

The practice of AI

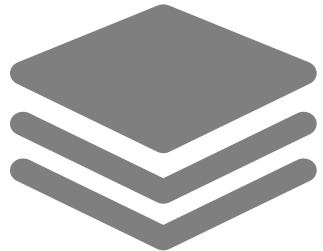
C2: Deep learning & computer vision

Jim Xie

2020/7/6

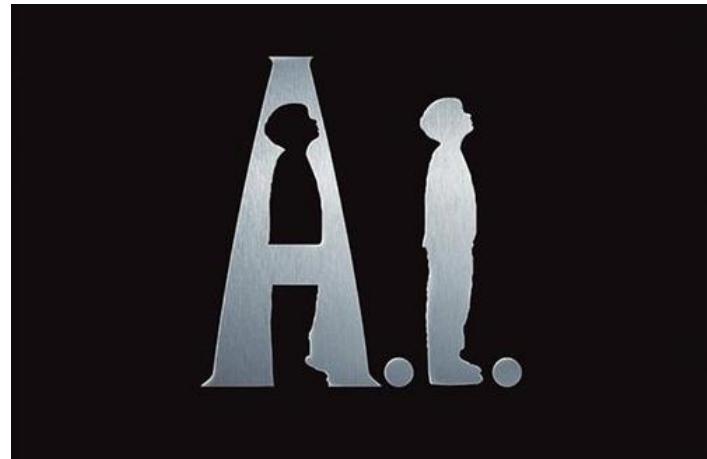


Outline



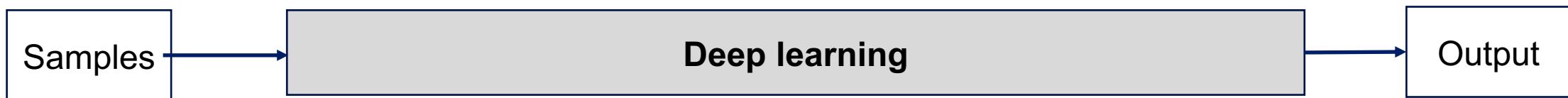
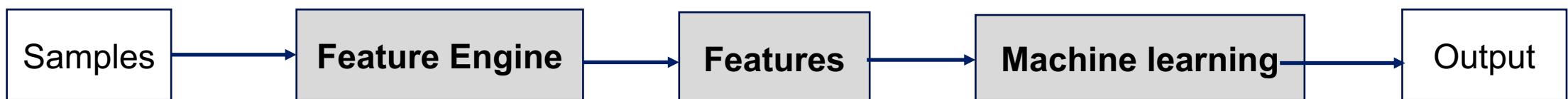
1. Goal
2. Deep learning overview
3. Demo for CNN model train
4. Demo from model Inference
5. Brief summary

