# *Face verification?*

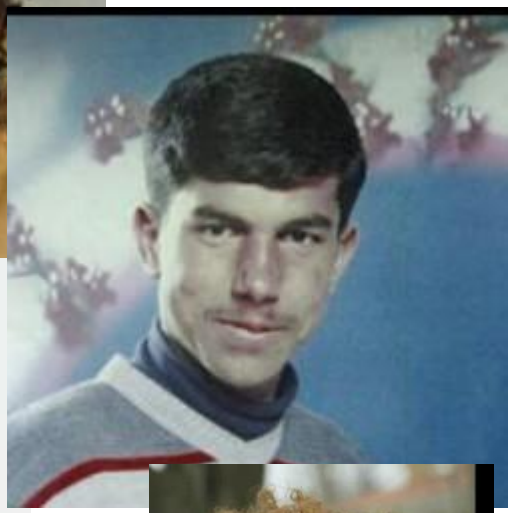
*v.s.*

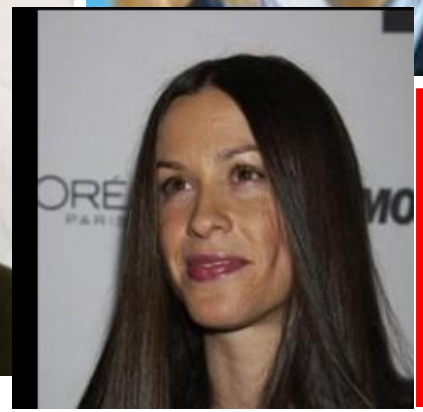
***Is or Not the same person?***

# *Face Recognition?*

***Who am I?***

***(mutli-times verification)***





# *Challenge ?*

## *1. Wild enviroment*

*illumination*


*occlusion*

*pose*

# *Challenge ?*

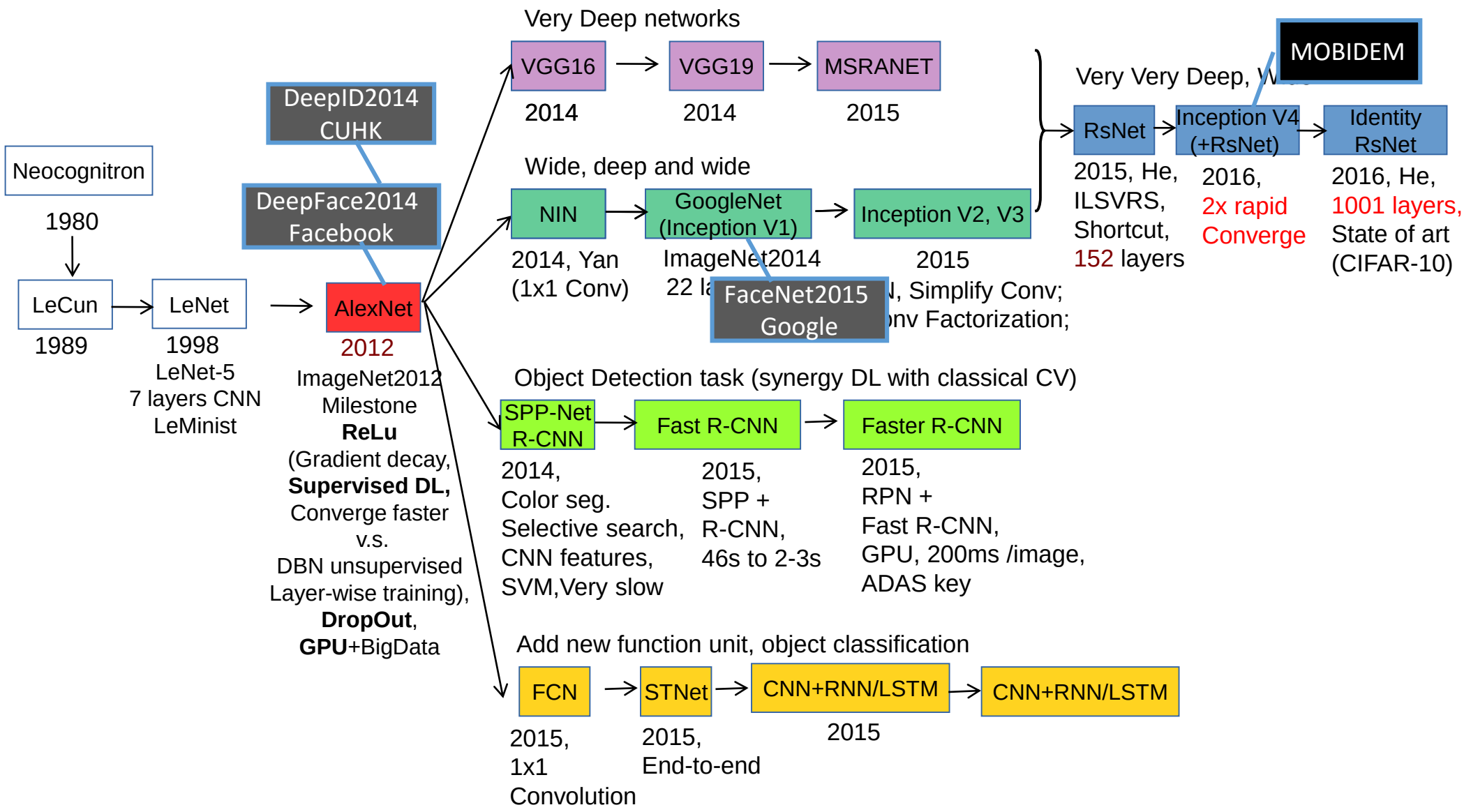
## *2. Anti-fraud*


***photo of object***

***mask***

***makeup***

# State of art of Deep CNN for face verification / recognition



# How to do ?

## 1. Face verification protocol

Face registration

Face Alignment

Feature extraction

classifier

Yes or No

Face boundary

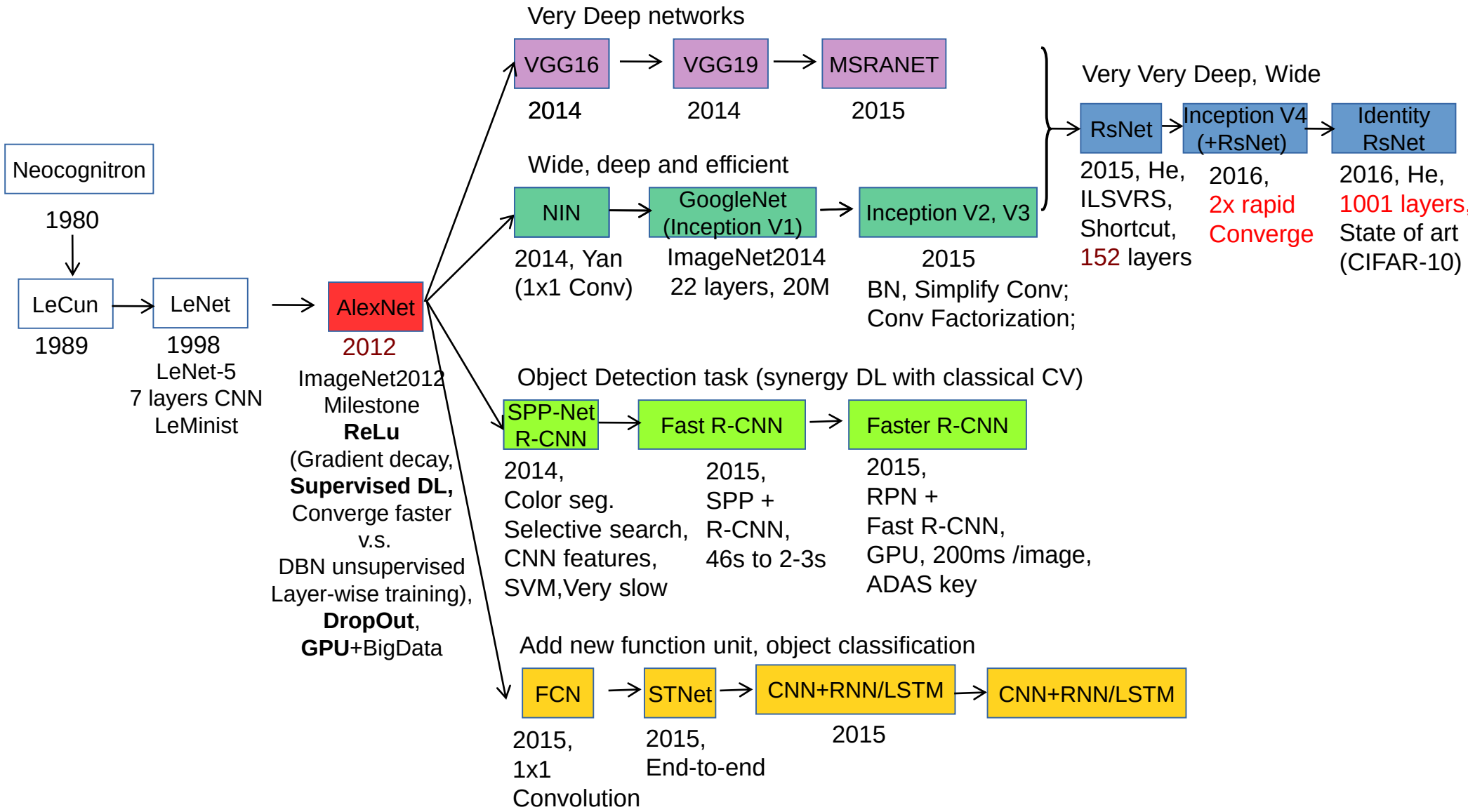
