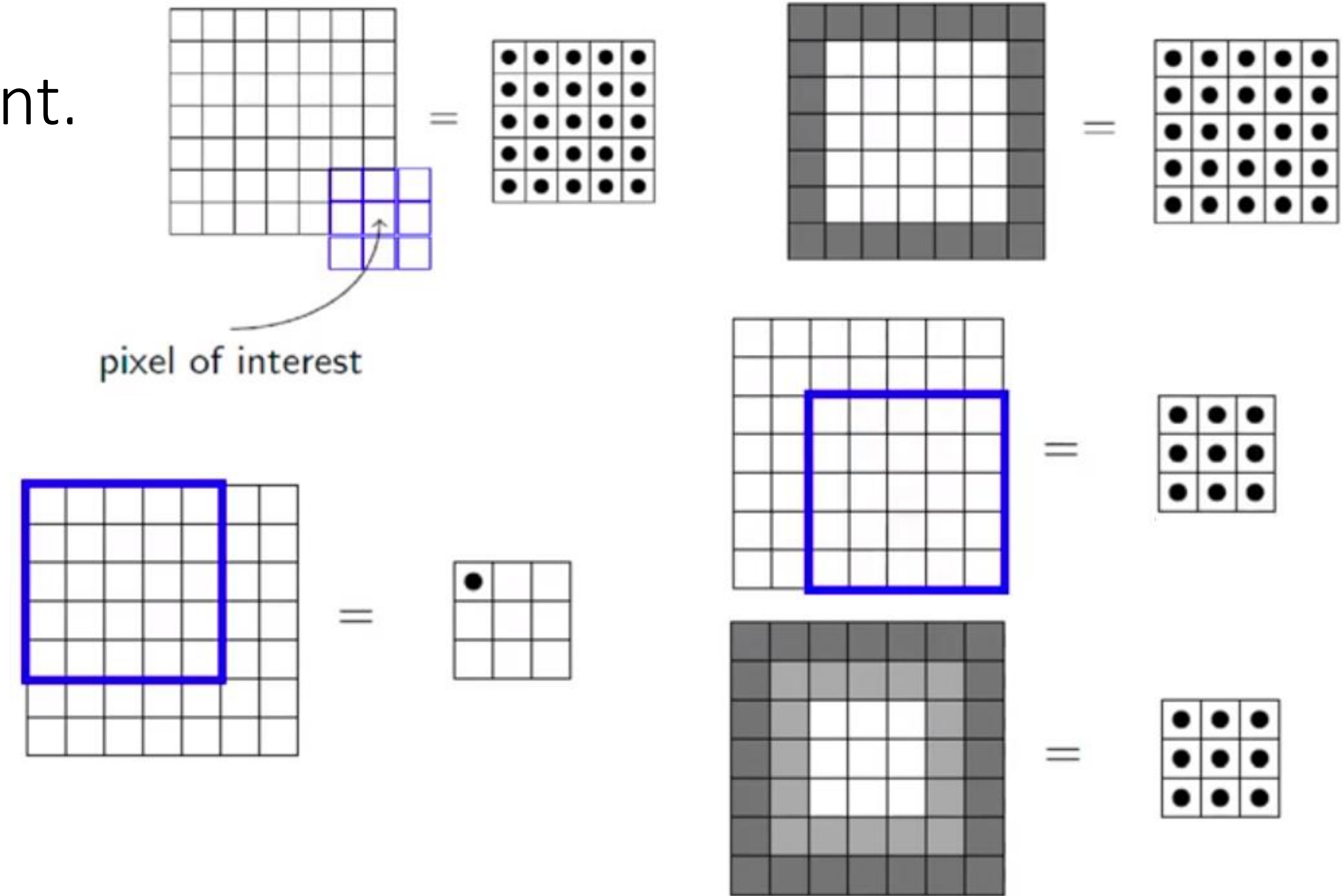
- Find dimensions (W_2, H_2)
- Stride, S
- Number of filters, K



Convolution (Hyper)Parameters

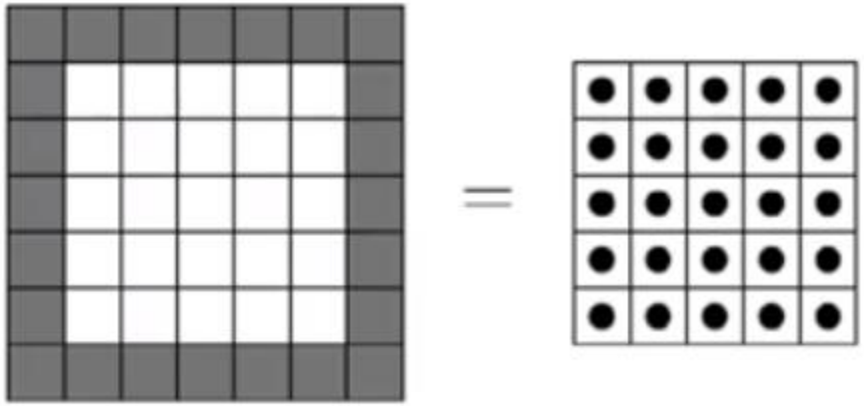
- Computation of (W_2, H_2) of output
- 

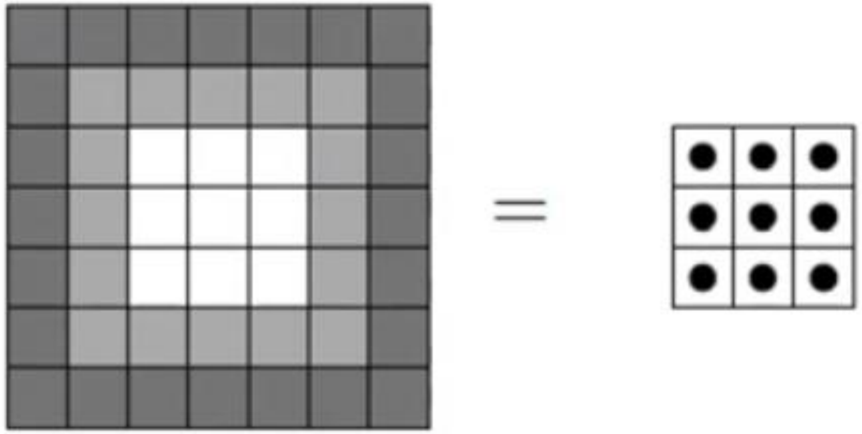
Cont.

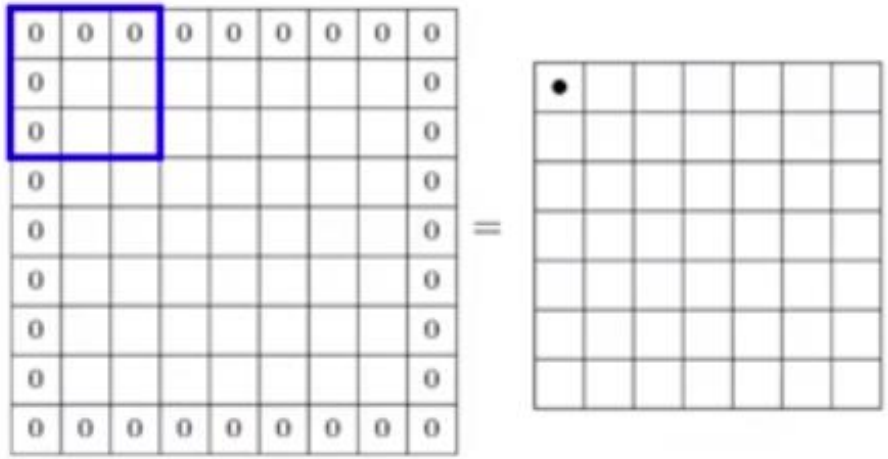


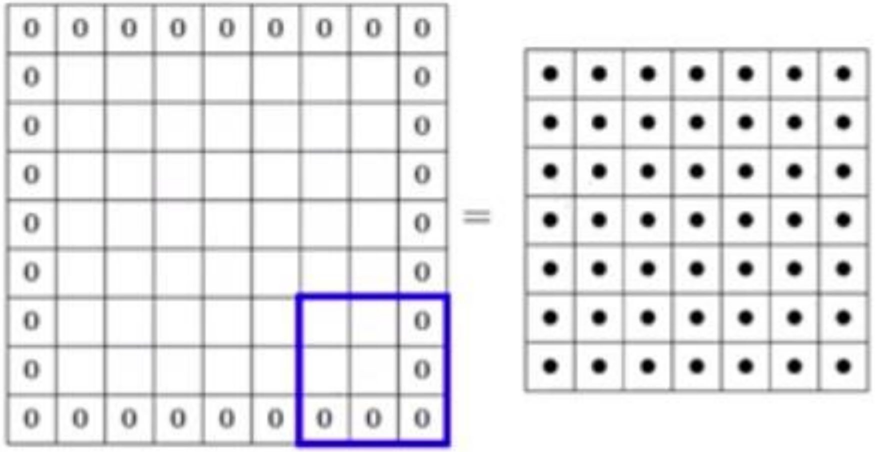
Padding

- New Dimensions of the output:
  - $W_2 = W_1 - F + 1$
  - $H_2 = H_2 - F + 1$
- Padding
  - P=1, on 3x3 kernel
- New Dimensions of the output:
  - $W_2 = W_1 - F + 2P + 1$
  - $H_2 = H_2 - F + 2P + 1$





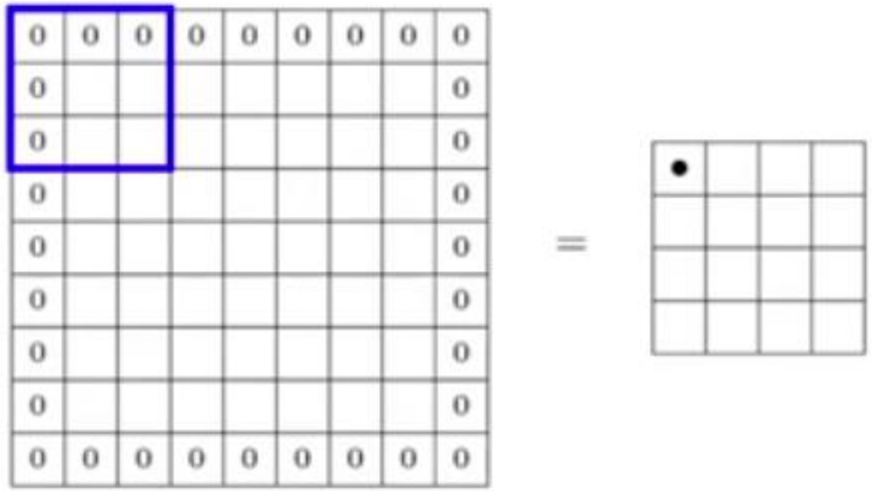


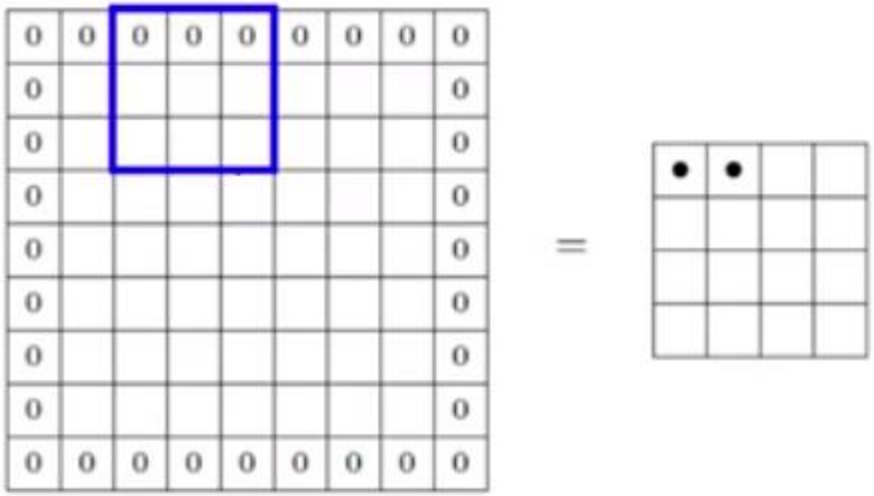


Stride

- It defines the interval at which the filter is applied
- S = 2, skips every 2nd pixel
  - Result in smaller dimensions
- New Dimensions of the output:
  - $W_2 = \frac{W_1 - F + 2P}{S} + 1$
  - $H_2 = \frac{H_2 - F + 2P}{S} + 1$
  - $D_2 = K$