Goal

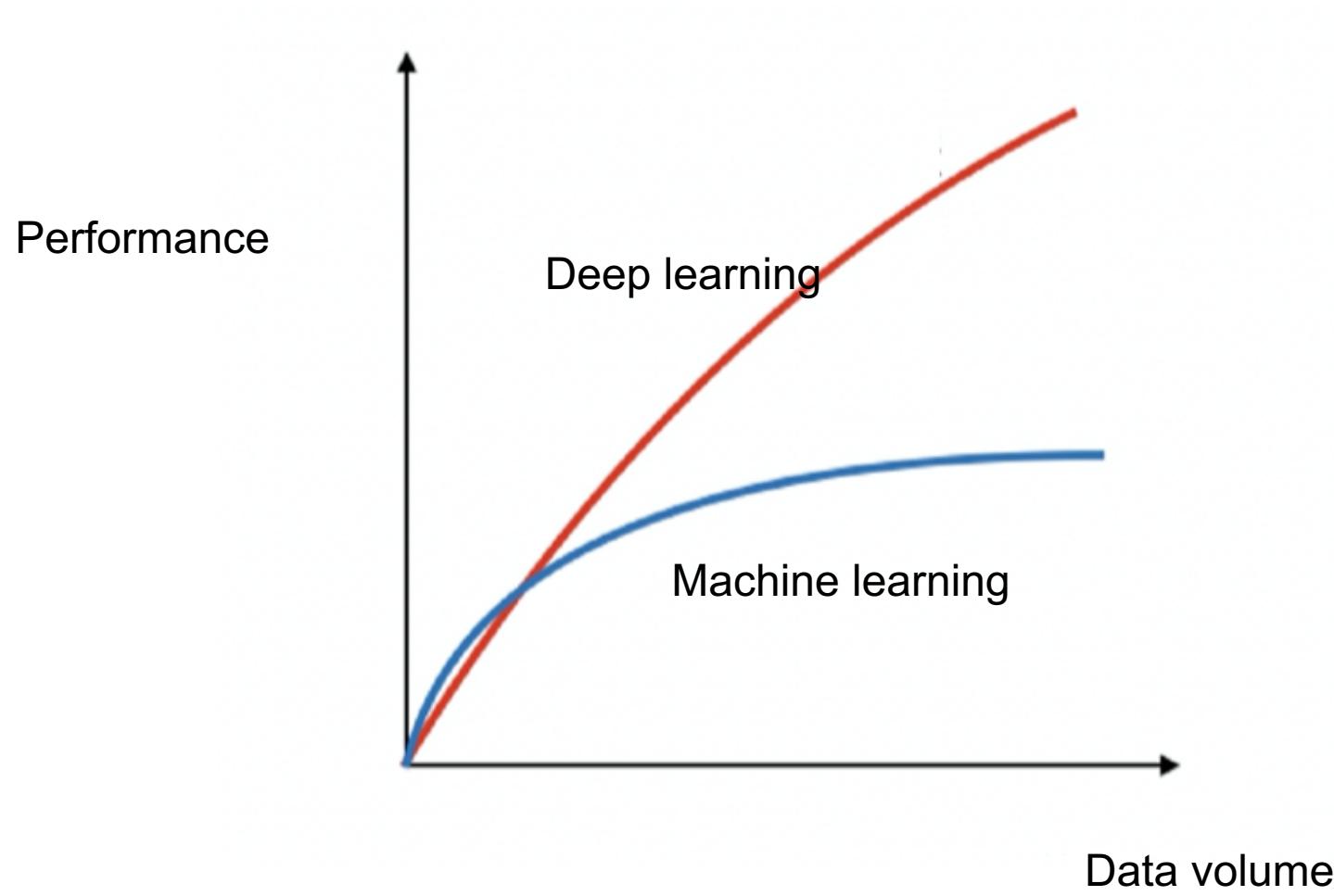


Learn how to train and use a DL model

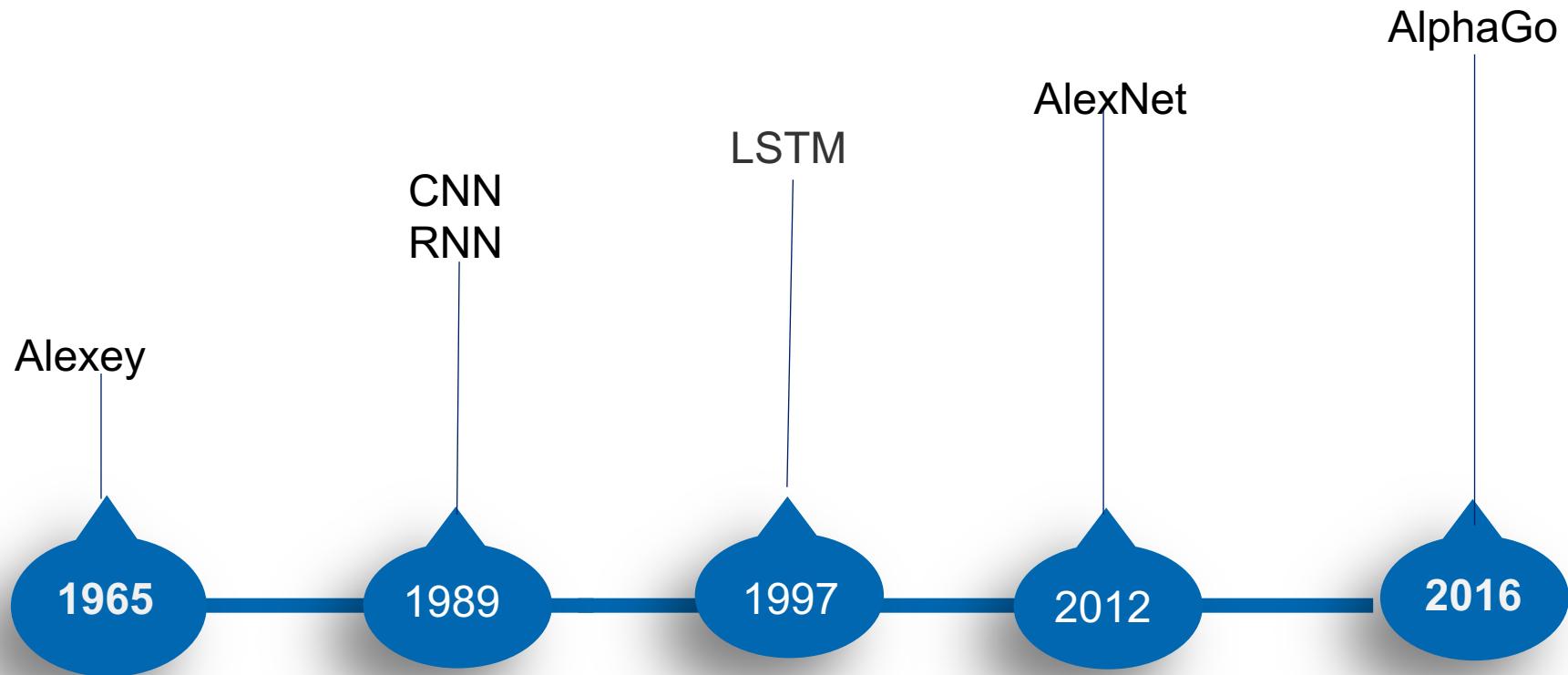
ML vs DL #1



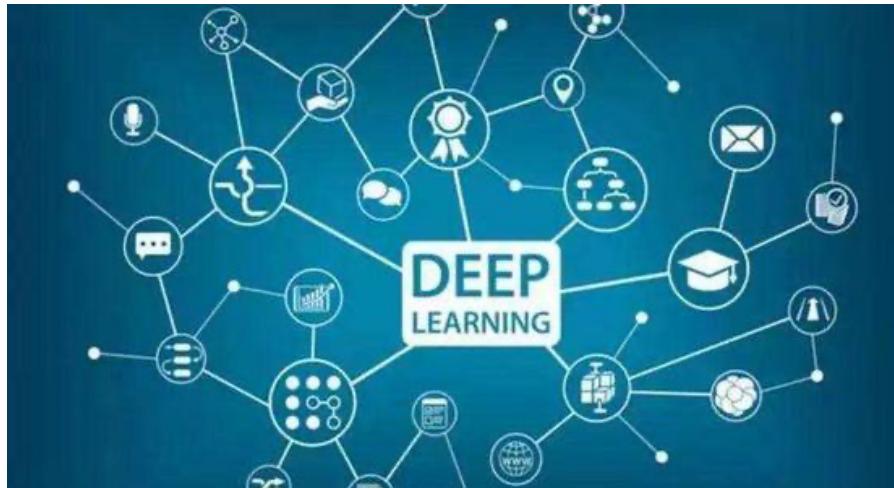
ML vs DL #2



Time machine



Why now

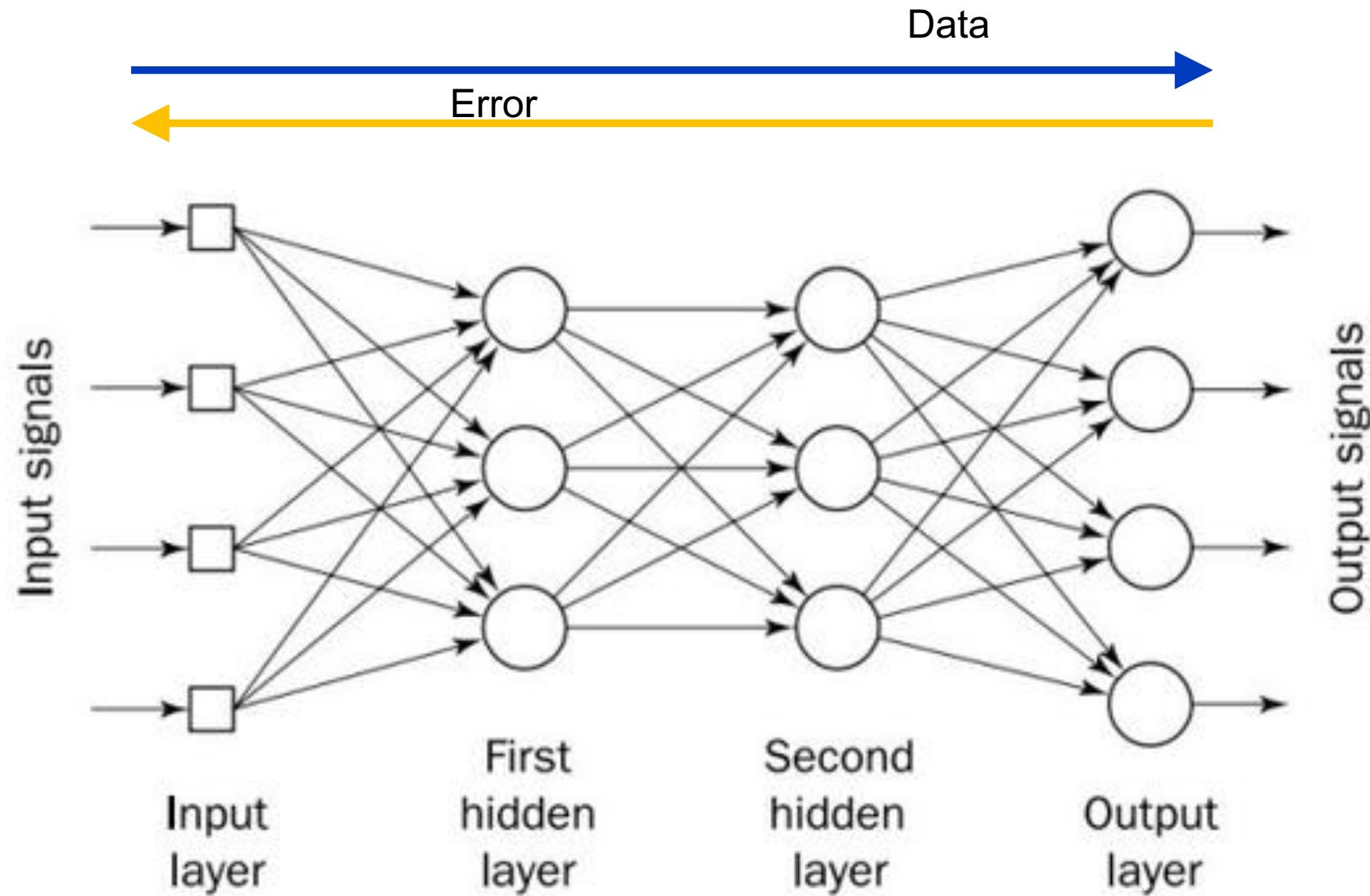


- Prevalent data
- High hardware performance
- More software framework

How

Methodologies

Neural networks



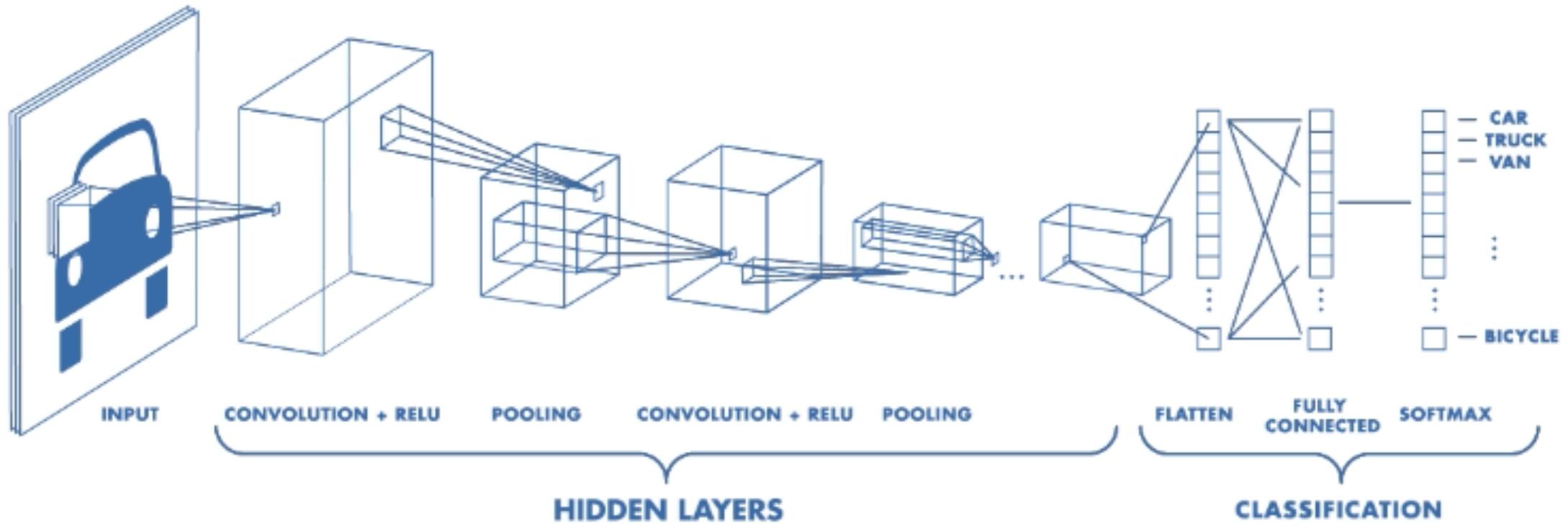
Models

Neural network

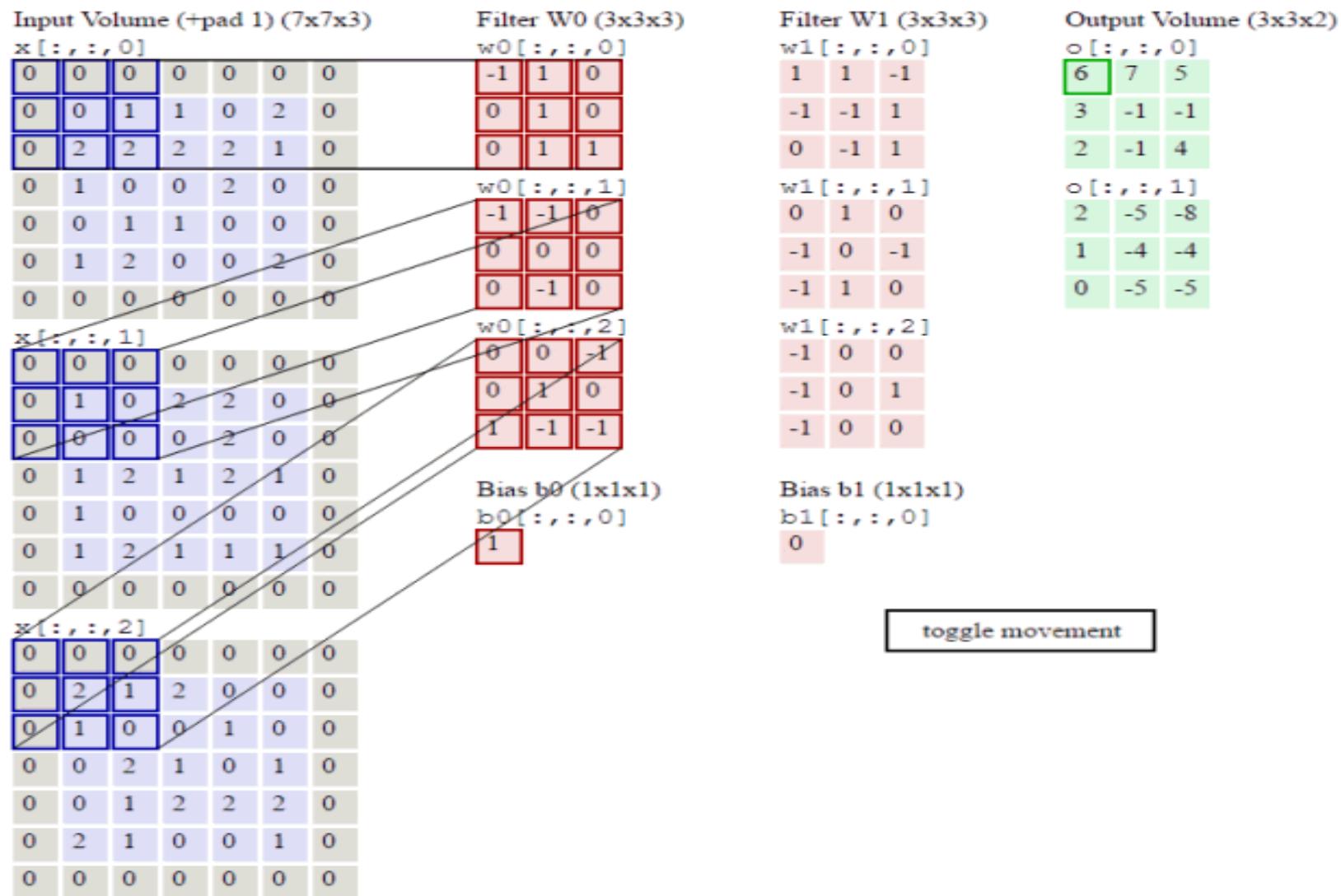


- LeNet
- AlexNet
- VGG/GoogleLeNet
- ResNet
- InceptionNet
- DenseNet
-

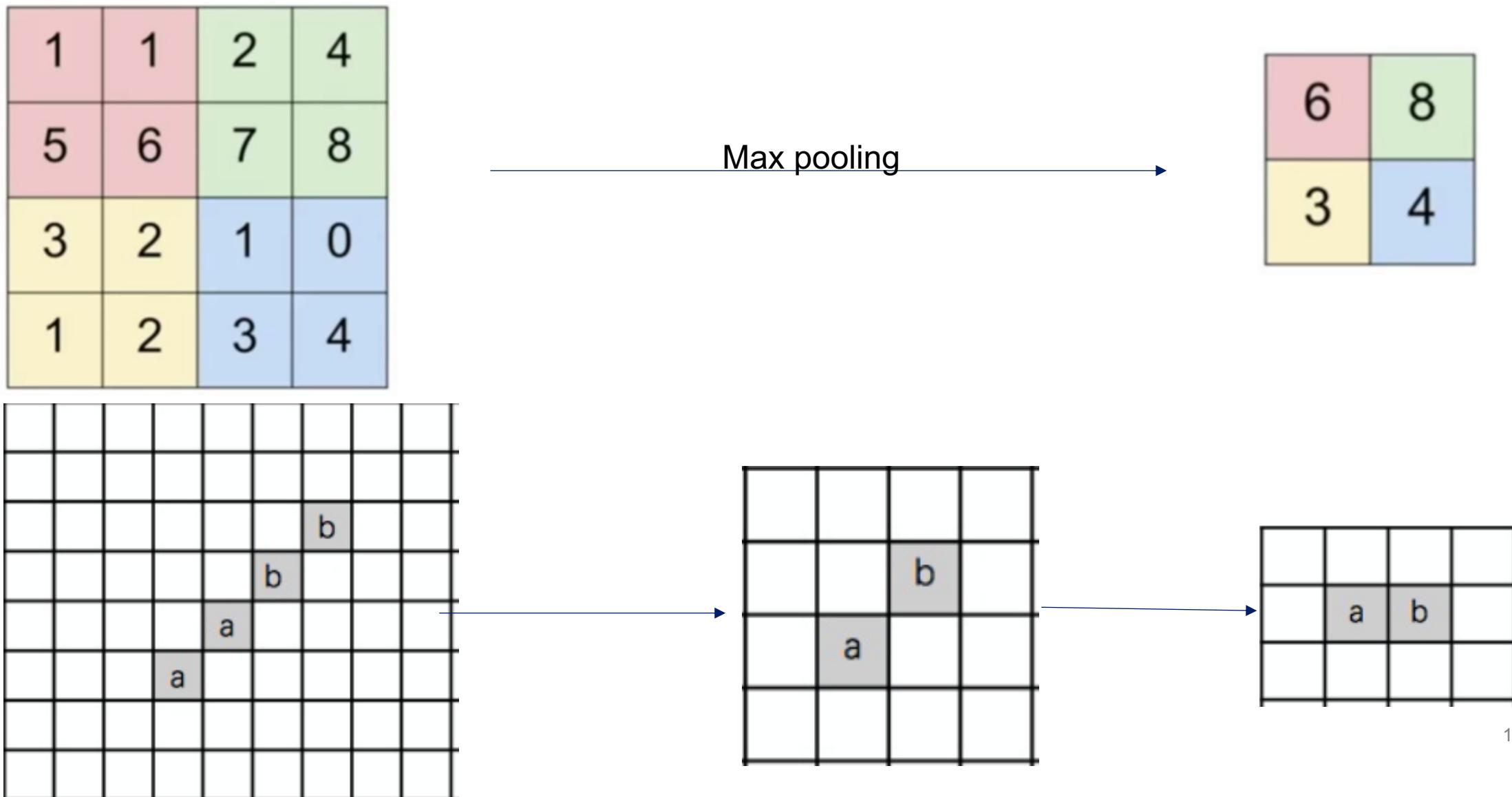
CNN architecture



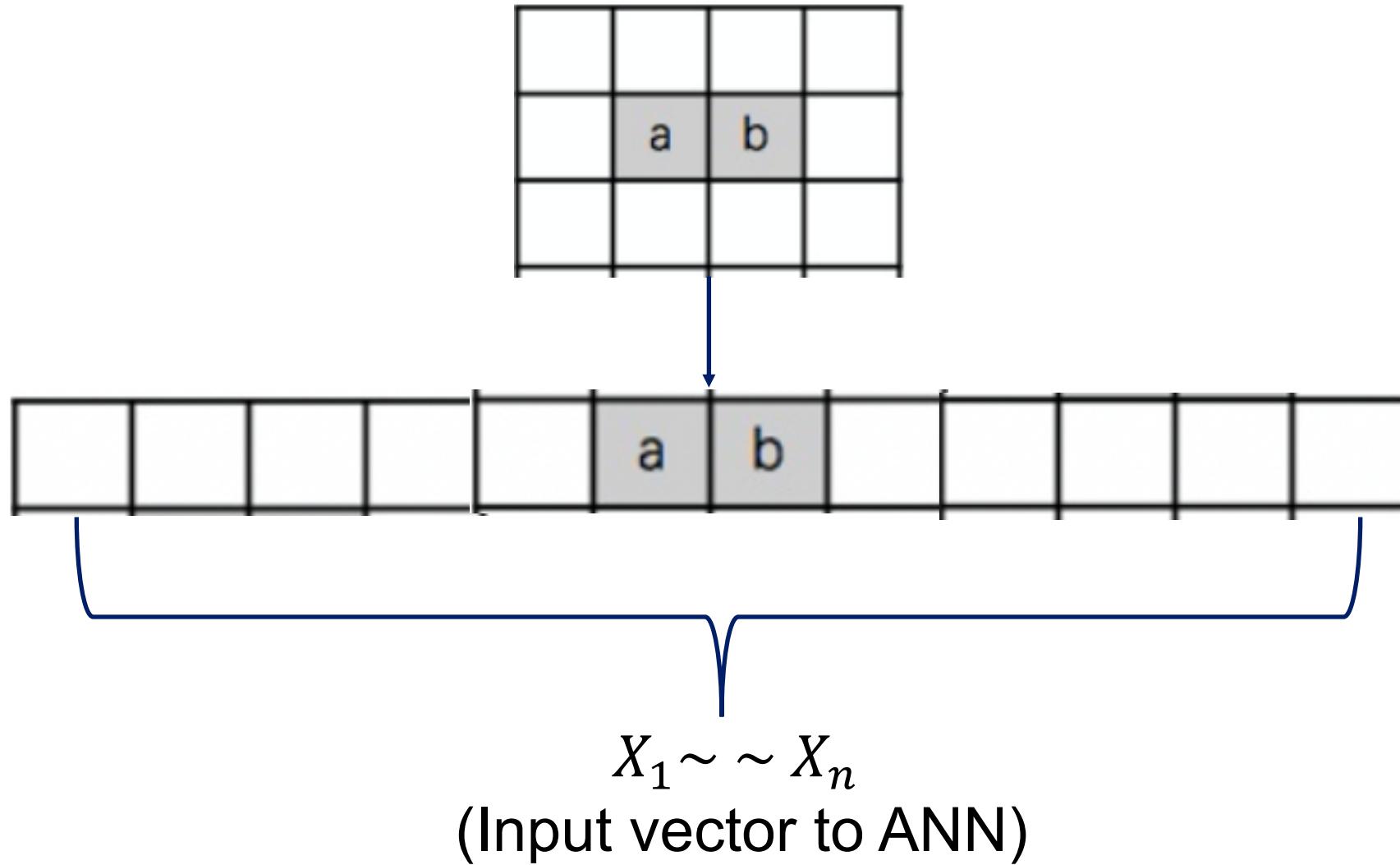
Feature extraction of CNN



Pooling layer



Flatten/Dense