- Viola & Jones
- Constrained local Model
- Deep learning: R-CNN

- 2D
- 3D

- Deep learning
  - CNN

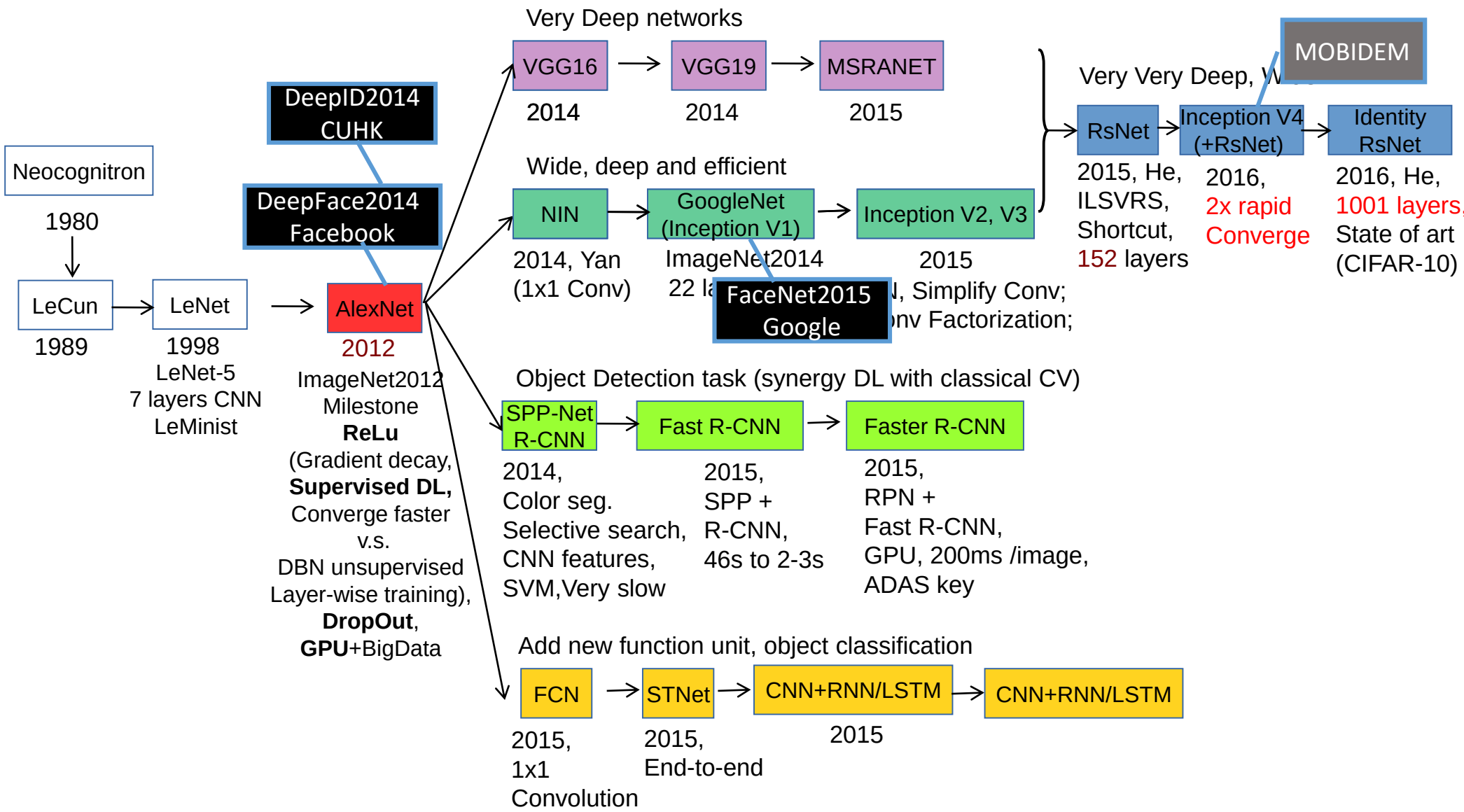
- Softmax
- SVM

# CNN development (visual object recognition)




* **Gradient decay**: **ReLU** instead of Sigmoid, **BN** (Batch Normalization ) regularization;* **Gradient overflow**: **BN (**Batch Normalization), Xavier variance;
* **Overfitting**: **Dropout** (Regularization); * **Converge slow**: **Mini-batch SGD** instead SGD (Stochastic Gradient Descent);
* **Activation function**: Sigmoid-> **ReLU->LeakReLU->PReLU->ELU->MaxOut** (solving 'dying ReLU'-gradient decay in the -x ax, 'bias shift' problem);