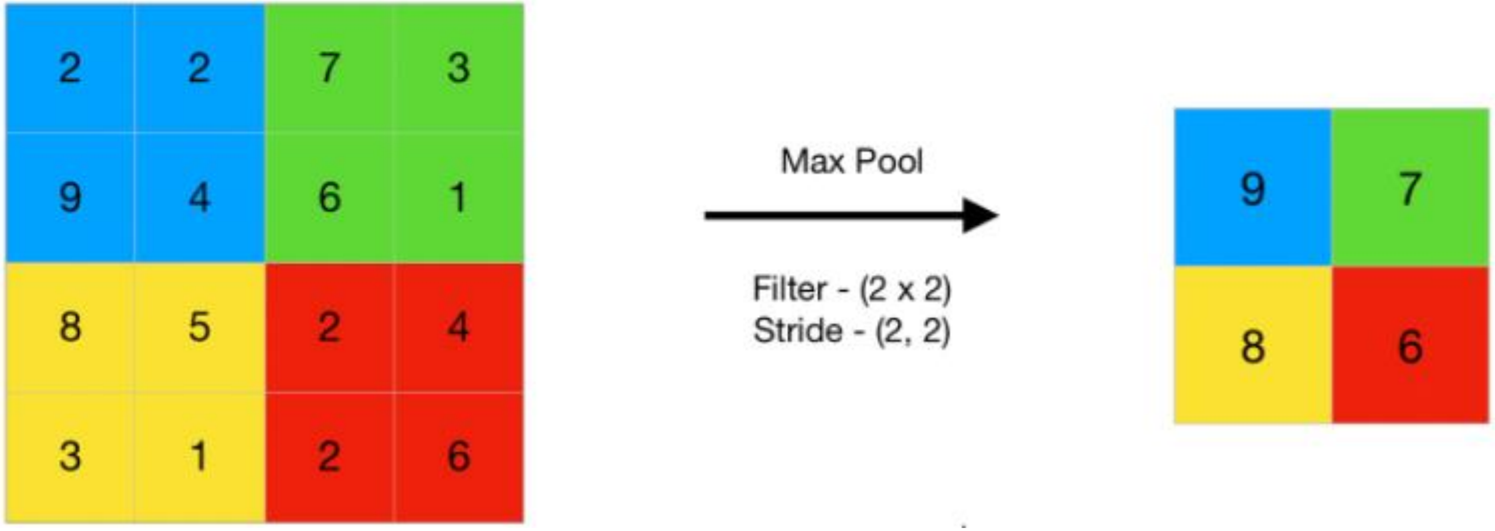
https://setosa.io/ev/image-kernels/





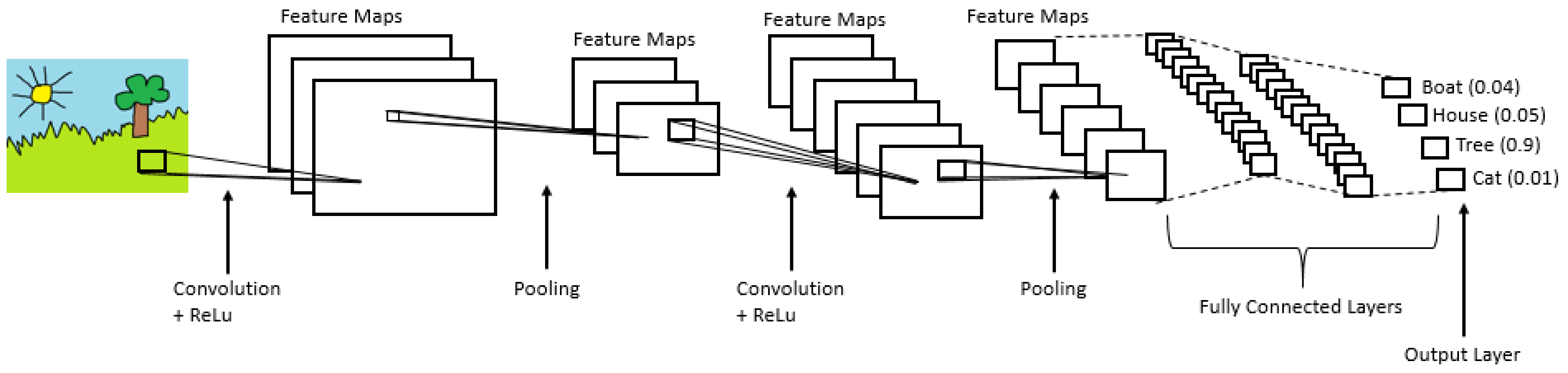
Pooling

- Refers to a small portion
  - Average Pooling - take a small portion of the input and compute the average value
  - Max Pooling – if we take a maximum value
- we are not taking out all the values we are taking a summarized value over all the values present



CNN

- Traditional machine learning based computer vision solutions – static feature engineering
  - Do not scale well to the real world images
- Can we learn meaningful kernels as apart of learning algorithm in addition to learning weights of classifier?



CNN

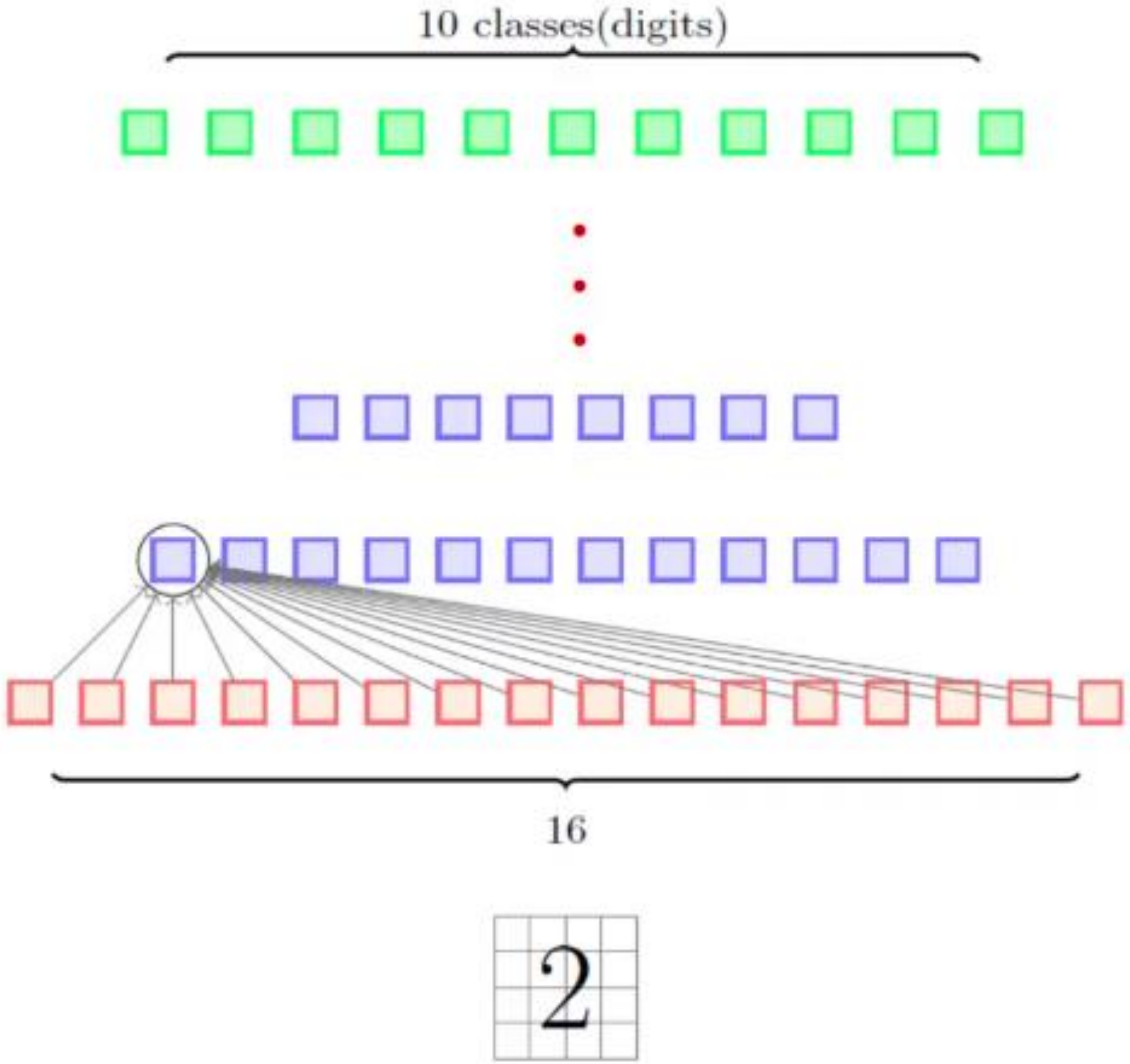
- Why can't we just take raw pixels of the input image to the FNN, why there is this convolution in between?
- MNIST dataset – FNN
  - Good performance – 2% error
- Limitations
  - Ignores spatial structure
  - No way for the network to learn same features at different places in the image
  - Computationally expensive

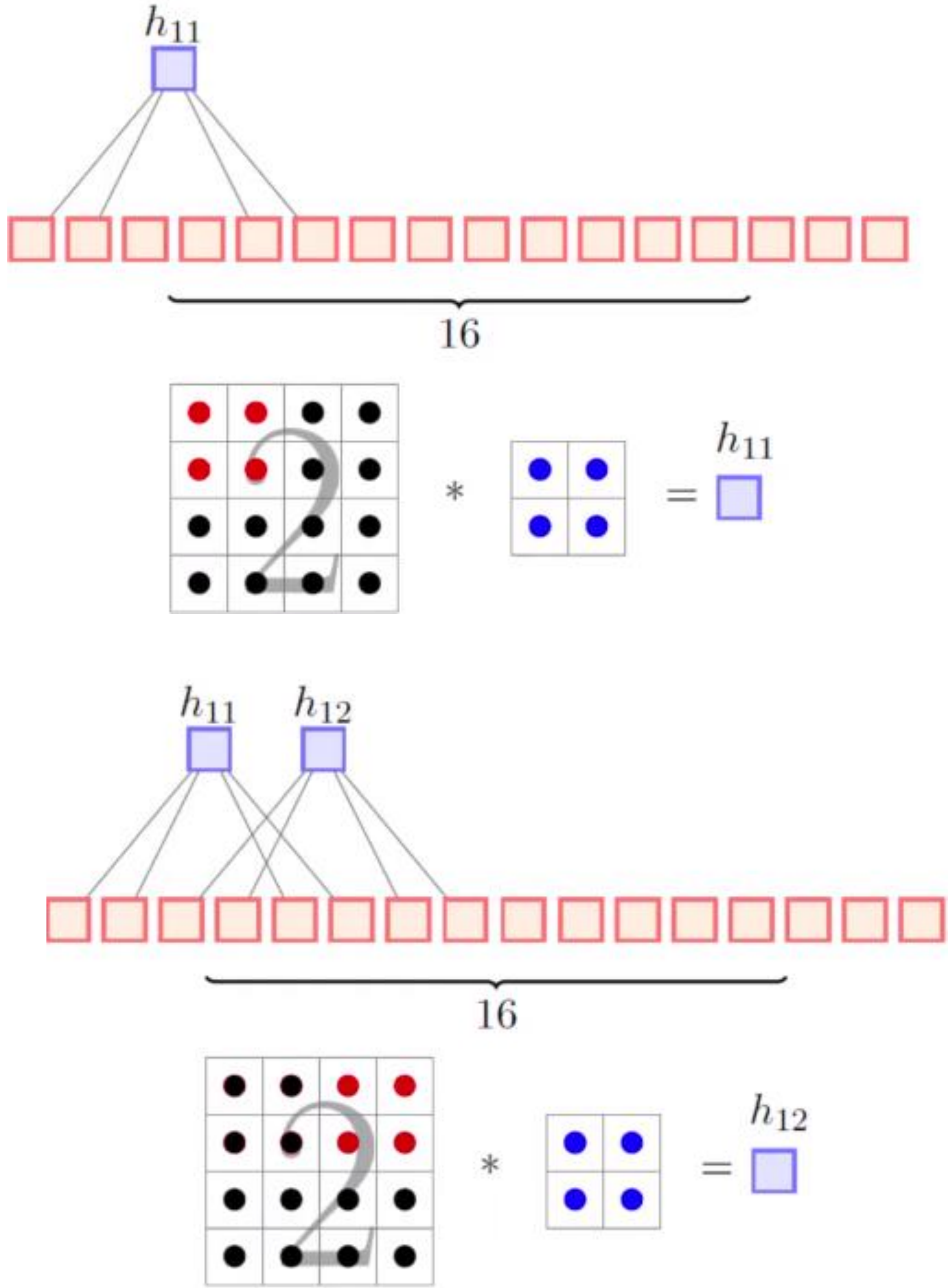
MNIST Dataset

How CNN solve these issues

- Local receptive fields
  - Capture local spatial relationships in pixels
  - Greatly reduces number of parameters
- Weight sharing
  - Enables translation invariance of NN to objects in the images
  - Reduces number of parameters
- Pooling
  - Aggregates information
  - Reduces size of the output, which reduces number of computations in the later layers