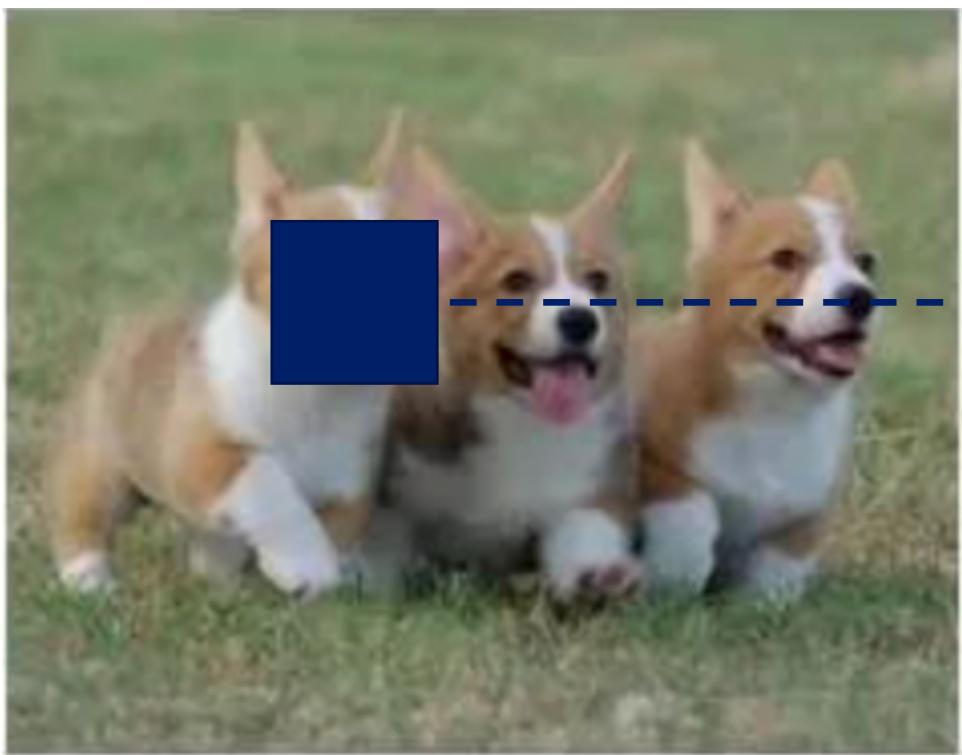


Demo #1

Train a model

CV: Image format

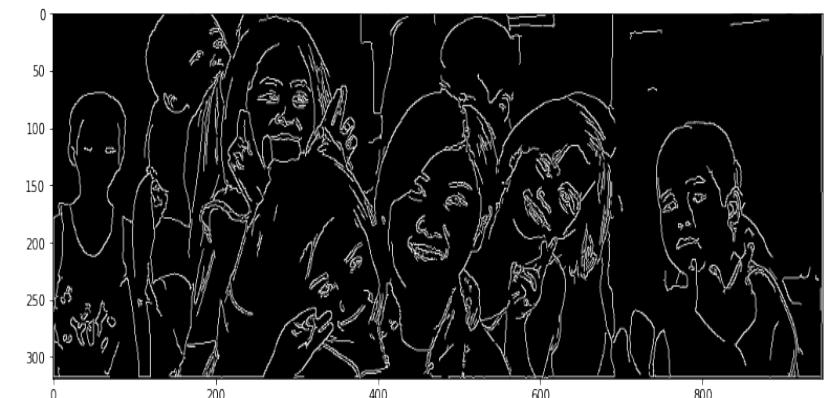
Shape : [height,weight,channel]



08 02 22 97 38 15 00 40 00 75 06 05 07 78 52 12 50 77 91 08
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 34 65
52 70 95 25 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91
22 31 16 71 51 67 43 59 41 92 36 54 22 40 40 28 66 33 13 80
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 64 23 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 39 94 39 43 08 40 91 66 49 94 21
24 55 56 05 66 73 7 78 78 96 83 14 88 34 89 63 72
21 36 23 09 75 00 5 35 14 00 61 33 97 34 31 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 85 24 00 17 54 24 36 29 85 57