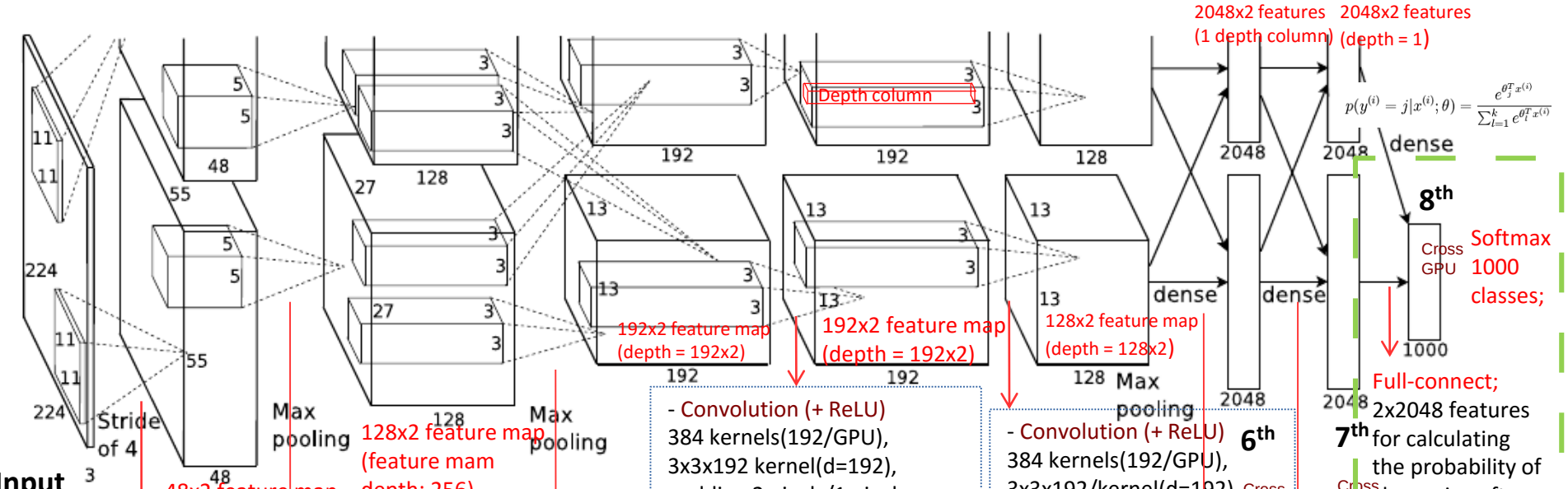
# CNN development (Spatial relations: image/supervised training)




* **Gradient decay**: **ReLU** instead of Sigmoid, **BN** (Batch Normalization ) regularization;* **Gradient overflow**: **BN (**Batch Normalization), Xavier variance;
* **Overfitting**: **Dropout** (Regularization); * **Converge slow**: **Mini-batch SGD** instead SGD (Stochastic Gradient Descent);
* **Activation function**: Sigmoid-> **ReLU->LeakReLU->PReLU->ELU->MaxOut** (solving 'dying ReLU'-gradient decay in the -x ax, 'bias shift' problem);

# AlexNet (ImageNet 2012 winner):- 5 layers convolution layers + 3 full connect layers + 1000 way softmax (1000 classes) + 60 million parameters;

**Input image**(Preprocessing: Extracted and generated artificially from original 256X256 images for data augmentation reducing overfitting)




**Input Image**
- 224x224x3 (RGB, zero-padding 3 pixels for convolution 227x227x3: (227-11)/4+1=55)

48x2 feature map (1 channels/fm, Size: 55x55x96)

**1st convolution layer**

Convolution (+ ReLU)
96 (48/GPU) kernels (kernel size:11x11x3, 3 channels/kernel, number of convolution templates: 2x48x3;
Parameters/kernel:(11x11x3+1);
Parameters: (11x11x3+1)x2x48
output Feature map size: 55x55x2x48x3),
stride/step size 4 pixels;

128x2 feature map (feature mam depth: 256)

**2nd layer**
- same GPU connection

- Response-normalization (average of 5 neighborhood kernels of the same location);
- Max overlap pooling (step=2, z=3x3, (55-3)/2+1=27);
- Convolution (+ ReLU)
256 kernels(128/GPU),
Channels/depth=48;
Parameters/kernel:(5x5x48+1);
Parameters: (5x5x48+1)x2x128;
(*padding 4 pixels, stride 1 pixel, keeping the same size before convolution? (27+4-5)/1+1*)

**3rd layer**
- cross GPU connection

- Convolution (+ ReLU)
384 kernels(192/GPU),
3x3x192 kernel(d=192),
padding 2 pixels/1 pixel s;
Paras: (3x3x192+1)x2x192

**4th layer**
- same GPU connection

- Response-normalization
- Max overlap pooling
(step=2, z=3x3, (27-3)/2+1=13);
- Convolution (+ ReLU)
384 kernels(192/GPU),
3x3x256 /kernel(channel/depth=256)
(cross GPU connection/full connection);
Parameters/kernel: (3x3+1)x256;
Parameters: (3x3x2x128+1)x2x192
(*padding 2 pixels, stride 1 pixel, (13+2-3)/1+1=13, keeping size*)

**5th layer**
- same GPU connection

- Convolution (+ ReLU)
384 kernels(192/GPU),
3x3x192/kernel(d=192)
Paras: (3x3x192+1)x2x128

- Max overlap pooling
- Vectorization (output 1x1 feature map = neurons;
Using kernels have the same side size as input feature map 13x13)
Full-connected layer (channel/depth =2x128), 4096 kernel (2048/GPU),
Parameters (full connect): (13x13+1)x(2x128)x2x2048

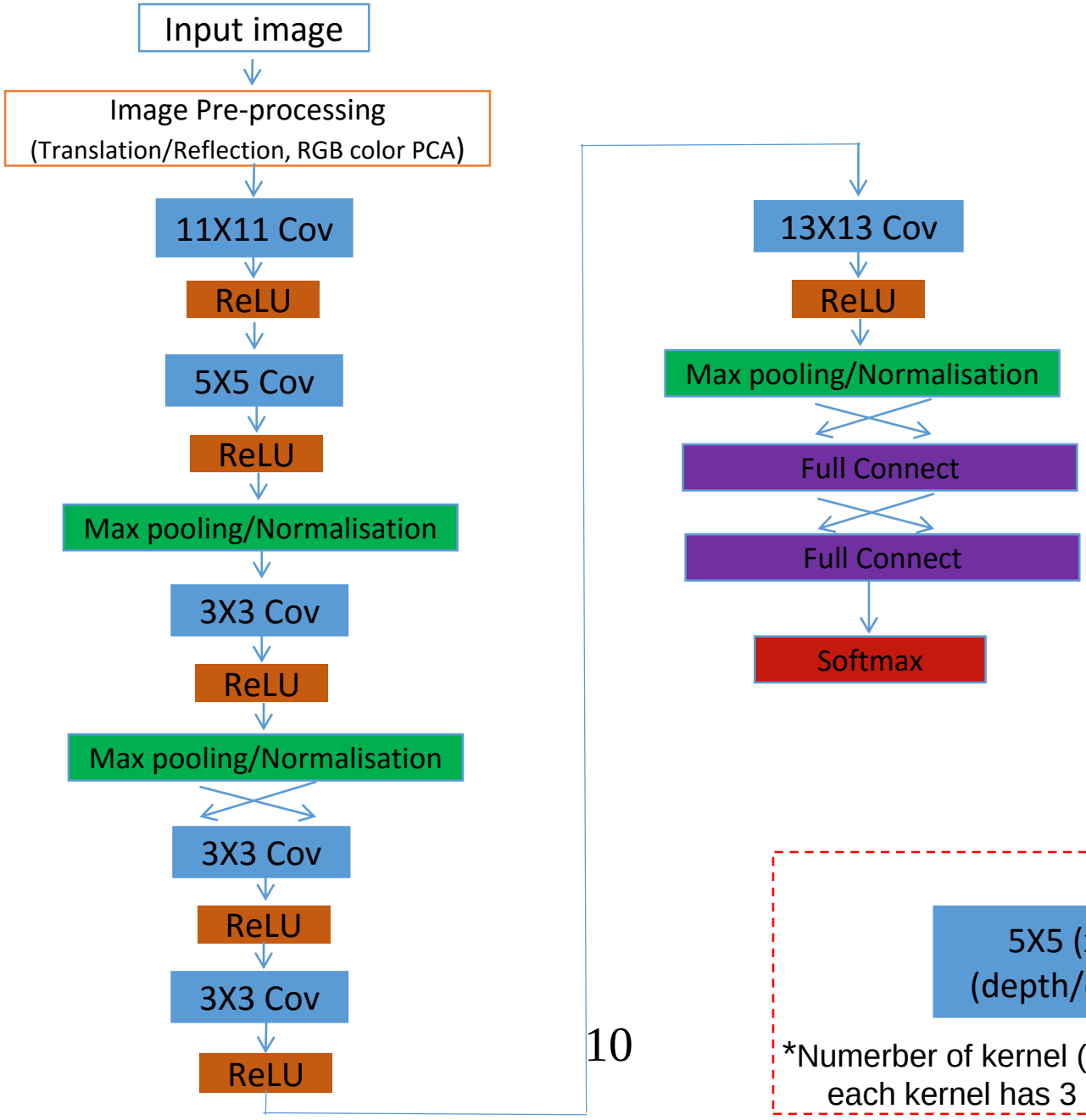
# Protocol of AlexNet

Training: Back-propragation
(Gradient Descente)