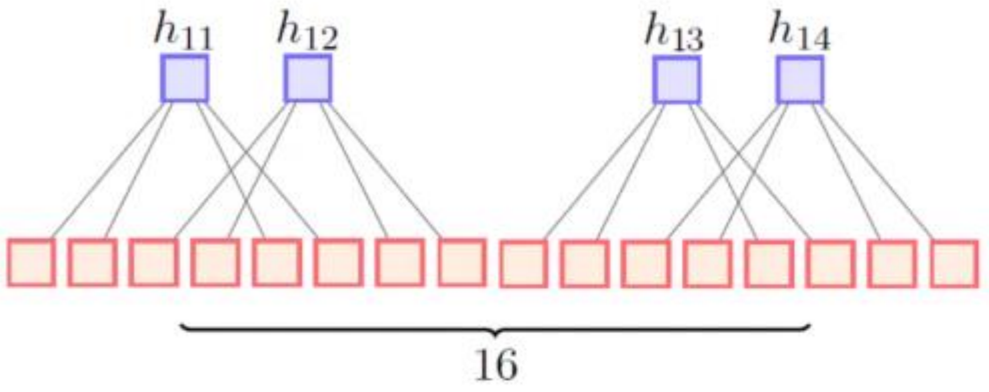
Local Receptive Fields

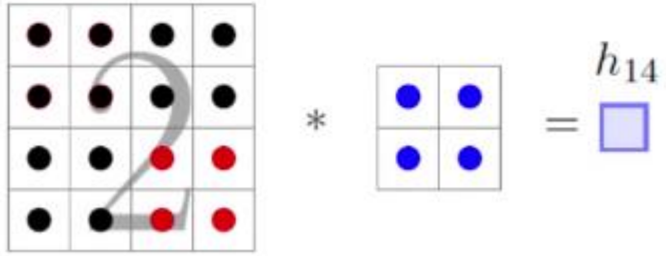


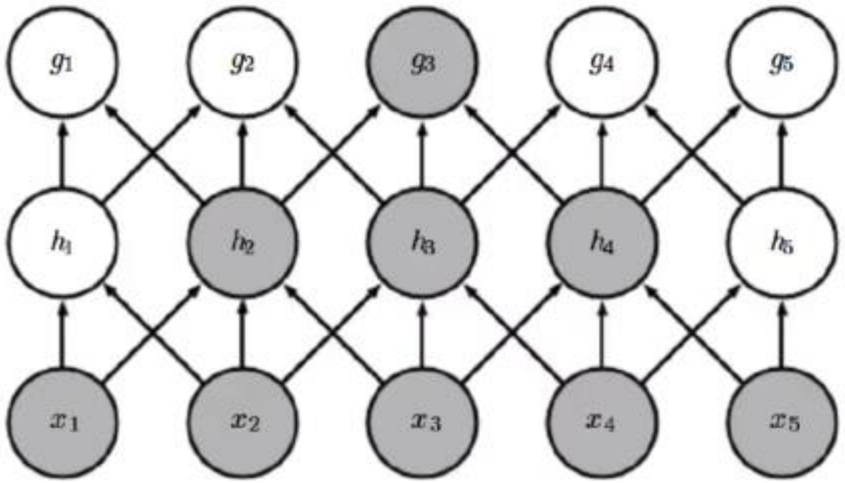


Local Receptive Fields

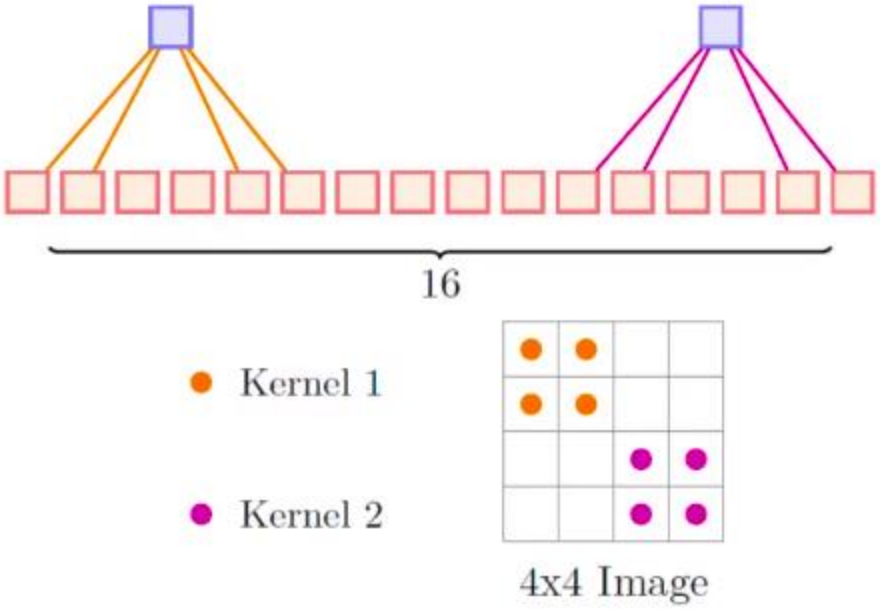
- How does this helps?
  - Makes connection sparser
    - Reduces number of parameters
  - Taking advantage of image structure
- Don’t we loose information through this process
  - All the neurons interact over the depth of the NN