86 56 00 48 35 71 89 07 05 44 46 37 41 60 21 58 51 54 17 55
19 80 81 68 05 94 47 69 28 75 92 13 86 52 17 77 04 89 55 40
04 52 08 53 97 35 99 14 07 97 57 32 16 26 26 79 33 27 98 66
88 36 68 07 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 60 62 76 36
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 57 05 54
01 70 54 71 83 51 54 69 16 92 33 48 48 86 81 16 23 57 05 54

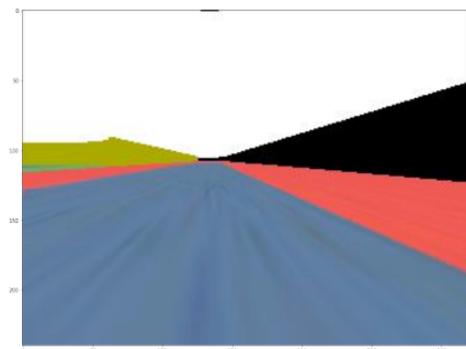
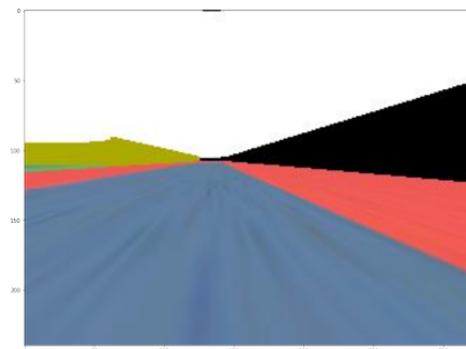
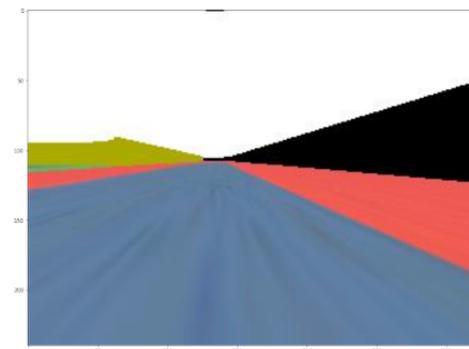
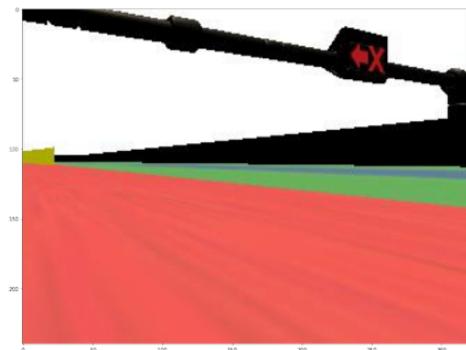
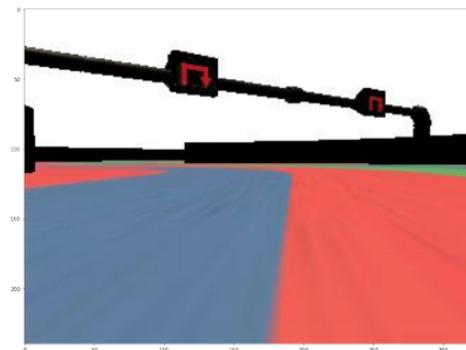
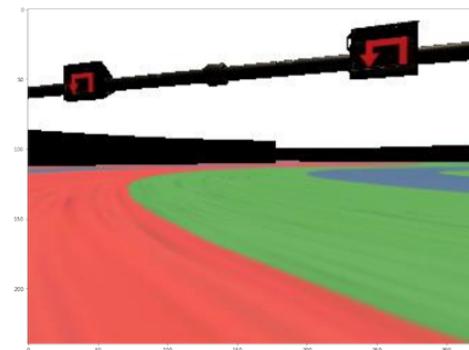
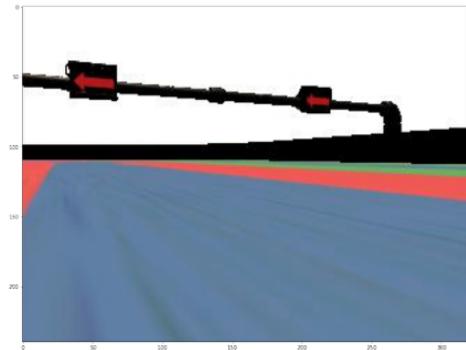
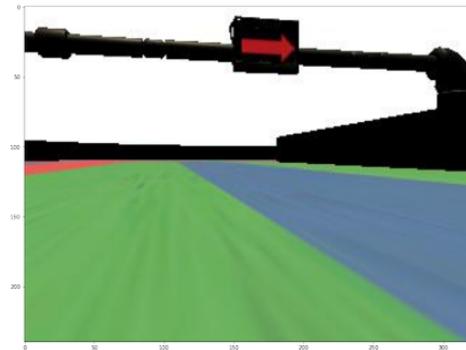
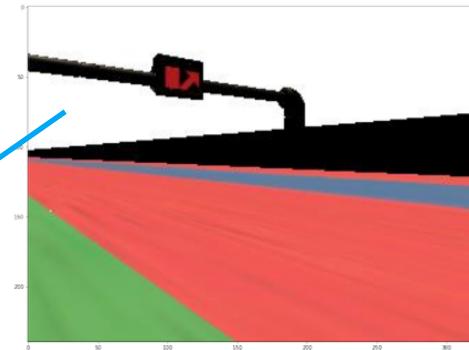
CV: base