(Pre-training:
- Unsupervised Autocoder;)

Training (BP):
- Dropout;
- Mini-batch SGD
  - learning rate;
  - momentum;
  - weight decay;

Initialization:
- Weights: 0-0.01 Gaussian;
- Bias: 0 (1,3), 1 (2,4,5);
- Learning rate: 0.01(/10, 3)
- Batch size: 128 examples;
- Momentum factor: 0.9
- Weight decay: 0.0005

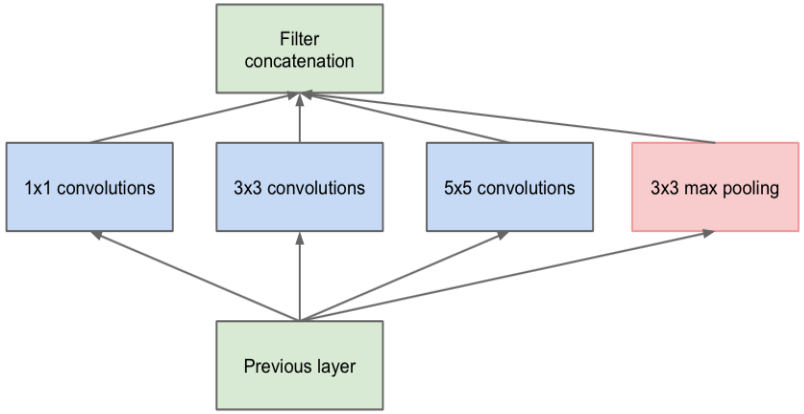


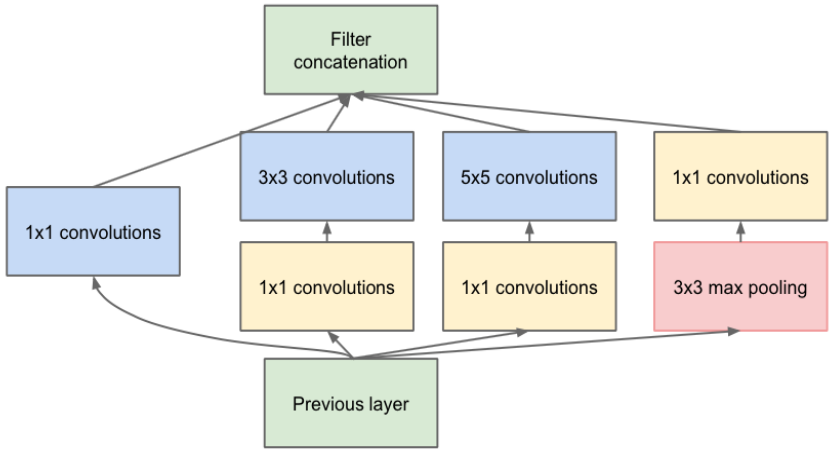

5X5 (x48) Cov
(depth/channel=3)

*Numerber of kernel (5x5 conv) is 48, each kernel has 3 channels/depth

# GoogLeNet

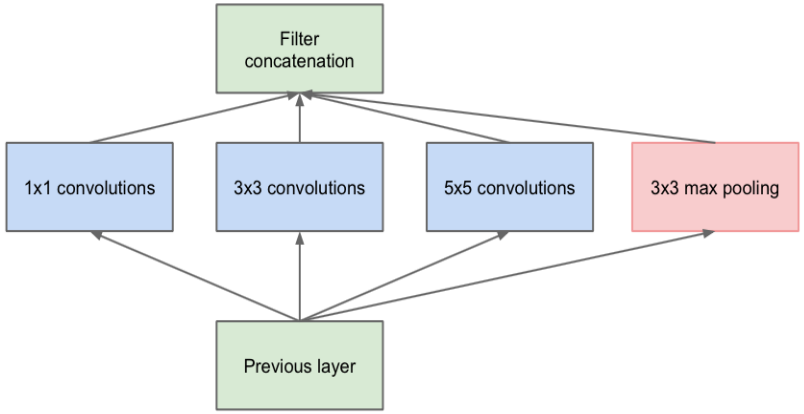
**(wide, conv conection  ,1X1 conv, rapide)**

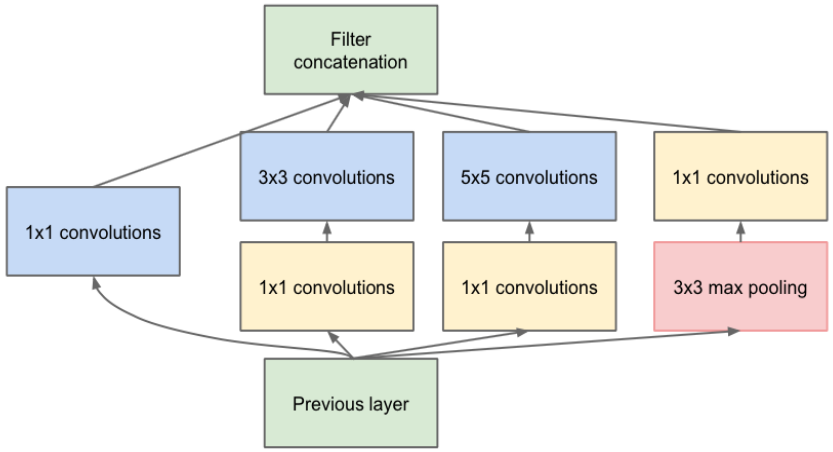


# GoogLeNet

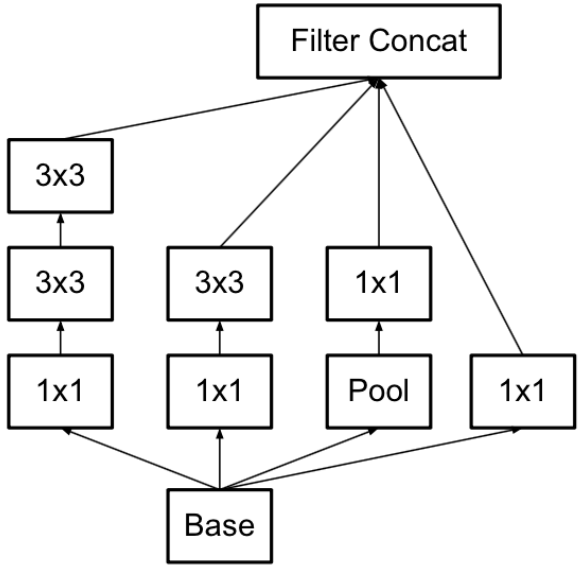
**(wide, conv conection ,1X1 conv, rapide)**