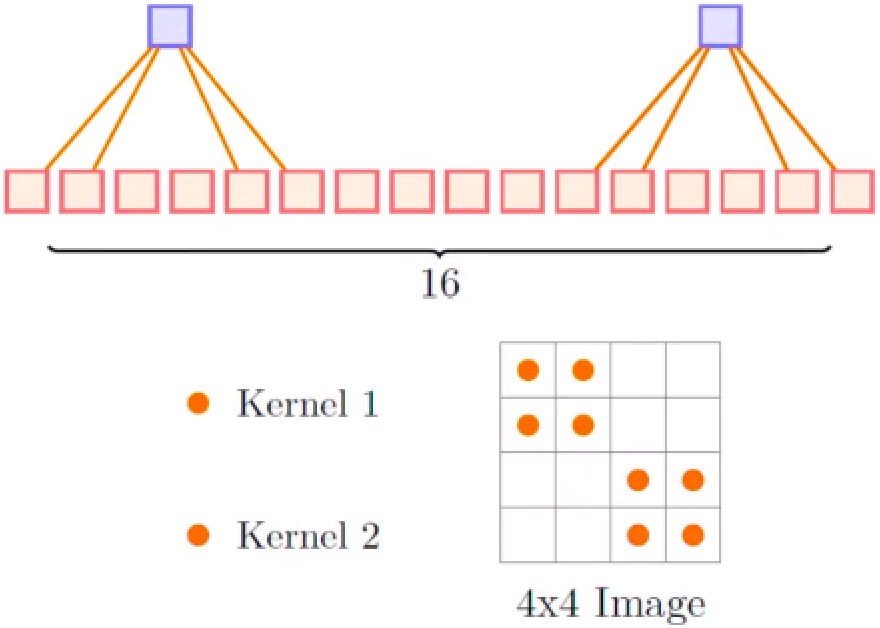


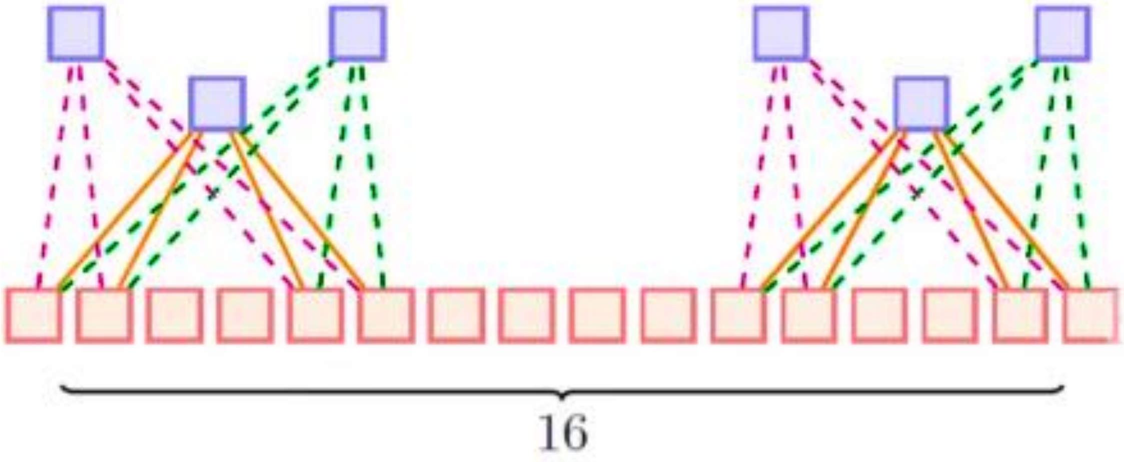




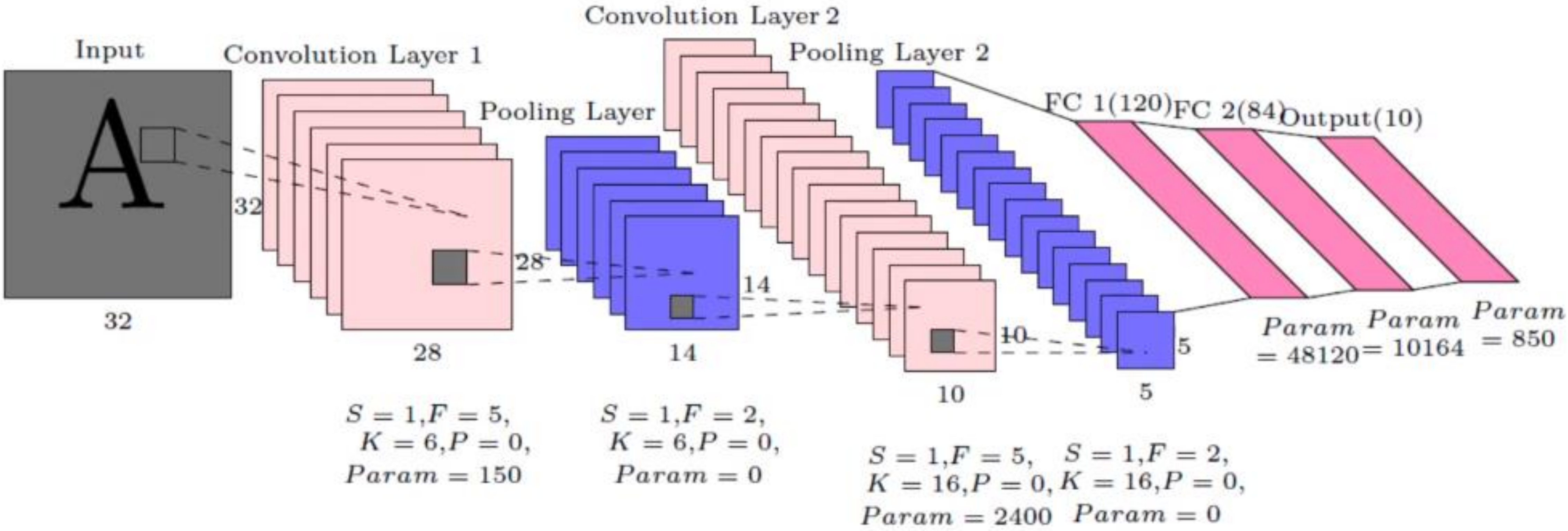
Weight Sharing





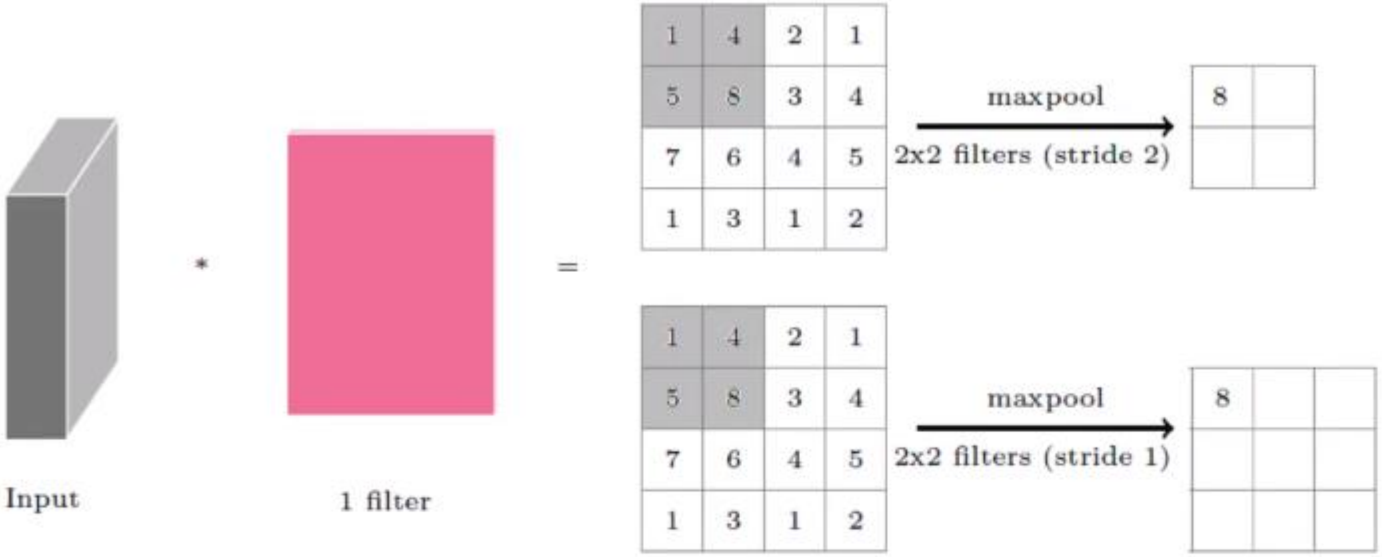


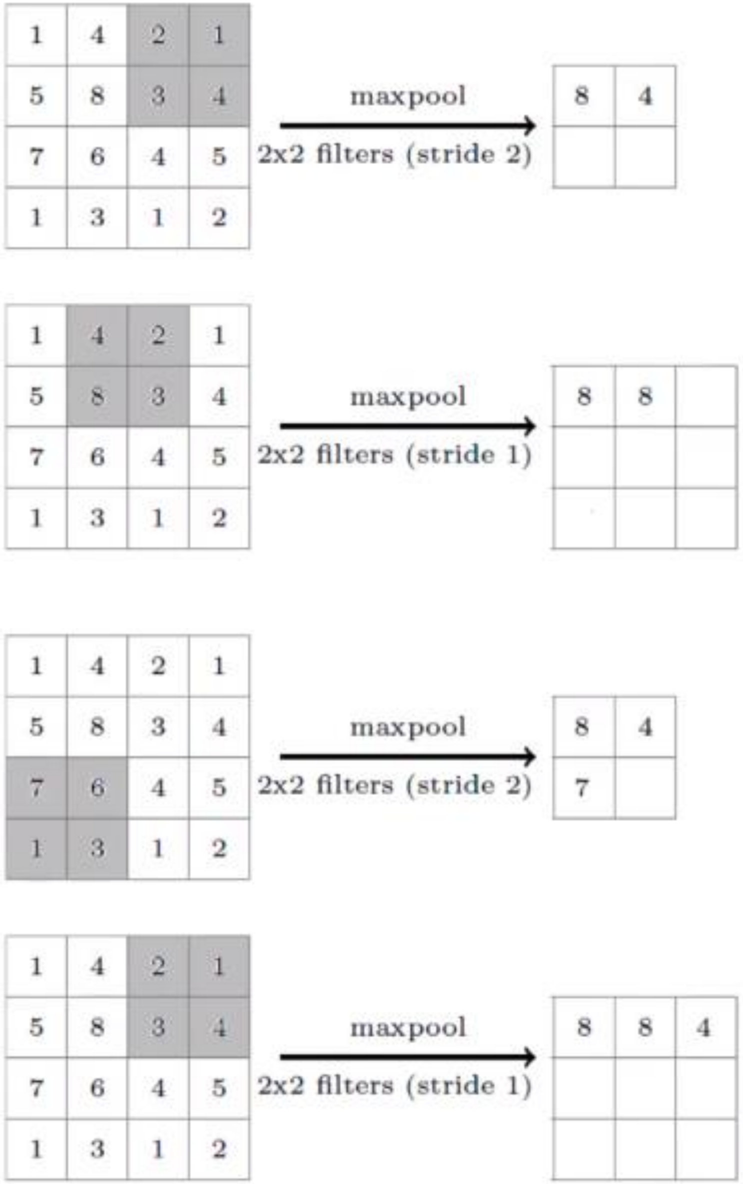
CNN


Input
Convolution Layer 1
Pooling Layer
Convolution Layer 2
Pooling Layer 2
FC 1(120)
FC 2(84)
Output(10)
32
32
28
28
14
14
10
10
5
5
$S = 1, F = 5,$
$K = 6, P = 0,$
$Param = 150$
$S = 1, F = 2,$
$K = 6, P = 0,$
$Param = 0$
$S = 1, F = 5,$
$K = 16, P = 0,$
$Param = 2400$
$S = 1, F = 2,$
$K = 16, P = 0,$
$Param = 0$
$Param = 48120$
$Param = 10164$
$Param = 850$