- Grayscale
- Blur
- Binarization
- Edge detection

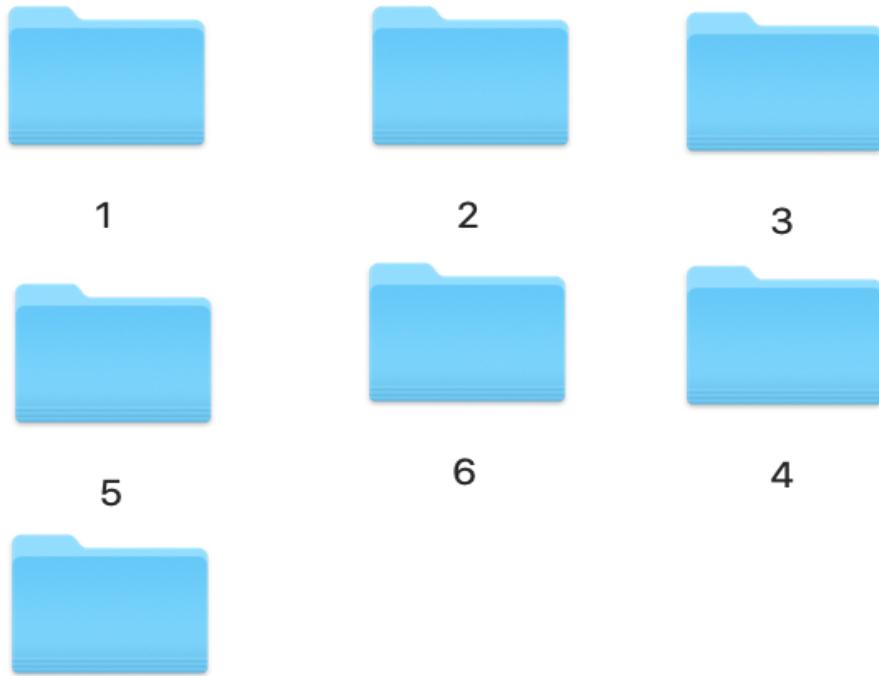
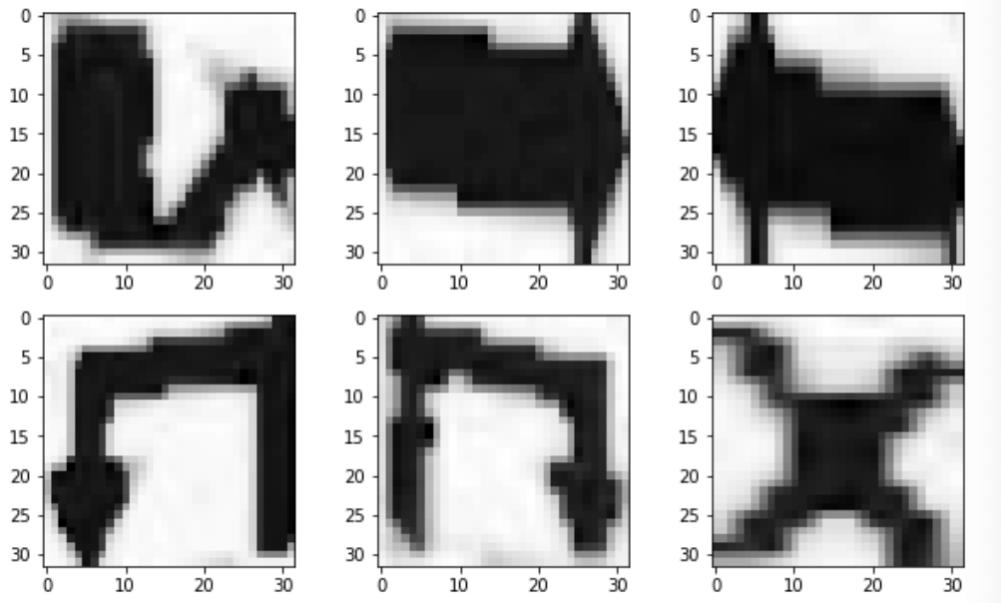


Samples

Signal



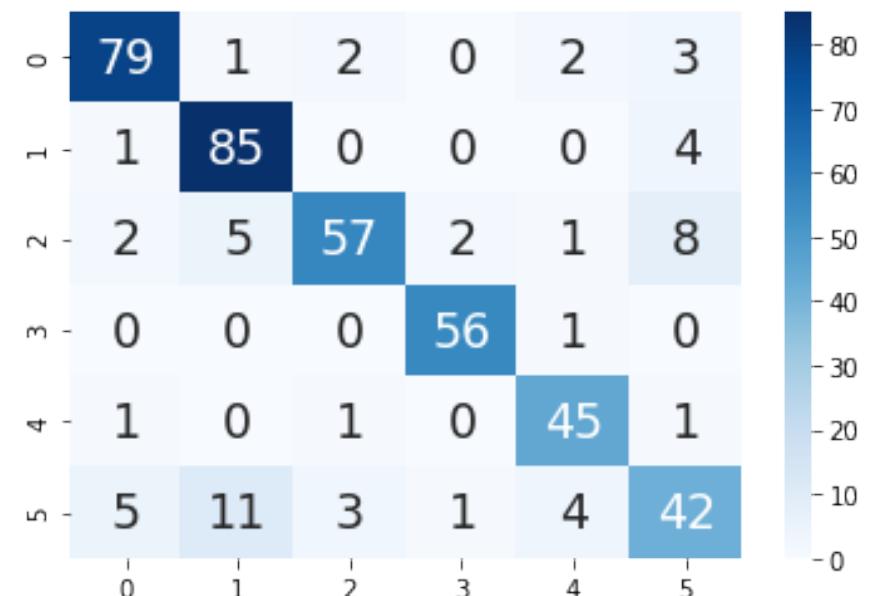
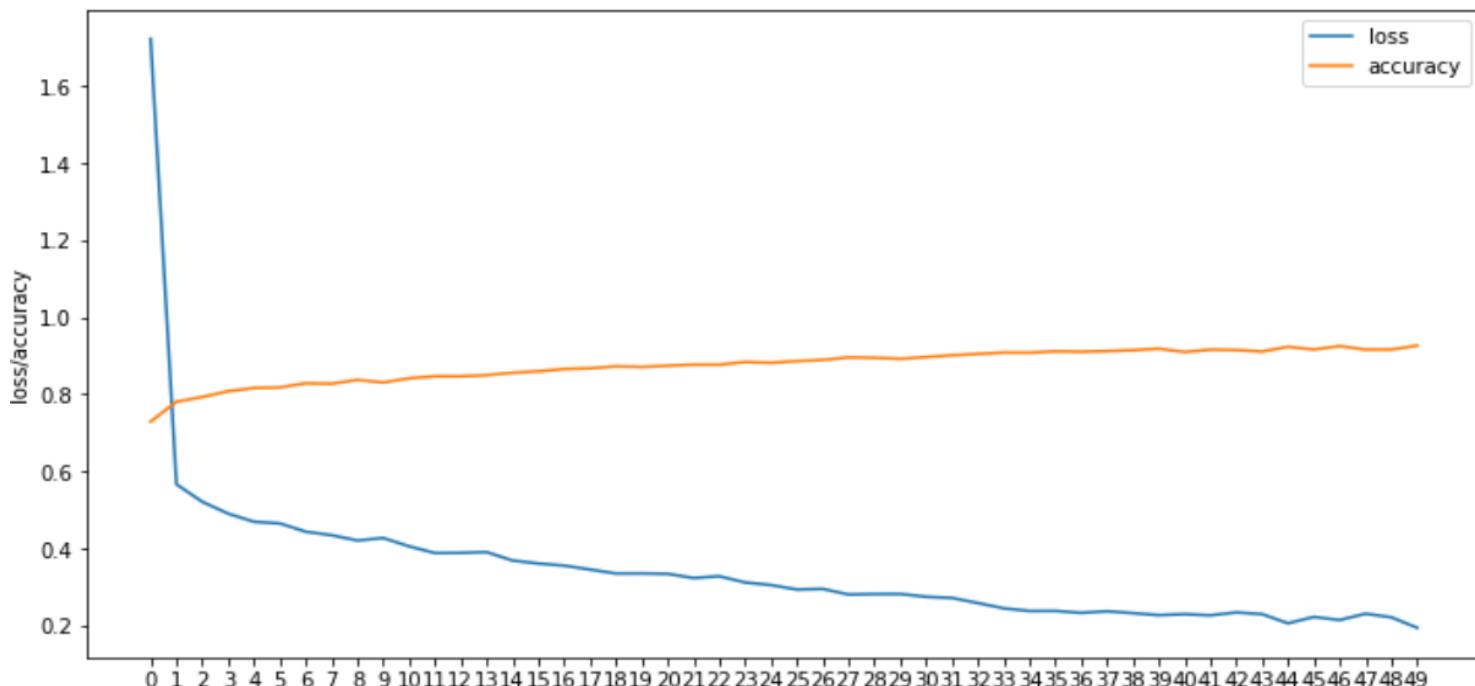
Demo: traffic signal recognition