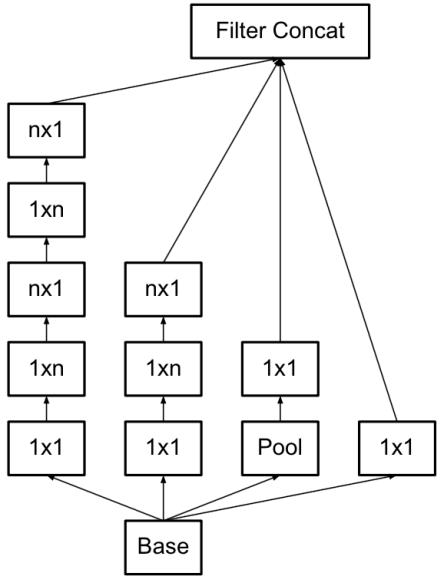


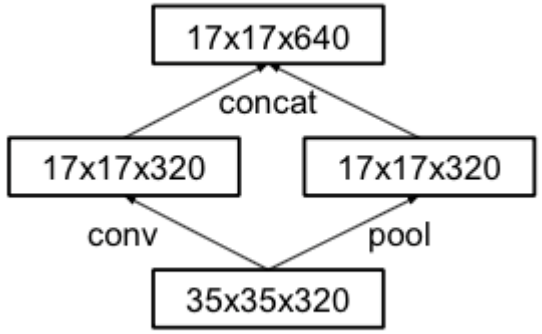




# Inception V2,V3

**(Simplify the conv,decomposition)**

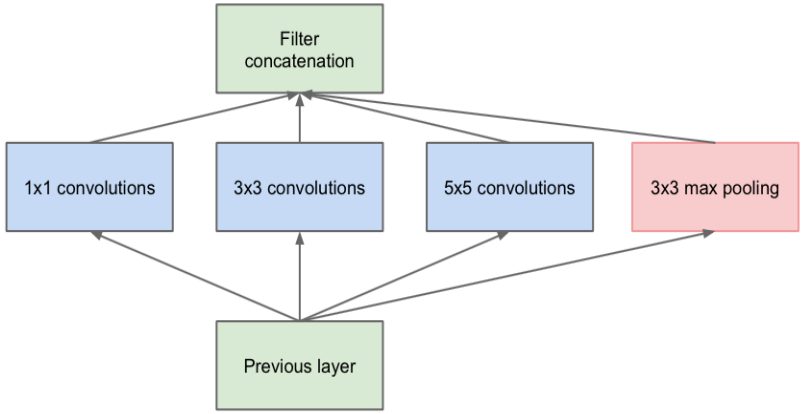


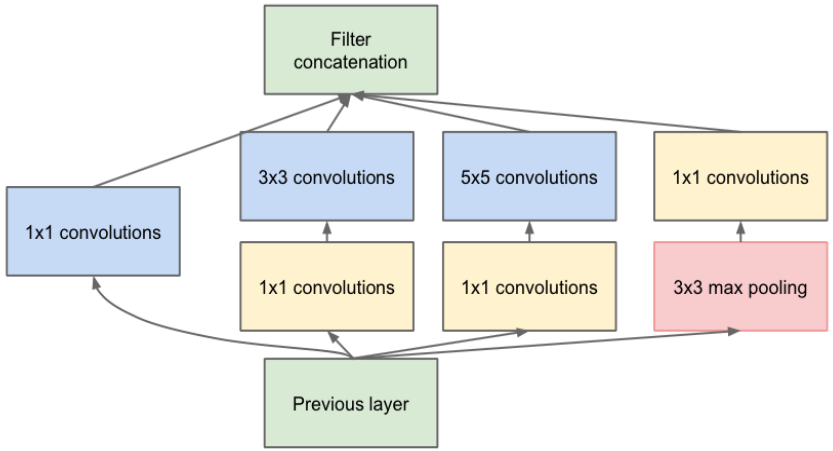



# GoogLeNet

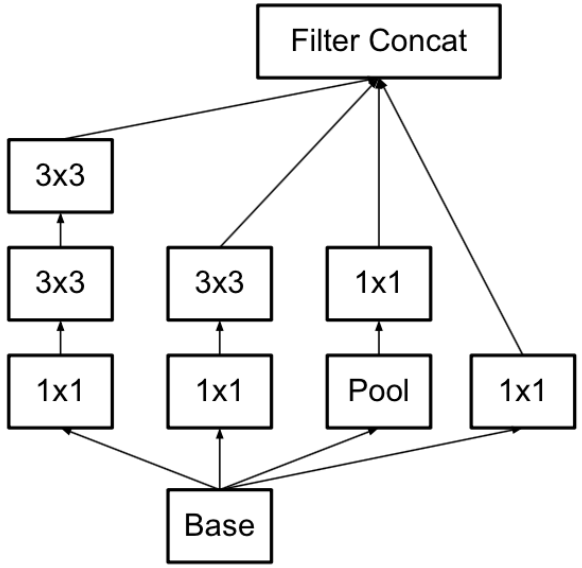
(wide, conv conection ,1X1 conv, simple,5M parameters)