Pooling Layers

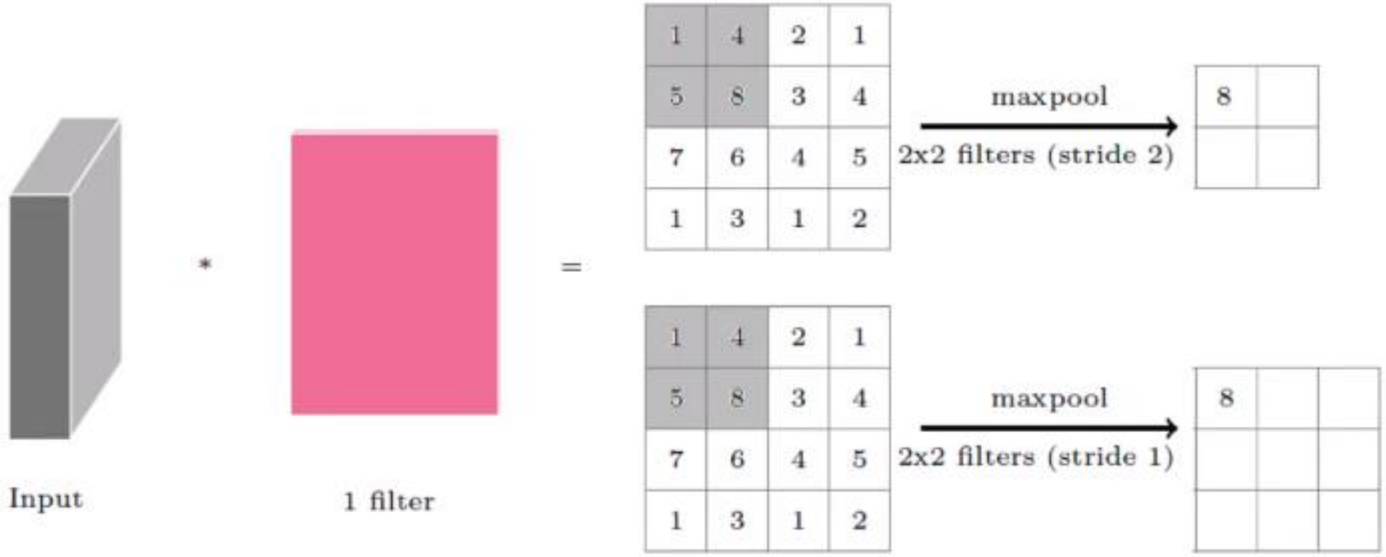
- Parameter-free down sampling

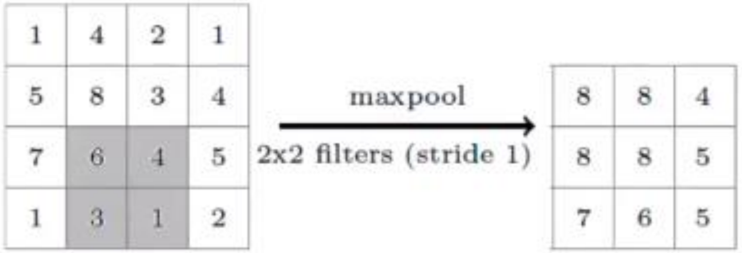


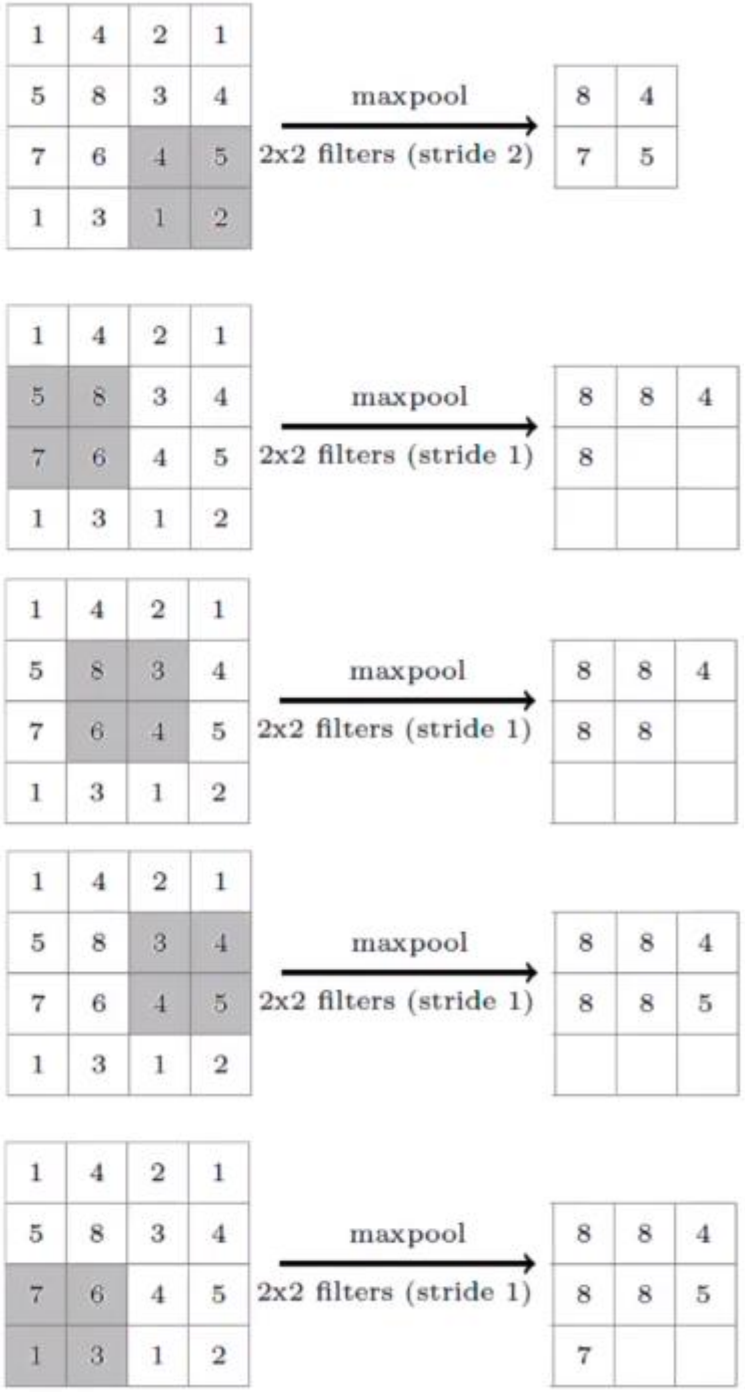


Pooling Layers

- Parameter-free down sampling





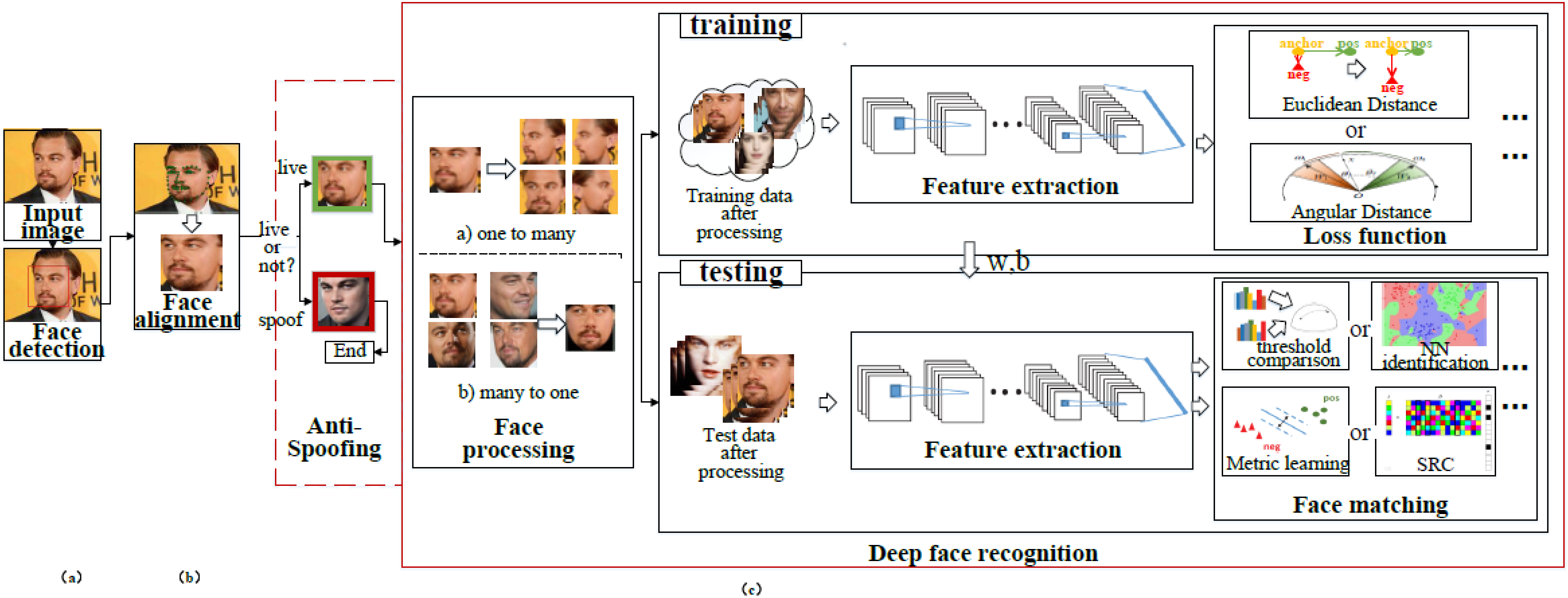


CNN in Understanding Human Faces

- Face Recognition (FR)
  - Security, Finance, Healthcare
- Face Verification (FV)

Credit: VGG Face2 Dataset

Face Recognition Pipeline

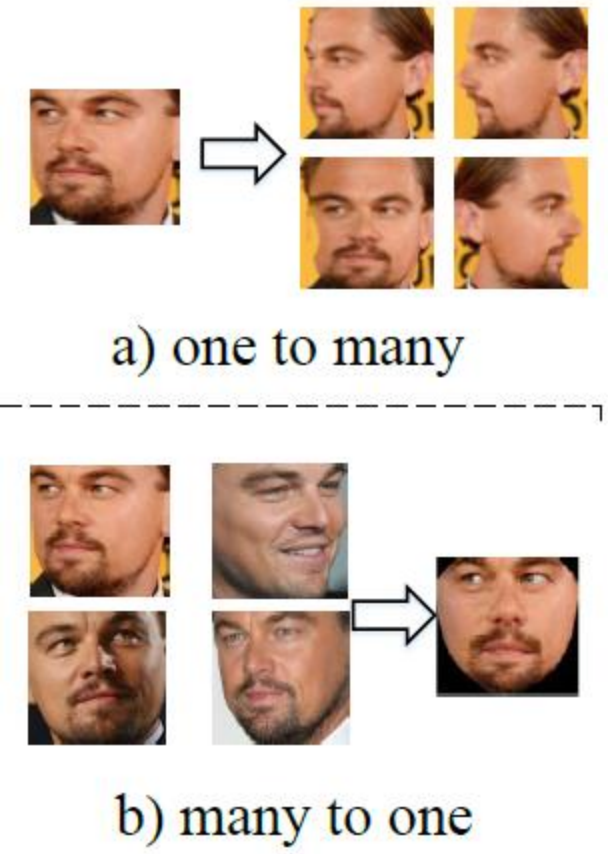


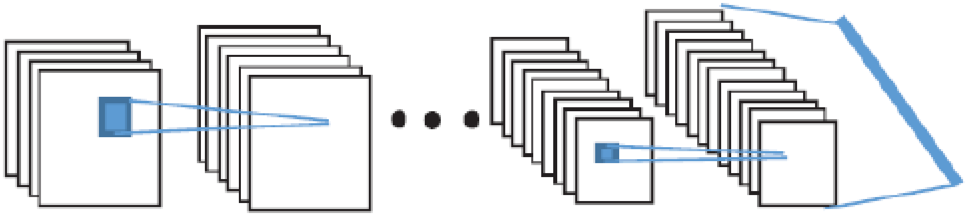
Face recognition = Face detection + Face alignment + Face matching

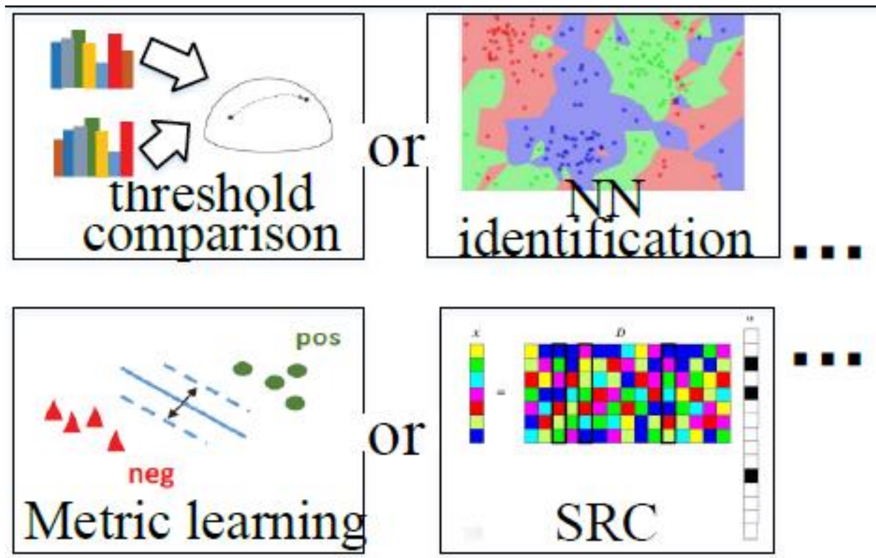
Ref. : Wang et al., Deep Face Recognition: A Survey

Components of Face Recognition System

- Face Processing
- Deep Feature Extraction
- Face Matching



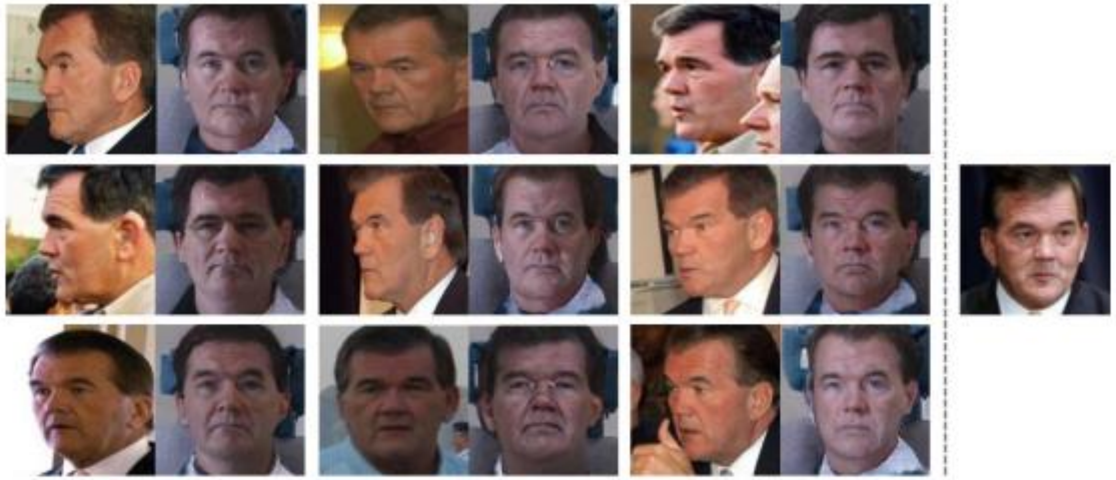




Face Processing

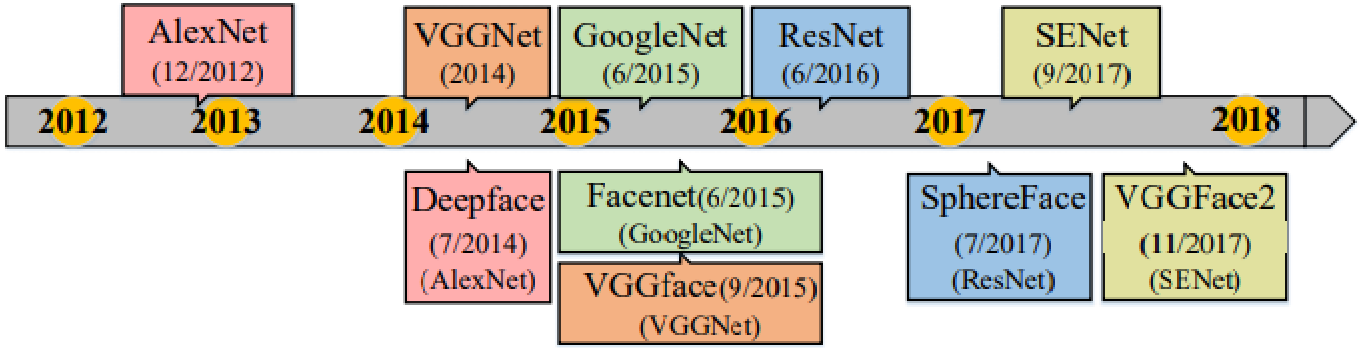
- 1 to many Augmentation
  - Ref.: Wang at al., A Survey on Face Data Augmentation
  - rotation

- Many to 1 normalization
  - Ref: Qian et al., Unsupervised Face Normalization with Extreme Pose and Expression in the Wild
  - Preserve identity despite of variations in pose, lighting, expression and background



Deep Feature Extraction

- Network Architecture
  - Ref. : Wang et al., Deep Recognition: A Survey

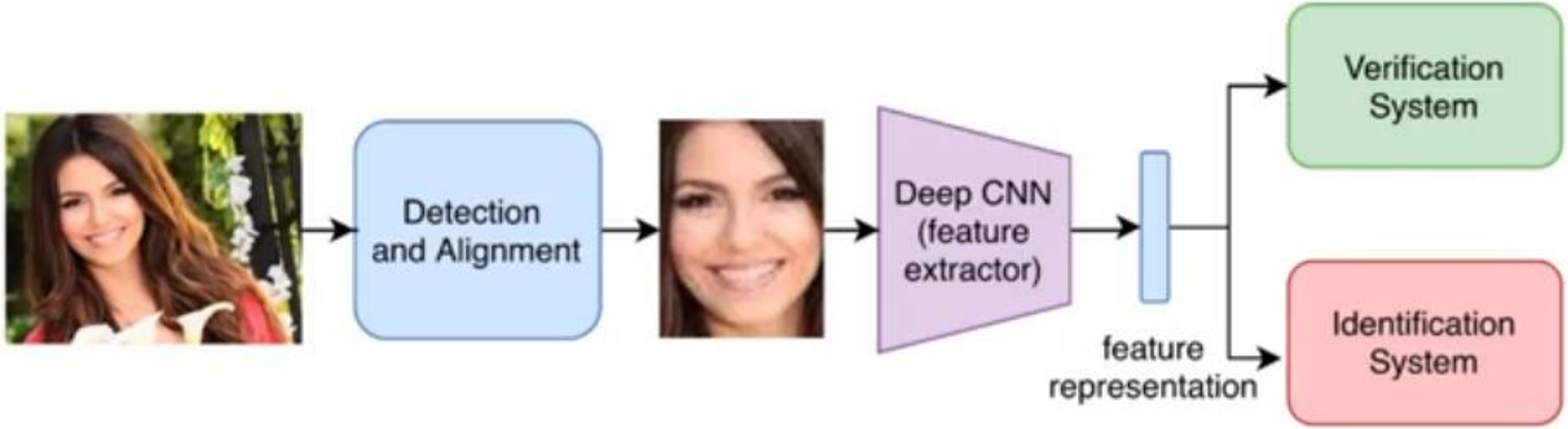


THE ACCURACY OF DIFFERENT METHODS EVALUATED ON THE LFW DATASET.