Train

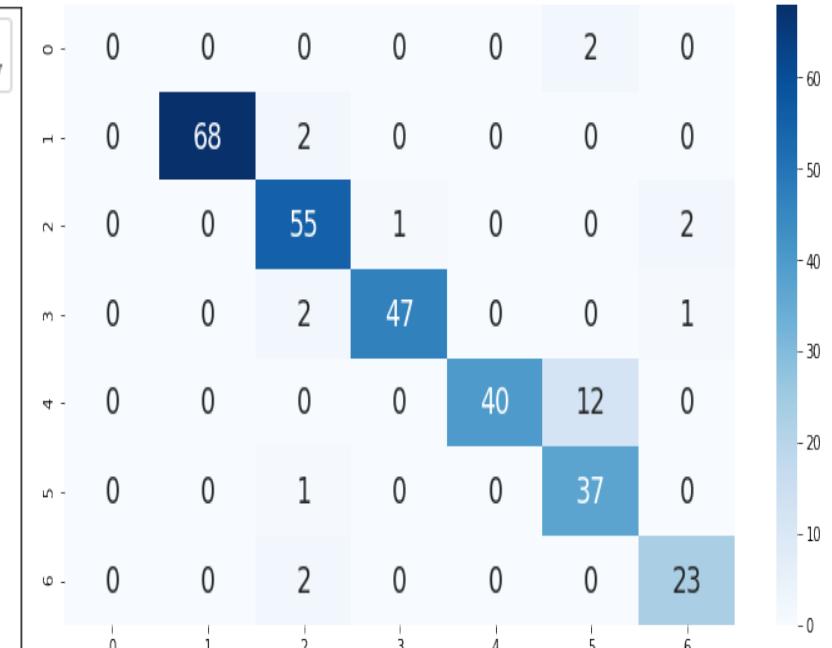
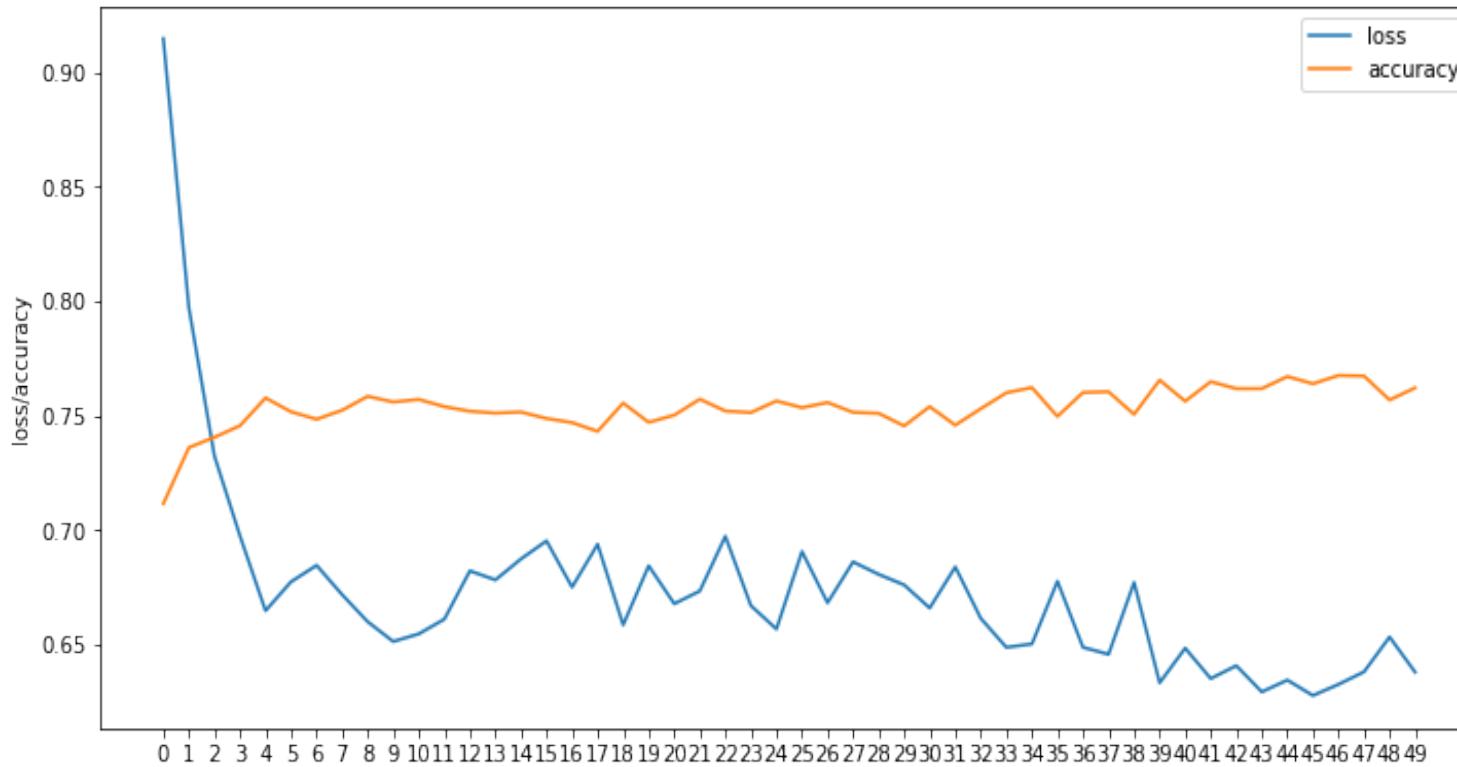
- Create train/test dataset (756)
- Build a CNN model
- Train model
- Performance evaluate
- Visual analysis
- Inference and deployment