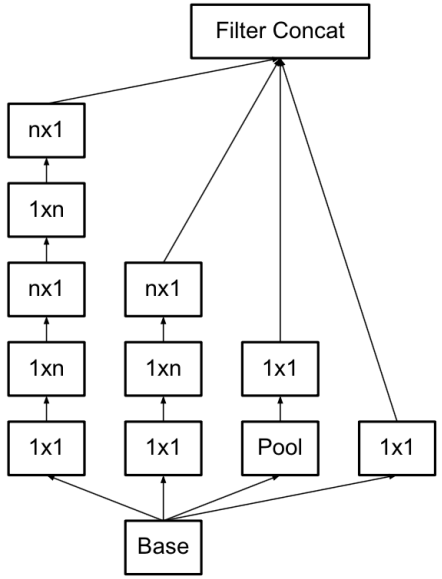


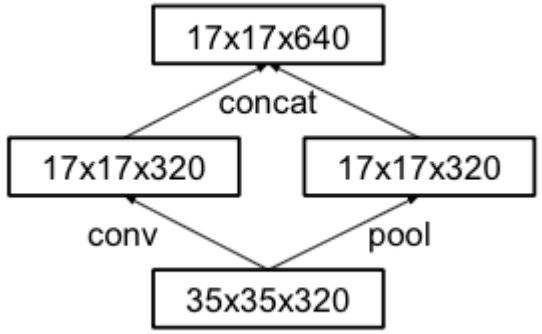




# Inception V2,V3

**(Simplify the conv,decomposition)**

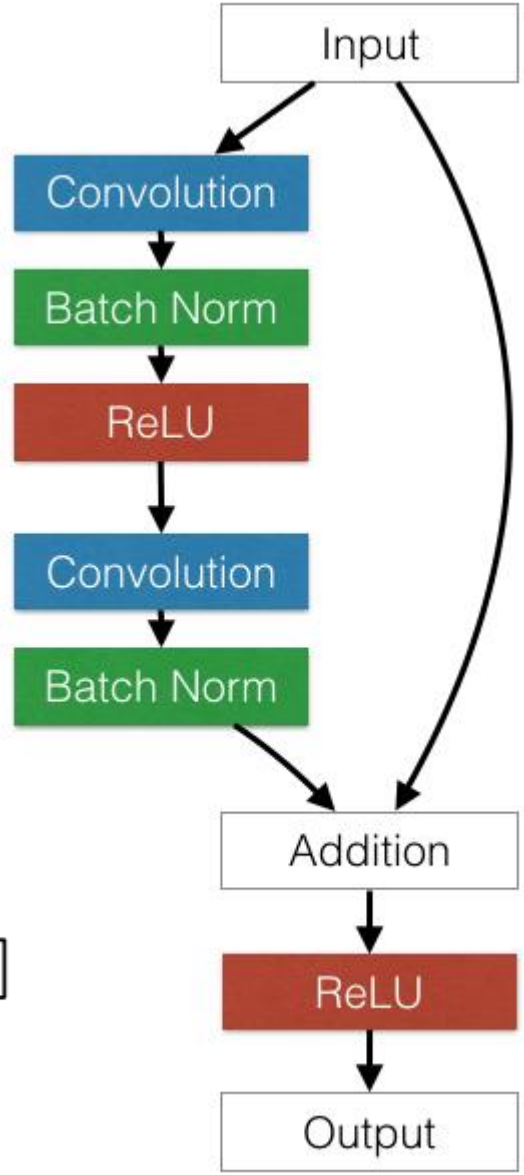


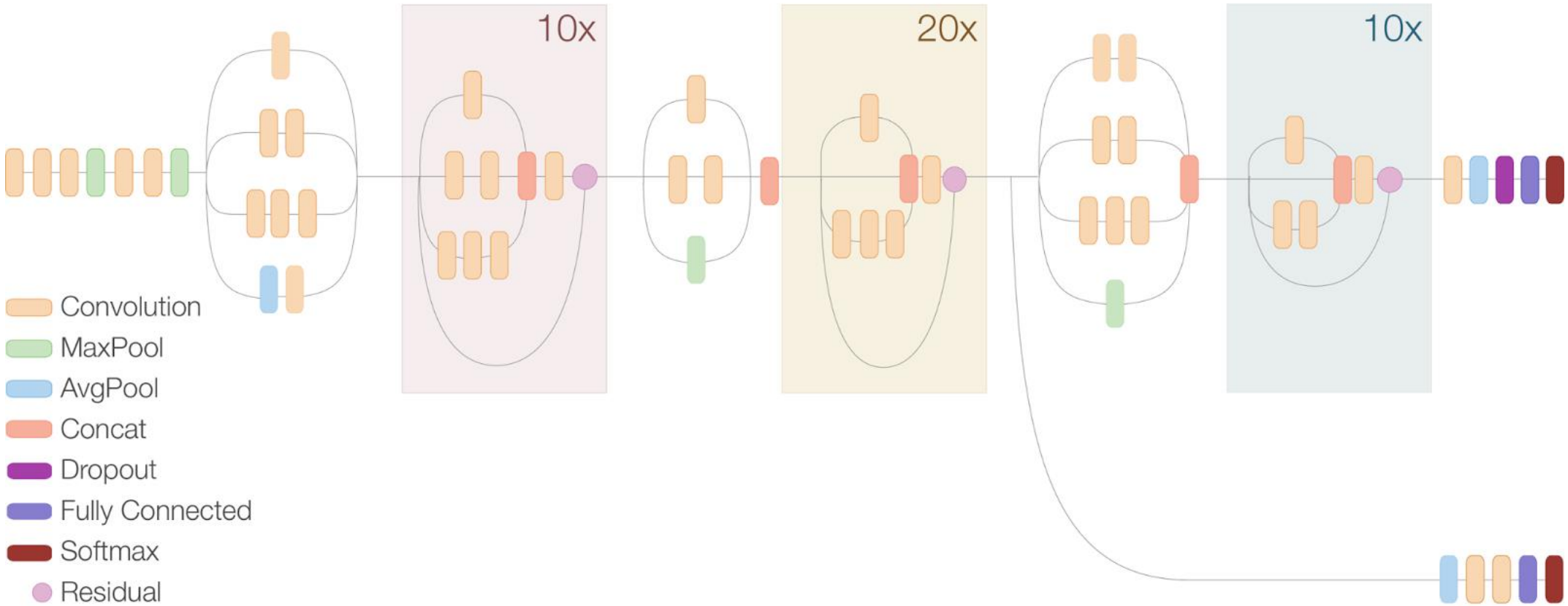

# Rsnet

**(Connection span the layers)**



# Inception Resnet V2 Network

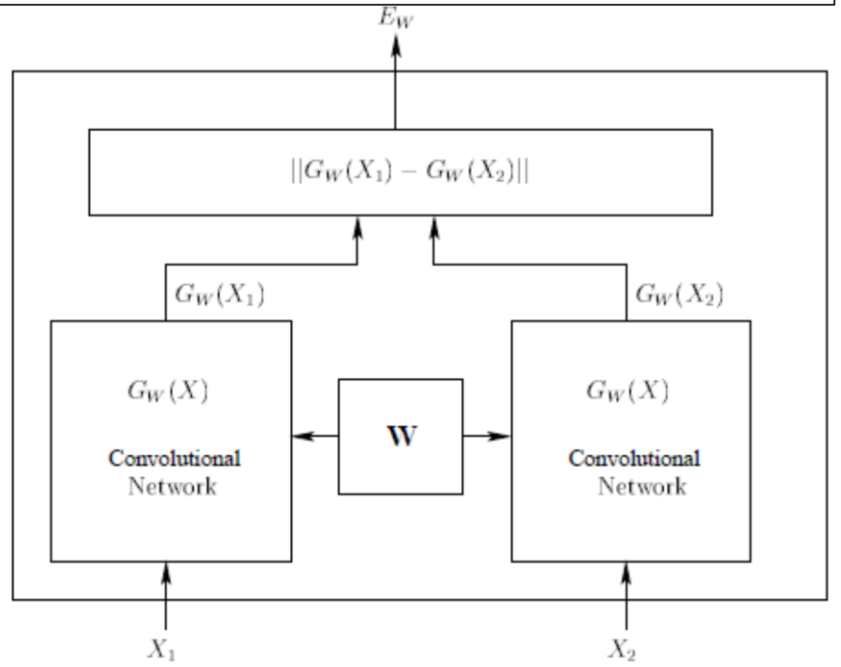
# Compressed View



**Deepface(facebook,2014):- 7 layers CNN /** Siamese network; 97.35% LFW， 0.29M images, 4M images,4000identities (SFC), 120 millions parameters.

- **Detect->3D Align->Represent(CNN)->Verification;**
- **SVM classification (L1 distance)**

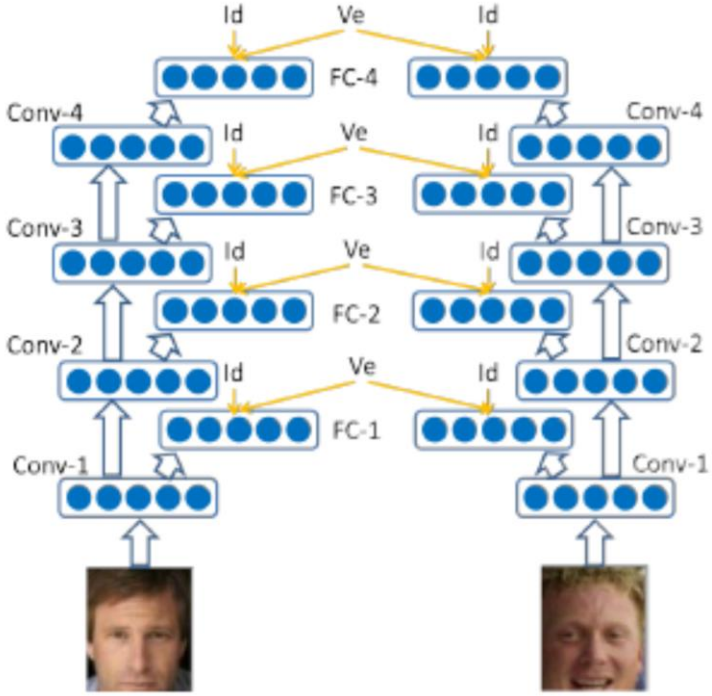
Loss function: single logistic function L(E(w)): 1/0