Method	Public. Time	Loss	Architecture	Number of Networks	Training Set	Accuracy±Std(%)
DeepFace [20]	2014	softmax	Alexnet	3	Facebook (4.4M,4K)	97.35±0.25
DeepID2 [21]	2014	contrastive loss	Alexnet	25	CelebFaces+ (0.2M,10K)	99.15±0.13
DeepID3 [36]	2015	contrastive loss	VGGNet-10	50	CelebFaces+ (0.2M,10K)	99.53±0.10
FaceNet [38]	2015	triplet loss	GoogleNet-24	1	Google (500M,10M)	99.63±0.09
Baidu [58]	2015	triplet loss	CNN-9	10	Baidu (1.2M,18K)	99.77
VGGface [37]	2015	triplet loss	VGGNet-16	1	VGGface (2.6M,2.6K)	98.95
light-CNN [85]	2015	softmax	light CNN	1	MS-Celeb-1M (8.4M,100K)	98.8
Center Loss [101]	2016	center loss	Lenet+-7	1	CASIA-WebFace, CACD2000, Celebrity+ (0.7M,17K)	99.28
L-softmax [104]	2016	L-softmax	VGGNet-18	1	CASIA-WebFace (0.49M,10K)	98.71
Range Loss [82]	2016	range loss	VGGNet-16	1	MS-Celeb-1M, CASIA-WebFace (5M,100K)	99.52
L2-softmax [109]	2017	L2-softmax	ResNet-101	1	MS-Celeb-1M (3.7M,58K)	99.78
Normface [110]	2017	contrastive loss	ResNet-28	1	CASIA-WebFace (0.49M,10K)	99.19
CoCo loss [112]	2017	CoCo loss	-	1	MS-Celeb-1M (3M,80K)	99.86
vMF loss [115]	2017	vMF loss	ResNet-27	1	MS-Celeb-1M (4.6M,60K)	99.58
Marginal Loss [116]	2017	marginal loss	ResNet-27	1	MS-Celeb-1M (4M,80K)	99.48
SphereFace [84]	2017	A-softmax	ResNet-64	1	CASIA-WebFace (0.49M,10K)	99.42
CCL [113]	2018	center invariant loss	ResNet-27	1	CASIA-WebFace (0.49M,10K)	99.12
AMS loss [105]	2018	AMS loss	ResNet-20	1	CASIA-WebFace (0.49M,10K)	99.12
Cosface [107]	2018	cosface	ResNet-64	1	CASIA-WebFace (0.49M,10K)	99.33
Arcface [106]	2018	arcface	ResNet-100	1	MS-Celeb-1M (3.8M,85K)	99.83
Ring loss [117]	2018	Ring loss	ResNet-64	1	MS-Celeb-1M (3.5M,31K)	99.50

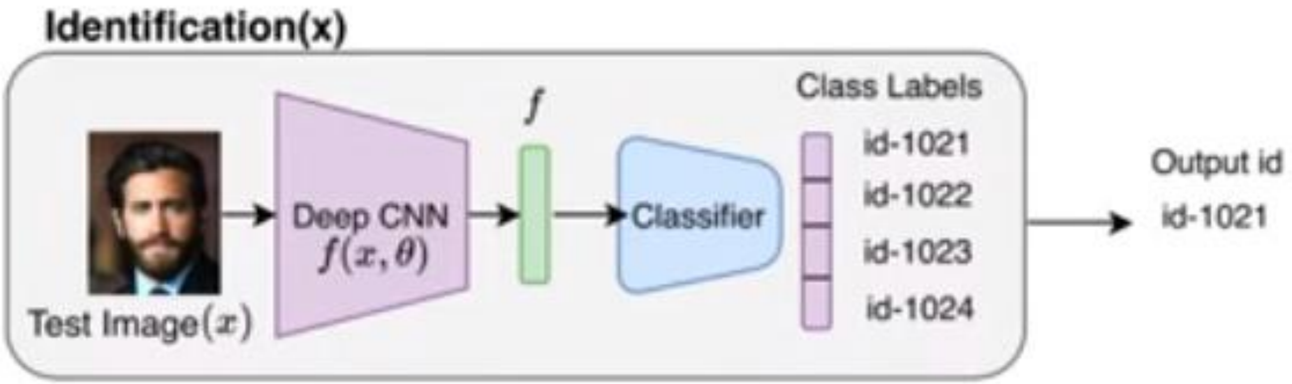
Face Recognition

- Face Identification
- Face Verification



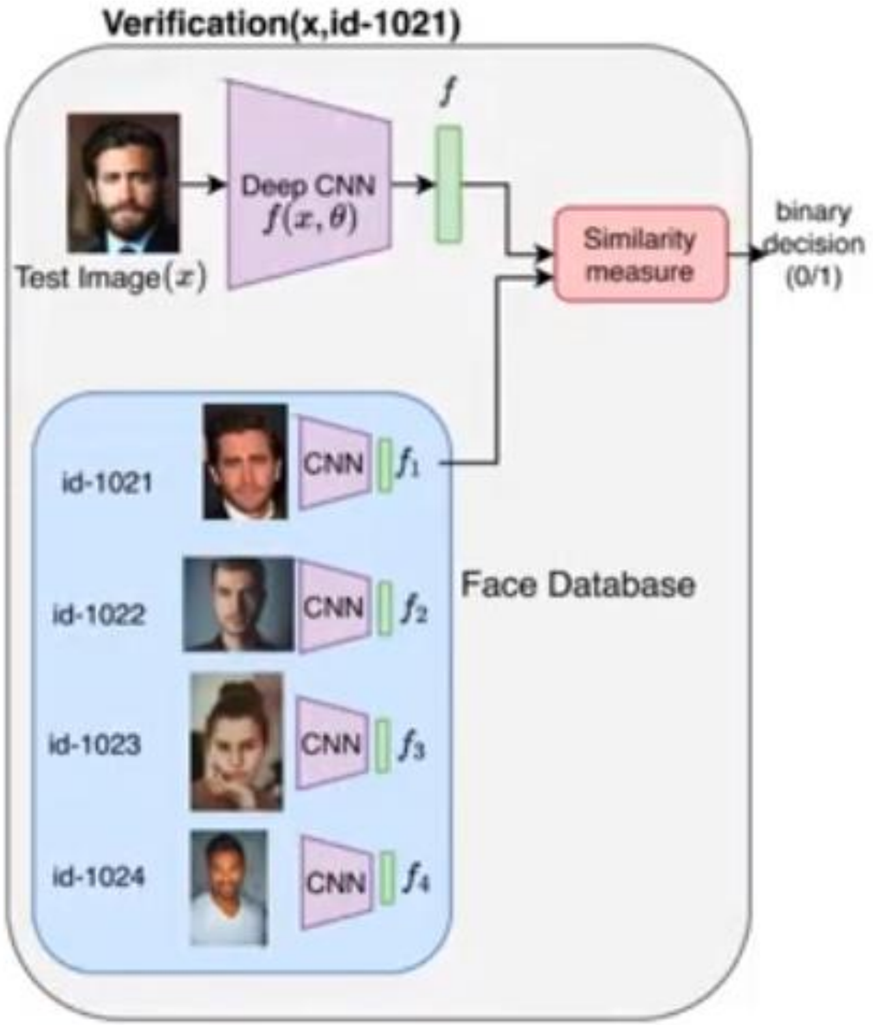
Face Identification

- Assign input image to person name/ID from the database
- K+1 multi-class classification
    - 1 additional class for unrecognized faces
- Input: Face Image and Output: Identity class/ Face ID



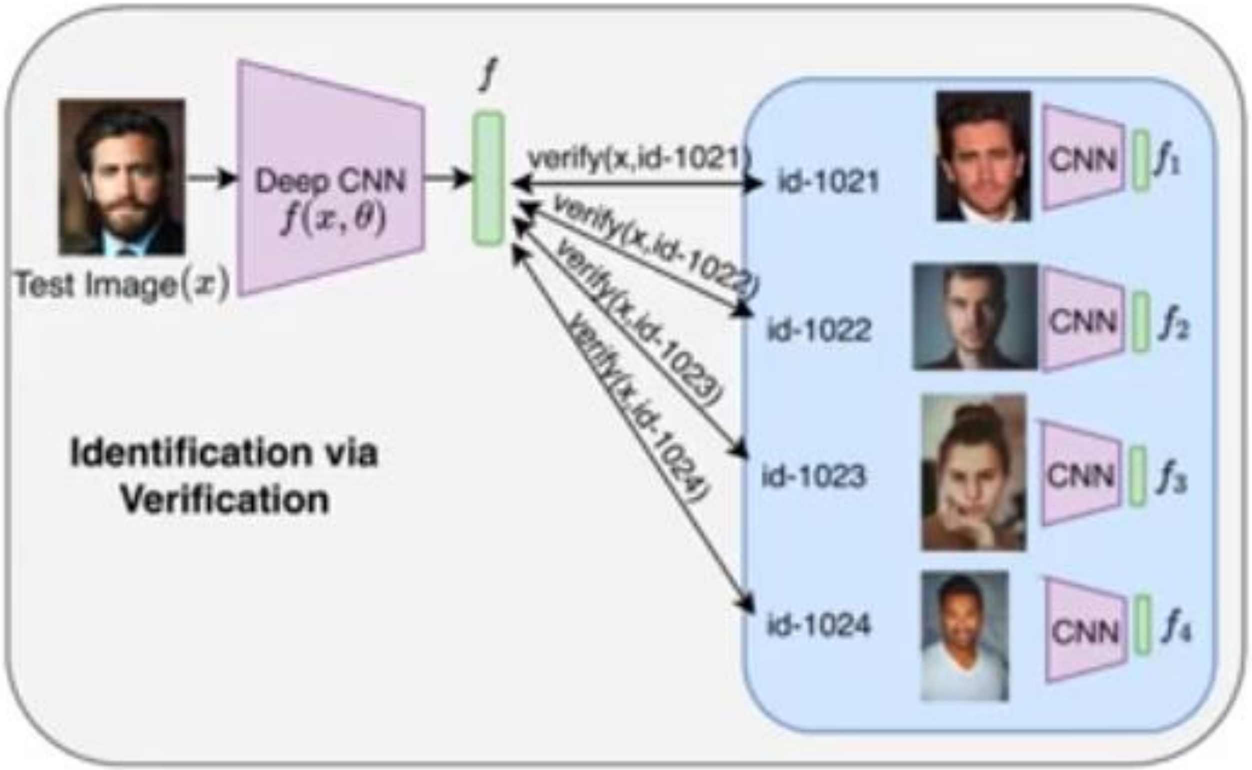
Face Verification

- Verifying whether two images belong to the same ID
- 1 to 1 matching
- Input: Face image + ID
- Output: Match/Not match