Relu



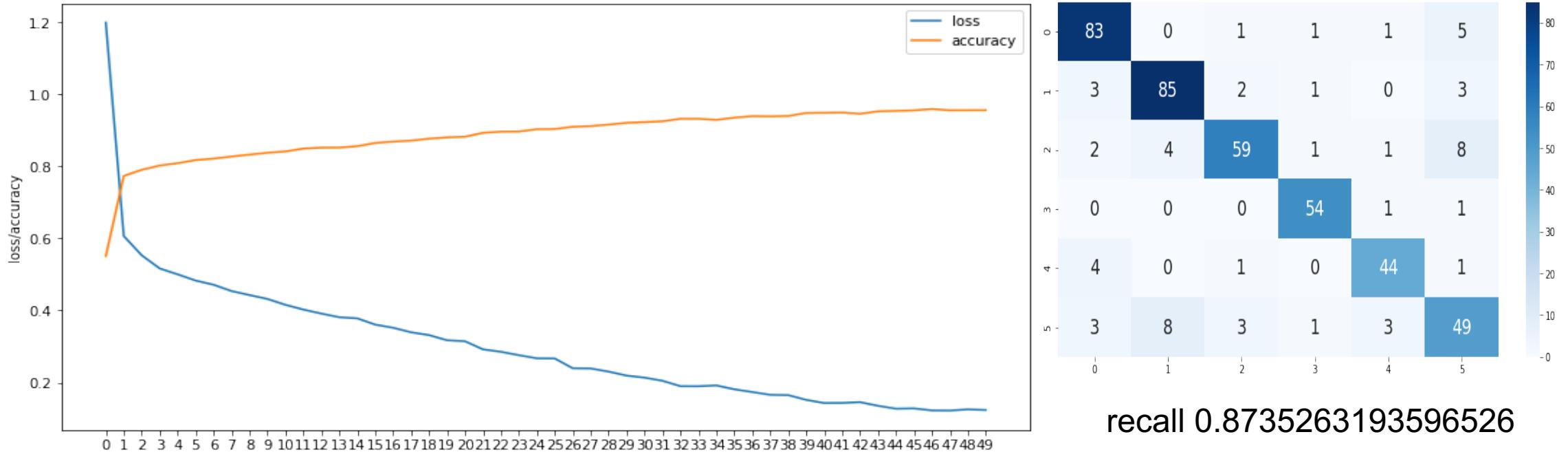
Recall: 0.8614683663617422

Tanh

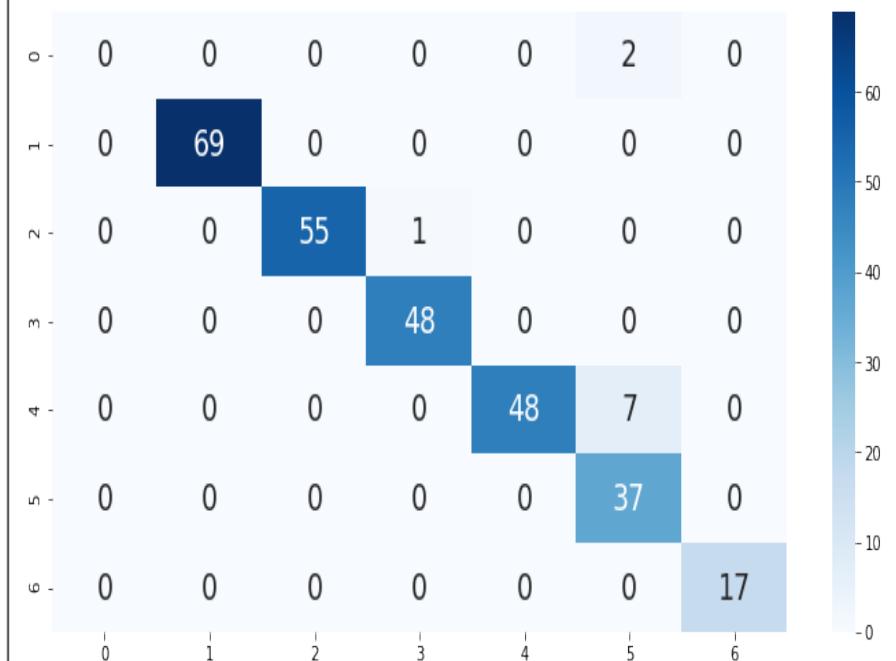
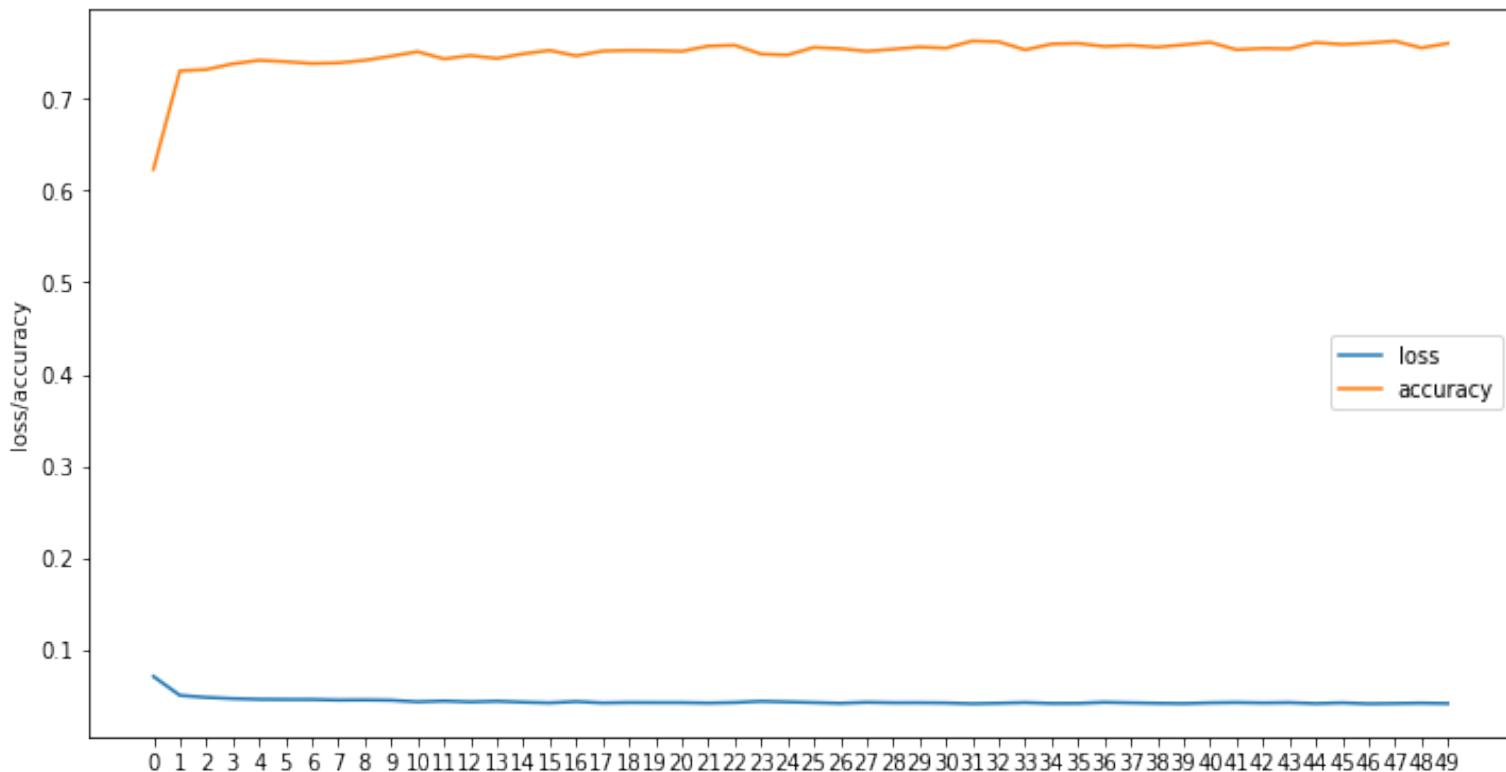


Recall: 0.7889456304649459

Sigmoid



MSE



recall 0.7889456304649459

Learn from Demo #1

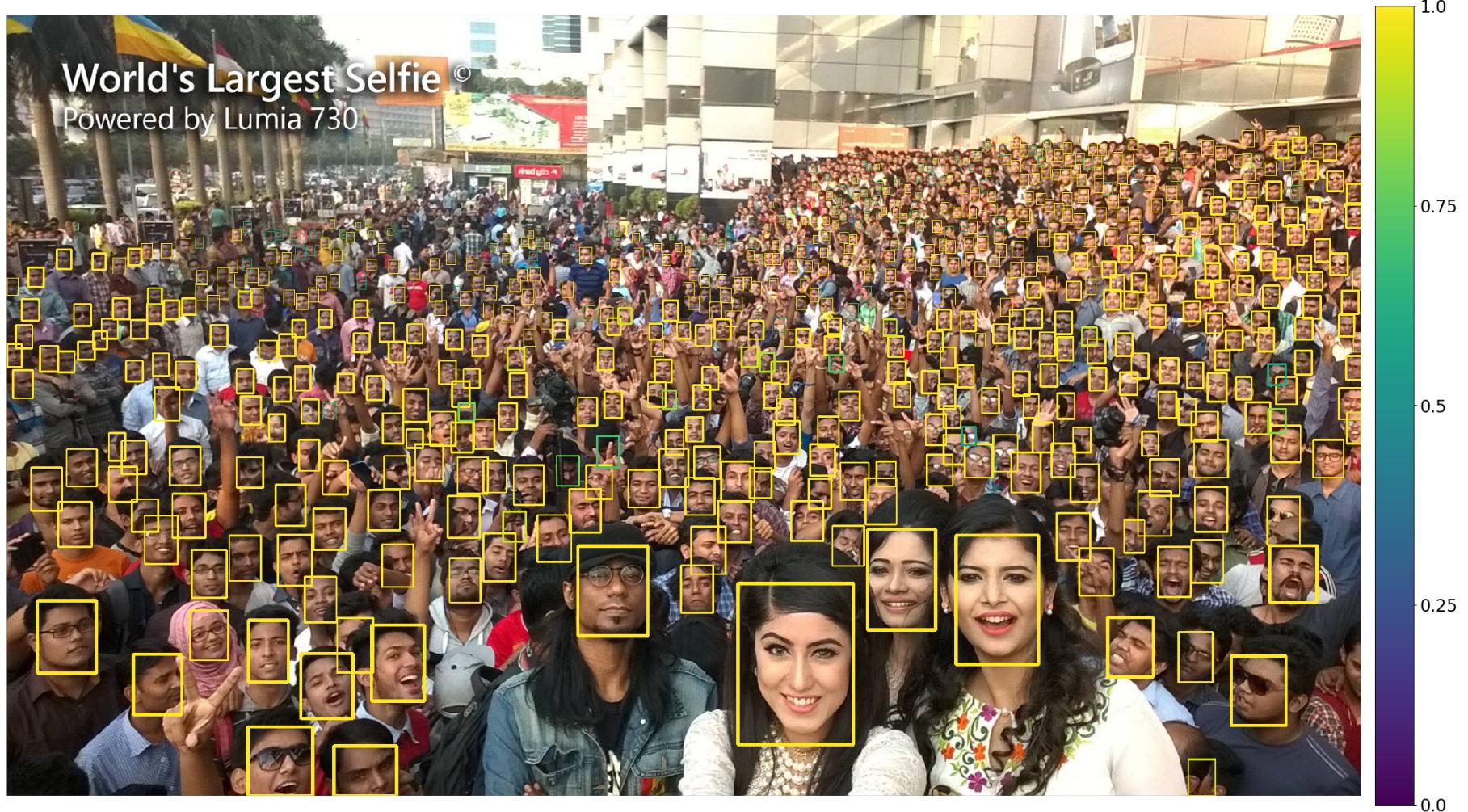
- 
1. Build model
 2. Initialize parameters
 3. Feed data to model
 4. Calculate error by predicted values and real values
 5. Adjust model's parameters base on error
 6. Iterate previous steps until error is accepted
- Sample
 - Effort for sample label
 - Preprocess
 - Train (Hyper parameters selection)
 - CNN layers
 - Learning rate
 - Iteration round
 - Loss/activation functions