Siamese network (network(2005))

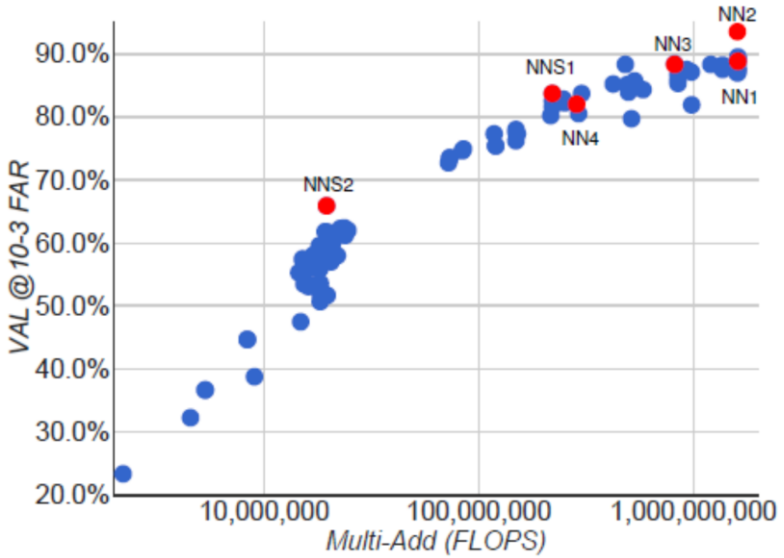
## DeepID/DeepID2/DeepID2+: (CUHK,2014):

- **DeepID:** Siamese networks**,** 7 layers CNN + softmax; 97.45% LFW; multi-scale image patch as input;
- **DeepID2**: Siamese networks DeepID+Verification function+SVM; 99.15% LFW; no 2D/3D alignement
- **DeepID2+**: Embedding space, triple loss, the face images of the same person projecting to a single point, feature dimension 160->512; 99.47% (CelebFaces+WDRef); no 2D/3D alignment.



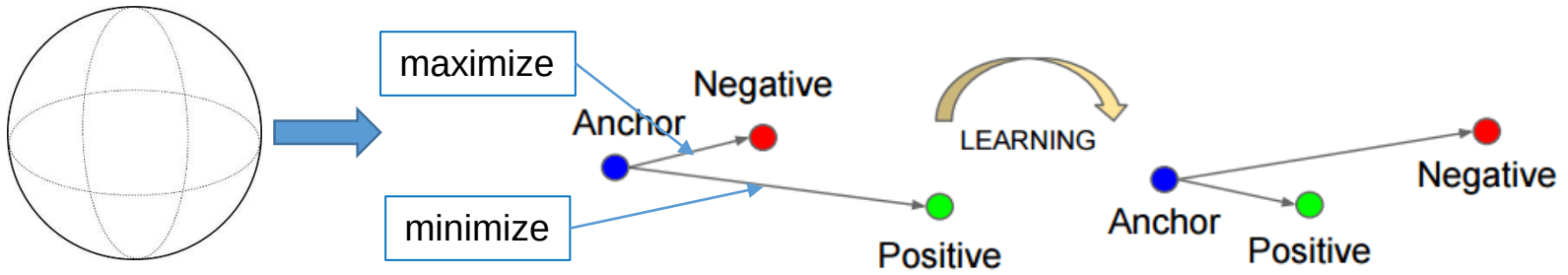
•**FaceNet(Google,2015)**: 22 layers GoogLeNet, 99.63% LFW (human-level verification ~97.5%), 95,12% on Youtube face; 1/20 parameters of AlexNet, 100-200M images (private dataset), 8M identities.

- Balance between accuracy and computation: Accuracy-FLOPS, NN2 can be run on mobile device;




- No 2D/3D alignement;

- Triple loss (Embedding learning): Likely Nearest-neighbor classification, in the hypersphere (||f(x)||=1), mixmizing the distance between different identity and minimizing the distance between the same identity ) :