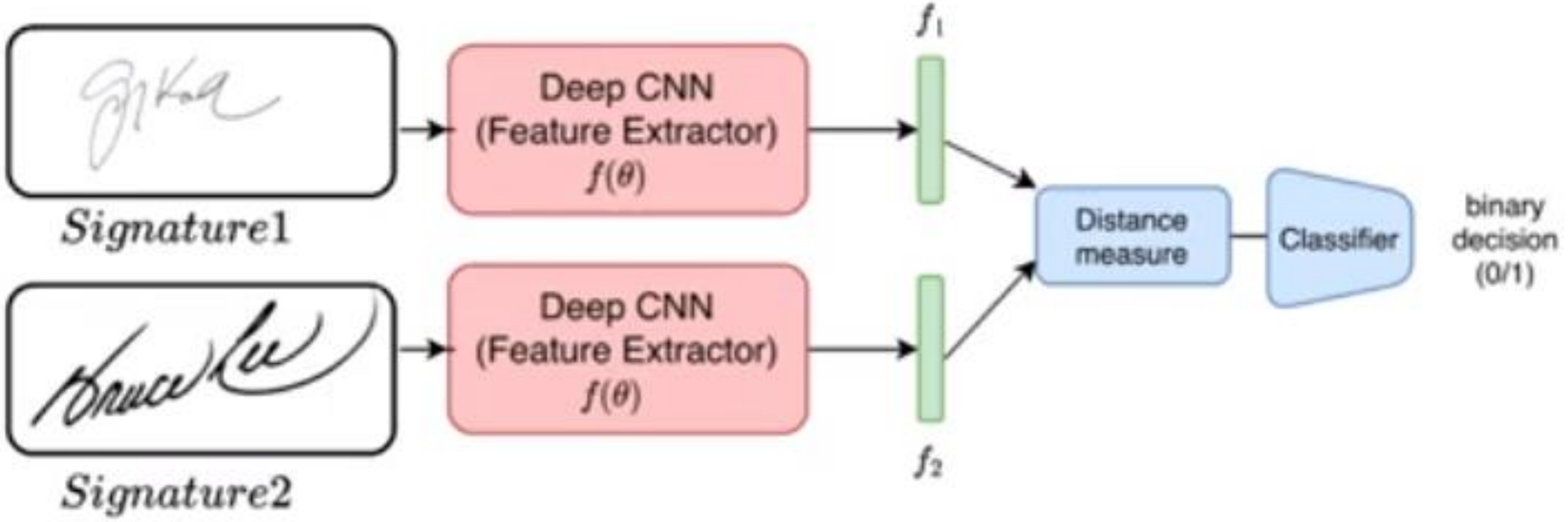
Identification via Verification

- Removes retraining
  - Scalable
- Goal: build an accurate/ efficient verification system



Verification: Siamese Network

- Proposed in 1994
- Two replicas of same CNN architecture parameterized with same weights



- Ref: Bromley et al., Signature verification using a Siamese Time Delay Neural Network, NIPS, 1994

DeepFace: Identification

- Step 1: Frontal crop of face
- Ref: Taigman et al., DeepFace: Closing the Gap to Human-Level Performance in Face Verification

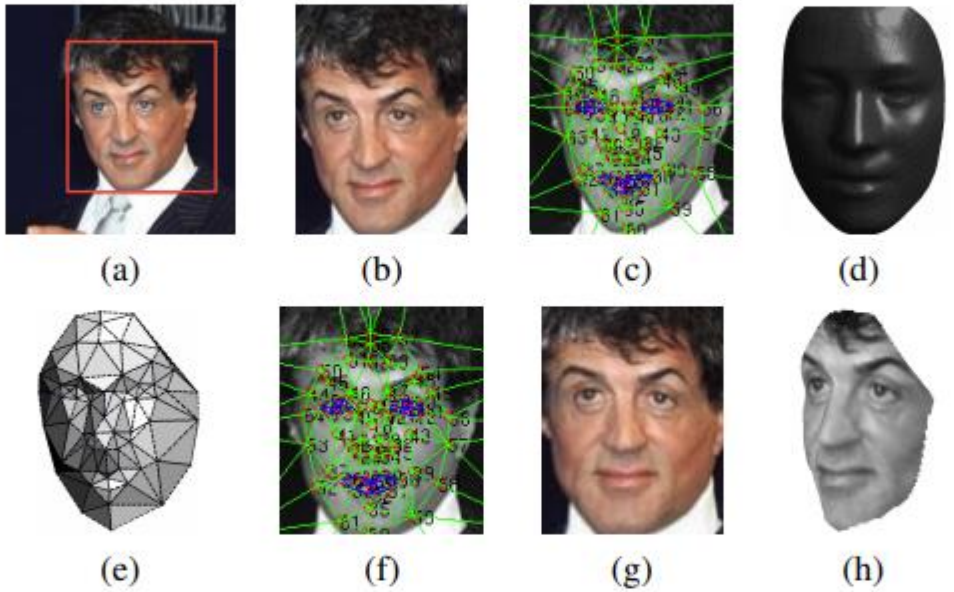


Figure 1. **Alignment pipeline.** (a) The detected face, with 6 initial fiducial points. (b) The induced 2D-aligned crop. (c) 67 fiducial points on the 2D-aligned crop with their corresponding Delaunay triangulation, we added triangles on the contour to avoid discontinuities. (d) The reference 3D shape transformed to the 2D-aligned crop image-plane. (e) Triangle visibility w.r.t. to the fitted 3D-2D camera; darker triangles are less visible. (f) The 67 fiducial points induced by the 3D model that are used to direct the piece-wise affine warpping. (g) The final frontalized crop. (h) A new view generated by the 3D model (not used in this paper).

DeepFace: Identification

- Step 2: Frontal crop passed for identification to deep CNN with K-way softmax

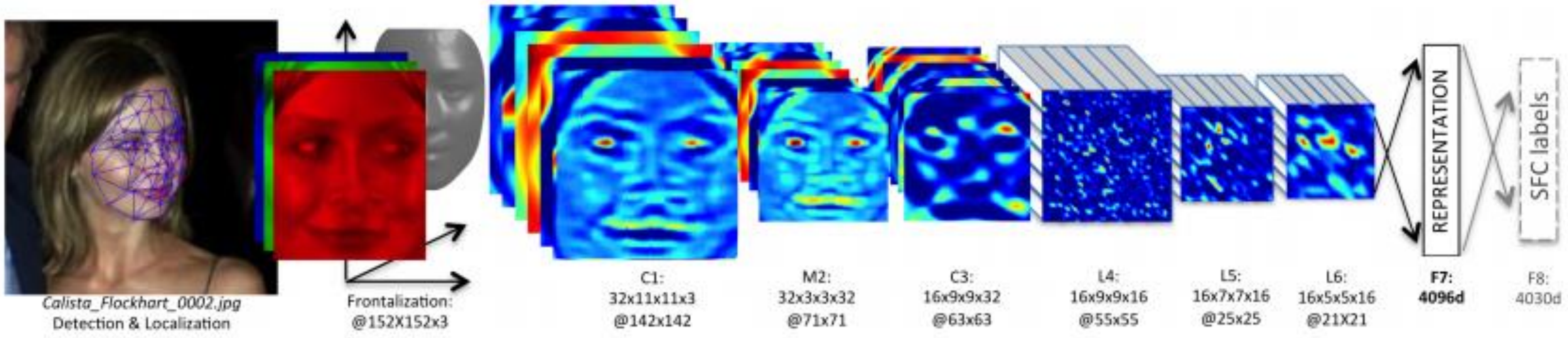
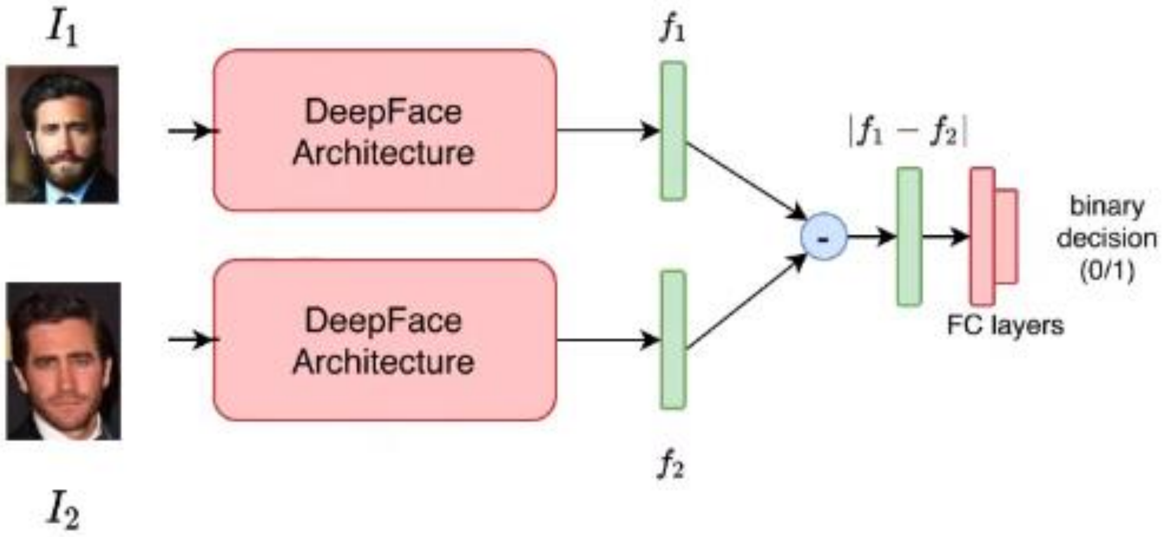


Figure 2. **Outline of the *DeepFace* architecture**. A front-end of a single convolution-pooling-convolution filtering on the rectified input, followed by three locally-connected layers and two fully-connected layers. Colors illustrate feature maps produced at each layer. The net includes more than 120 million parameters, where more than 95% come from the local and fully connected layers.

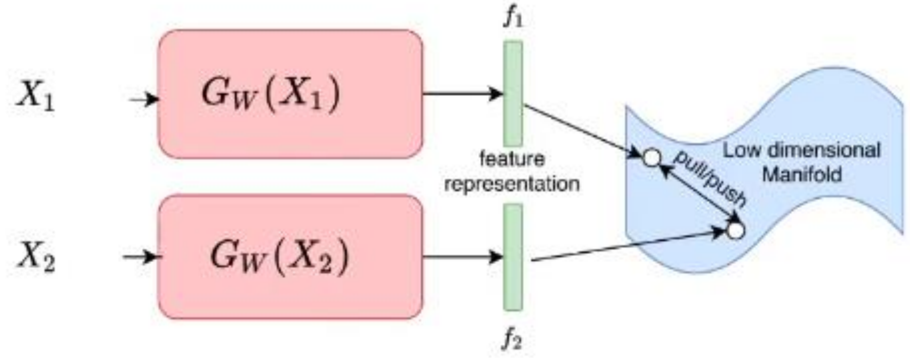
Ref: Taigman et al., DeepFace: Closing the Gap to Human-Level Performance in Face Verification

DeepFace Verification: Siamese Network

- Classification parameters are frozen
- Representations learnt from identification are used for verification
- Network trained by taking the absolute difference between features, followed by a fully connected network
- Distance Induced:
  - $d(f_1, f_2) = \sum_i \alpha_i \, |f_1[i], f_2[i]|$
  - α_i are trainable parameters
- Output:
  - Binary decision (match/not match)



Contrastive Loss



- Based on metric learning paradigm
- Learn distinctive discriminative feature representations
- Objective: Map input to embedding space where distance between the points corresponds to semantic similarity between the input points
- Hadsell et al. introduced an approach called pairwise contrastive loss, where similar points in the input space are mapped to nearby points on a lower dimensional manifold.
- Goal: learn W such that $D_W(X_1, X_2)$, approximates the semantic similarity of inputs
- Ref: Hadsell et al., Dimensionality Reduction by Learning an Invariant Mapping, 2006

Pairwise Contrastive Loss

- Let X_1, X_2 be high dimensional input vectors
- y – Binary label assigned to thi pair

$$\begin{cases} y = 0, if\, X_1, X_2\ are\ similar \\ y = 1, Otherwise \end{cases}$$

- Given (W, y, X_1, X_2), pairwise contrastive loss is given by:
- $L_{contrastive}(W, y, X_1, X_2) = \frac{1-y}{2} D_W^2 + \frac{y}{2} max(0, m - D_W^2)$

Step 1: For each input sample $\vec{X}_i$, do the following:

(a) Using prior knowledge find the set of samples $\mathcal{S}_{\vec{X}_i} = \{\vec{X}_j\}_{j=1}^{p}$, such that $\vec{X}_j$ is deemed similar to $\vec{X}_i$.

(b) Pair the sample $\vec{X}_i$ with all the other training samples and label the pairs so that: $Y_{ij} = 0$ if $\vec{X}_j \in \mathcal{S}_{\vec{X}_i}$, and $Y_{ij} = 1$ otherwise.

Combine all the pairs to form the labeled training set.

Step 2: Repeat until convergence:

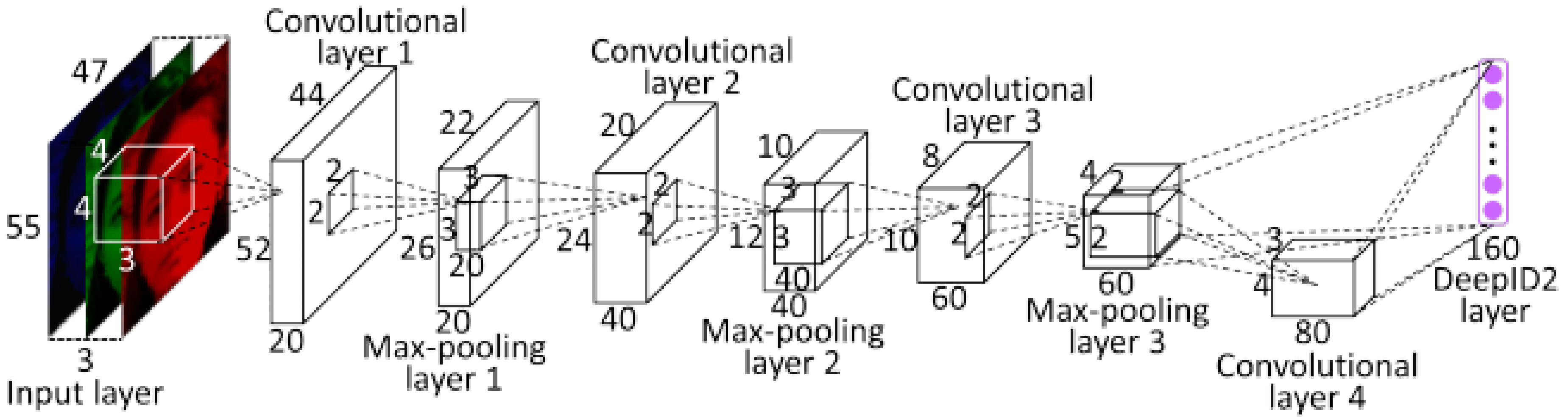
(a) For each pair $(\vec{X}_i, \vec{X}_j)$ in the training set, do

i. If $Y_{ij} = 0$, then update W to decrease $D_W = \|G_W(\vec{X}_i) - G_W(\vec{X}_j)\|_2$

ii. If $Y_{ij} = 1$, then update W to increase $D_W = \|G_W(\vec{X}_i) - G_W(\vec{X}_j)\|_2$

DeepID2

- Trains a deep CNN to jointly perform identification and verification
- Identification task: Increases inter-personal variations by pushing features from different IDs apart
- Verification Task: Reduces intra-personal variations by pulling features from same ID together



DeepID2

- Cross Entropy Loss: for training identification parameters, θ_{id}

$$\mathrm{Ident}(f, t, \theta_{id}) = -\sum_{i=1}^{n} -p_i \log \hat{p}_i = -\log \hat{p}_t$$

  - f: DeepID2 feature vector
  - t: Target Class
  - θ_{id}: Softmax layer parameters
  - p_i: Target probability distribution
  - $\hat{p}_i$: Predicted probability distribution
- Pairwise Contrastive Loss: for training verification parameters, θ_{ve}

$$\mathrm{Verif}(f_i, f_j, y_{ij}, \theta_{ve}) = \begin{cases} \frac{1}{2}\left\|f_i - f_j\right\|_2^2 & \text{if } y_{ij} = 1 \\ \frac{1}{2}\max\left(0, m - \left\|f_i - f_j\right\|_2\right)^2 & \text{if } y_{ij} = -1 \end{cases}$$

f_i and f_j are DeepID2 vectors extracted from the two face images in comparison

$\theta_{ve} = \{m\}$ is the parameter to be learned in the verification loss function

$y_{ij} = 1$ means that f_i and f_j are from the same identity

$y_{ij} = -1$ means different identities.

DeepID2: Learning Algorithm

The DeepID2 learning algorithm

input: training set $\chi = \{(x_i, l_i)\}$, initialized parameters θ_c, θ_{id}, and θ_{ve}, hyperparameter λ, learning rate $\eta(t)$, $t \leftarrow 0$

while not converge **do**

$t \leftarrow t + 1$ sample two training samples (x_i, l_i) and (x_j, l_j) from χ

$f_i = \text{Conv}(x_i, \theta_c)$ and $f_j = \text{Conv}(x_j, \theta_c)$

$\nabla\theta_{id} = \frac{\partial \text{Ident}(f_i, l_i, \theta_{id})}{\partial \theta_{id}} + \frac{\partial \text{Ident}(f_j, l_j, \theta_{id})}{\partial \theta_{id}}$

$\nabla\theta_{ve} = \lambda \cdot \frac{\partial \text{Verif}(f_i, f_j, y_{ij}, \theta_{ve})}{\partial \theta_{ve}}$, where $y_{ij} = 1$ if $l_i = l_j$, and $y_{ij} = -1$ otherwise.

$\nabla f_i = \frac{\partial \text{Ident}(f_i, l_i, \theta_{id})}{\partial f_i} + \lambda \cdot \frac{\partial \text{Verif}(f_i, f_j, y_{ij}, \theta_{ve})}{\partial f_i}$

$\nabla f_j = \frac{\partial \text{Ident}(f_j, l_j, \theta_{id})}{\partial f_j} + \lambda \cdot \frac{\partial \text{Verif}(f_i, f_j, y_{ij}, \theta_{ve})}{\partial f_j}$

$\nabla\theta_c = \nabla f_i \cdot \frac{\partial \text{Conv}(x_i, \theta_c)}{\partial \theta_c} + \nabla f_j \cdot \frac{\partial \text{Conv}(x_j, \theta_c)}{\partial \theta_c}$

update $\theta_{id} = \theta_{id} - \eta(t) \cdot \theta_{id}$, $\theta_{ve} = \theta_{ve} - \eta(t) \cdot \theta_{ve}$, and $\theta_c = \theta_c - \eta(t) \cdot \theta_c$.

end while

output θ_c

FaceNet: Triplet Loss

ensure that an image x_i^a (*anchor*) of a specific person is closer to all other images x_i^p (*positive*) of the same person than it is to any image x_i^n (*negative*) of any other person

Thus we want,

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2 \,,$$

$$\forall\, (f(x_i^a), f(x_i^p), f(x_i^n)) \in \mathcal{T} \,.$$

where α is a margin that is enforced between positive and negative pairs. $\mathcal{T}$ is the set of all possible triplets in the training set

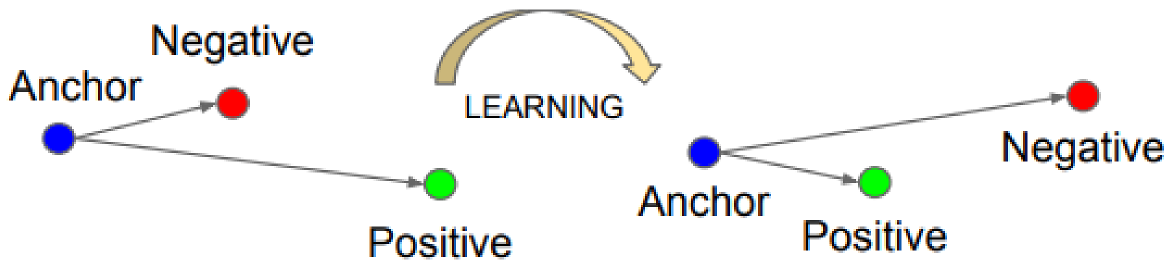


Figure . The **Triplet Loss** minimizes the distance between an *anchor* and a *positive*, both of which have the same identity, and maximizes the distance between the *anchor* and a *negative* of a different identity.

$$\sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]$$

- Ref: Schroff et al, FaceNet: A Unified Embedding for Face Recognition and Clustering, 2015.

FaceNet

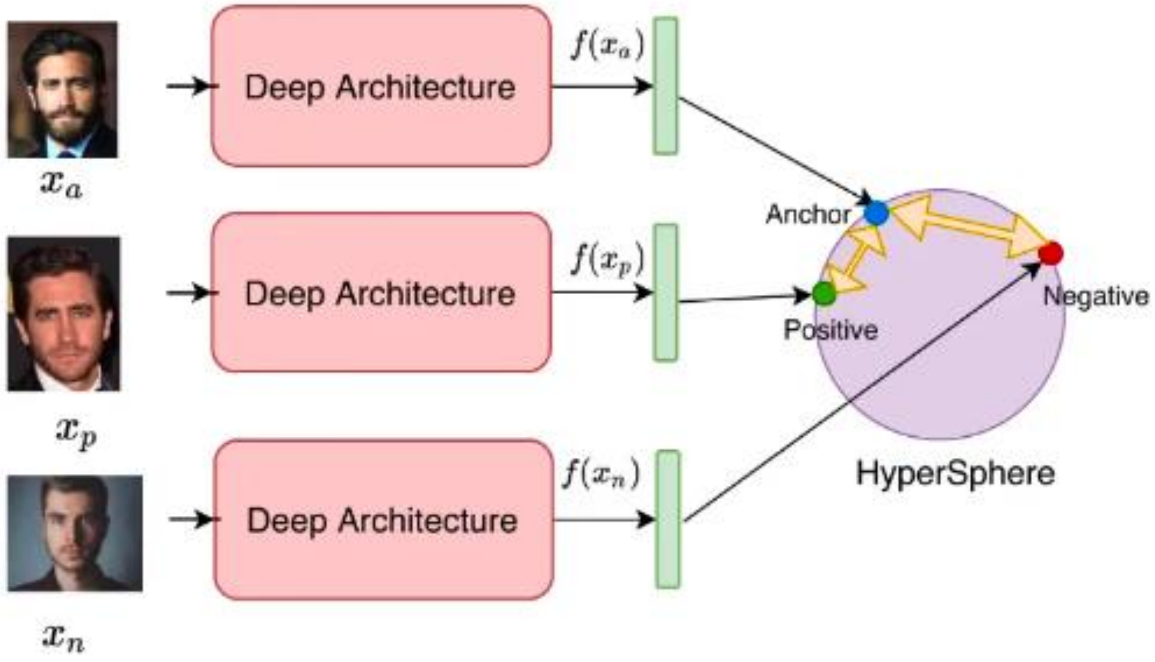
- Let $f(x)$: representation/embedding on d-dimensional hypersphere, s.t. $\|f(x)\|_2 = 1$
- Objective:
  - train θ to ensure that for all triplets x_a(anchor), x_p(positive), x_n(negative):

    $$||f(x_a) - f(x_p)||_2^2 + \alpha < ||f(x_a) - f(x_n)||_2^2$$

    where α is margin

    Achieved by training parameters θ to minimize:

    $$L_{triplet} = \sum_i [||f(x_a^i) - f(x_p^p)||_2^2 - ||f(x_a^i) - f(x_n^i)||_2^2 + \alpha]$$



Recent Efforts

CosFace:

Wang et al, CosFace: Large Margin Cosine Loss for Deep Face Recognition, CVPR 2018

UniformFace:

Duan et al, UniformFace: Learning Deep Equidistributed Representation for Face Recognition, CVPR 2019

RegularFace:

Zhao et al, RegularFace: Deep Face Recognition via Exclusive Regularization, CVPR 2019

GroupFace:

Kim et al, GroupFace: Learning Latent Groups and Constructing Group-based Representations for Face Recognition , CVPR 2020

CurricularFace:

Huang et al, CurricularFace: Adaptive Curriculum Learning Loss for Deep Face Recognition, CVPR 2020

Thank You!