$D(x_i, x_j) < d, d\ threshold$

Two Same persons

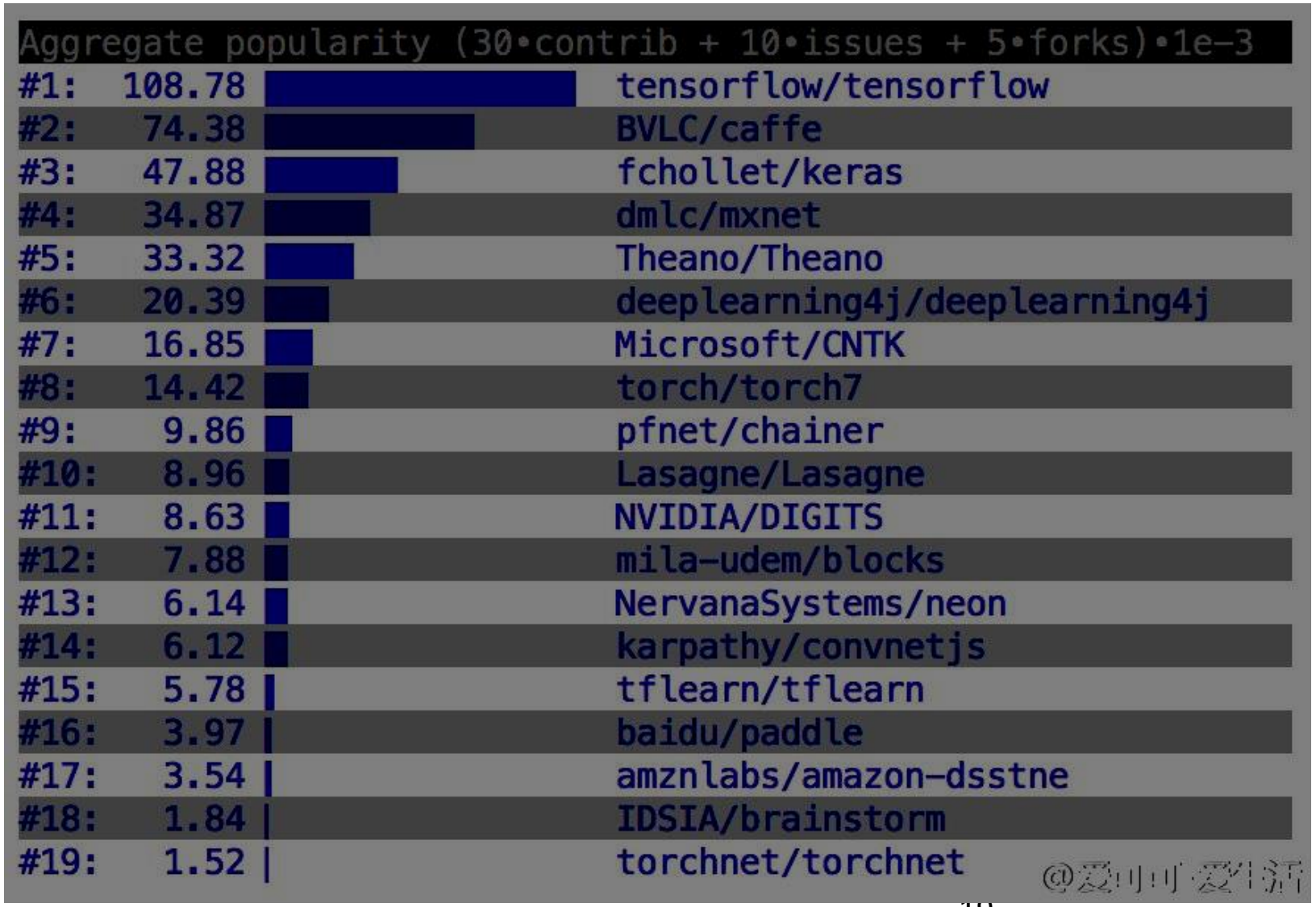
# Dataset

| Dataset | Subjects | Images | Availability |
|---|---|---|---|
| LFW (2007) | 5,749 | 13,233 | Public |
| WDRef | 2,995 | 99,773 | Public (feature only) |
| CelebFaces | 10,177 | 202,599 | Private |
| SFC (Facebook) | 4,030 | 4,400,000 | Private |
| CACD | 2,000 | 163,466 | Public (particial annotated) |
| CASIA-Webfaces (2014) | 10,575 | 494,414 | Public |
| MegaFace (2016) (U. Washington) | 690,000 | 1,027,060 | Public |
| Youtube face (2011) | 1595 | 3425 video 181,3 frames/clip | Public |
| FaceSrub | 530 | 106,863 | Public |
| *ImageNet (ILSVRC 2016) | 100,000 classes; (1000/200/30/365) | 1000 images/class; (50,000/8M…) | Public(category classification) |
| *CIFAR-10, CIFAR-100 | 10 classes<br>100 classes<br>(20 super / coarse classes) | 60,000 (32x32)<br>-10: (5000 training + 1000 test/ class)<br>-100: (500 training + 100 test/ class) | Public(category classification) |

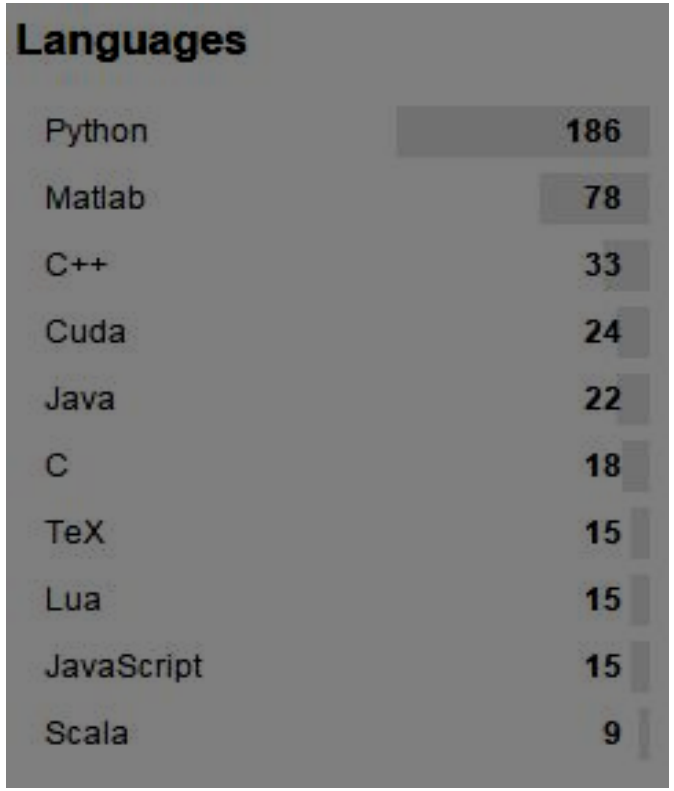
# Dataset

| Dataset | Subjects | Images | Availability |
|---|---|---|---|
| MegaFace<br>(MTCNN Extracted face images) | 485, 045 | 3,262,753 | public |
| Identities_0 | 33,756 | 230,051 | 160x160 pixel |
| Identities_1 | 50,000 | 340,006 | |
| Identities_2 | 29,629 | 195,378 | |
| Identities_3 | 23,209 | 156,314 | |
| Identities_4 | 50,000 | 336,328 | |
| Identities_5 | 34,430 | 231,471 | |
| Identities_6 | 30,228 | 201,868 | |
| Identities_7 | 50,000 | 334,328 | |
| Identities_8 | 26,441 | 177,722 | |
| Identities_9 | 11,943 | 81,845 | |
| Identities_10 | 23,352 | 158,201 | |
| Identities_11 | 50,000 | 331,231 | |
| Identities_12 | 50,000 | 339,164 | |
| Identities_13 | 22,057 | 148,846 | |

# The state of the deep learning frameworks landscape, September 2016



# Deep Learning Libraries by Language




| Python: |
|---|
| Tensorflow |
| Caffe |
| Theano |
| Keras |
| Pylearn2 |
| Lasagne |
| Blocks |
| deepnet |
| CXXNET |
| DeepPy |
| DeepLearning |
| Hebel |
| nolearn |

| Matlab: |
|---|
| ConvNet |
| DeepLearn ToolBox |
| cuda-convnet |
| MatConvNet |

| C: |
|---|
| Intel® Deep Learning Framework |
| NVIDIA DIGITS |
| eblearn |
| SINGA |

| Java: |
|---|
| N-Dimensional Arrays for Java |
| Deeplearning4j |
| Encog |
| MatConvNet |

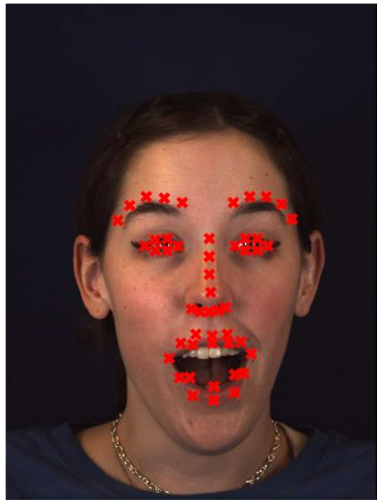
| Lua: |
|---|
| Torch (facebook) |

*** Inception V3: Python – Caffe**

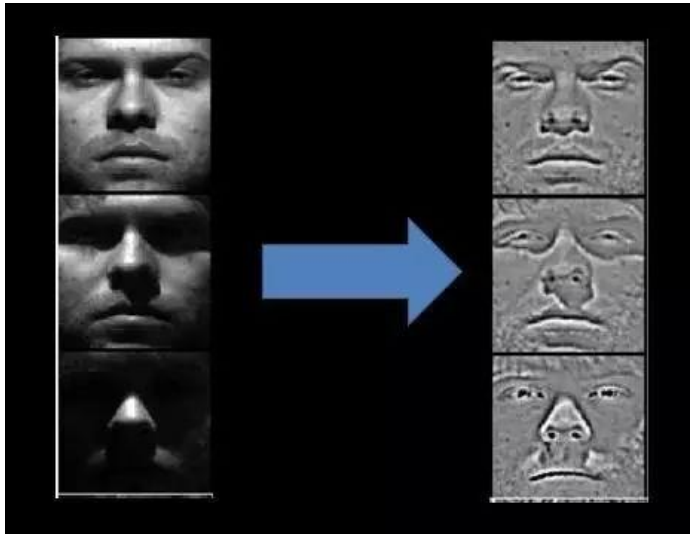
*** FaceNet: Python – Tensorflow (OpenFace)**
**Lua – Torch (openFace)**

# *How to do ?*

## *2. Liveness control (Anti-fraud)*



*Facial expression detection*



*Various Spectrum Image*

*Multi Bio-singnal detection (voice, heart beat,….)*