DL models and applications

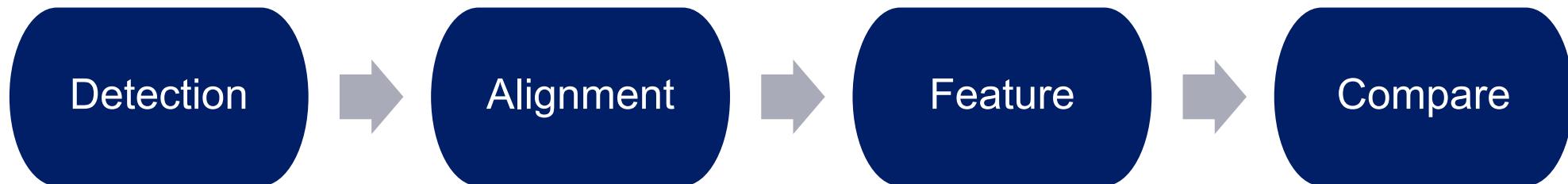
- **Computer vision**
 - CNN
 - R-CNN
 - Fast R-CNN
 - Mask R-CNN
- **NLP/Speech**
 - RNN
 - LSTM
 - GPT-3
 - Transformer

Demo #2

Use a model



Face recognition

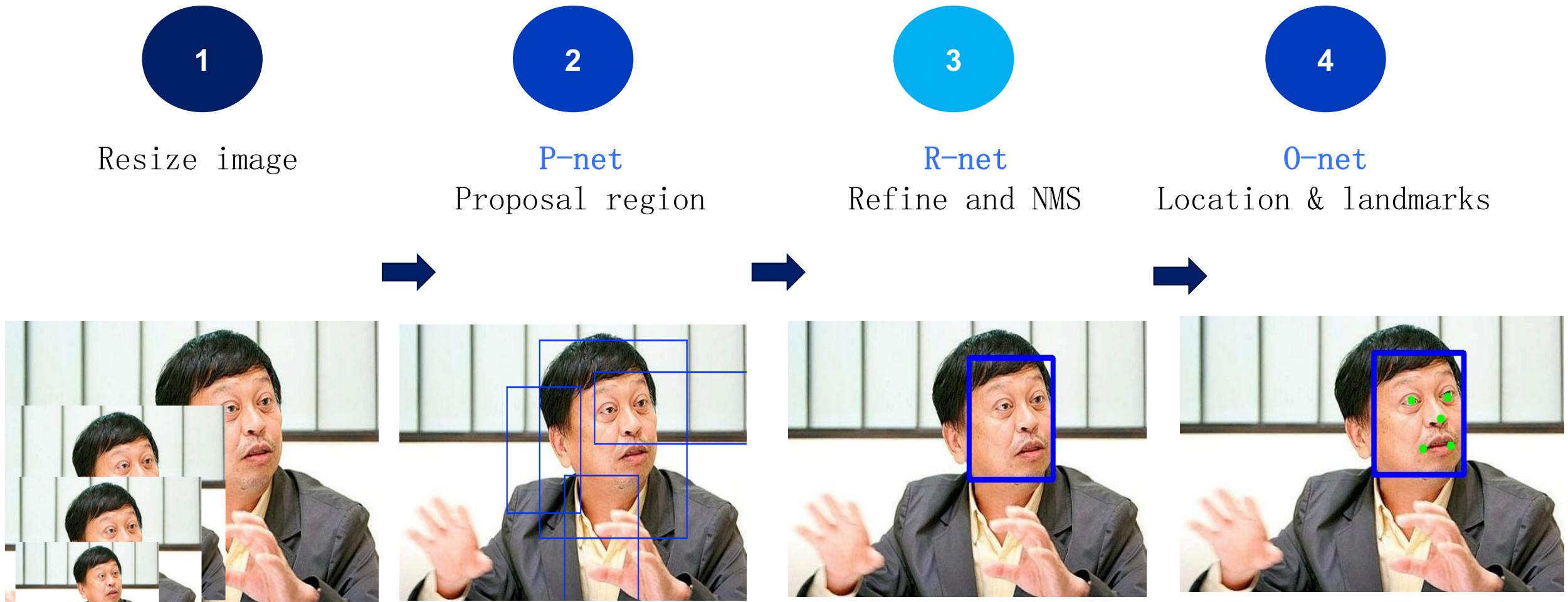


Detection

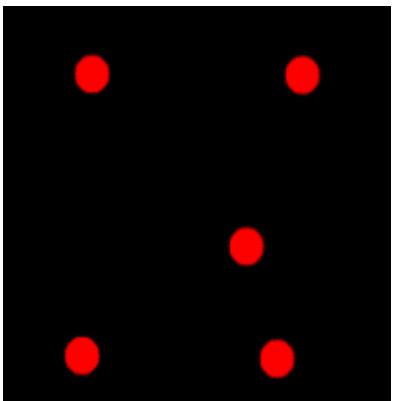


| 方法 | 优点 | 缺点 |
|--------------|---------|----------|
| Haar cascade | 可以检测小脸 | 不适应侧脸，遮挡 |
| HOG SVM | 适应侧脸，遮挡 | 不能检测小脸 |
| MTCNN | 鲁棒性强 | 占资源，检测慢 |
| | | |

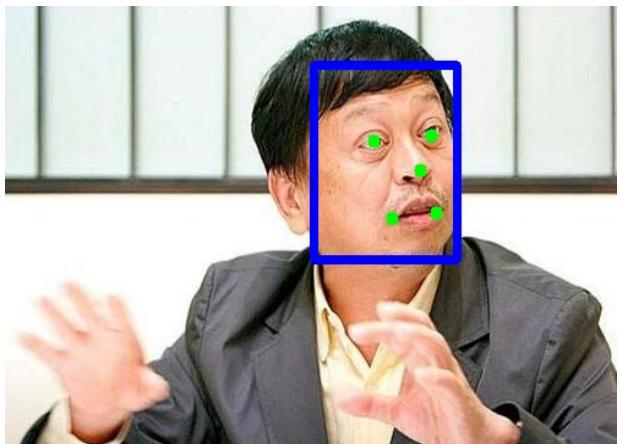
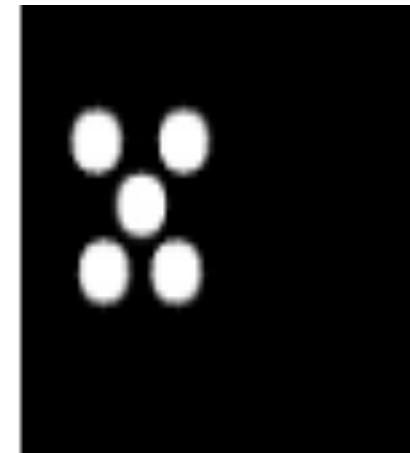
MTCNN



Alignment



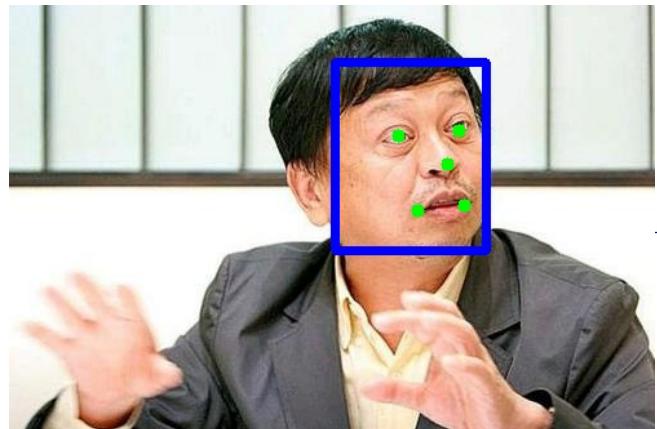
Affine matrix



Affine transformation

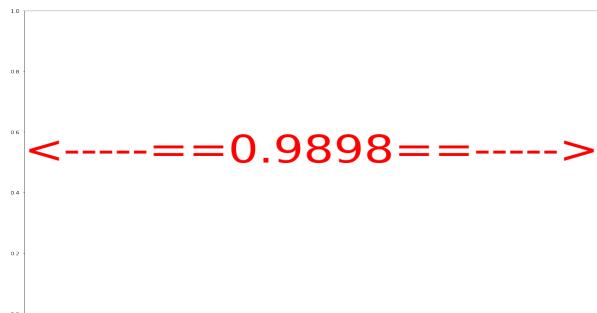
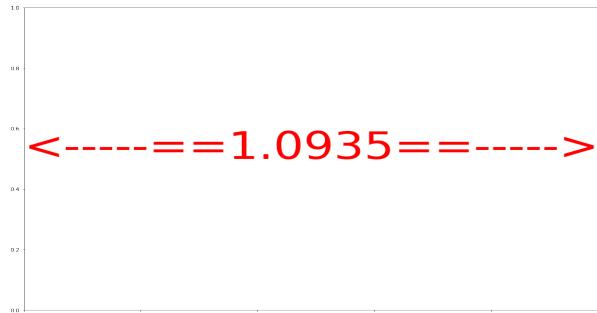
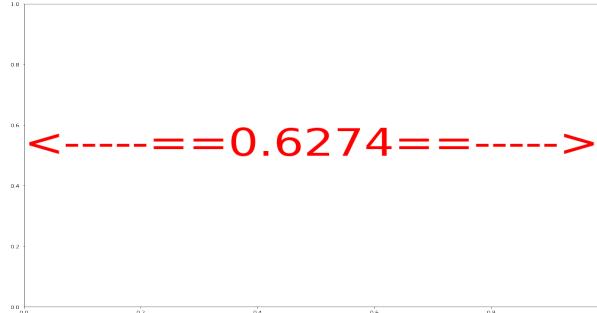


Feature



```
[0.04027741029858589, 0.08351167291402817,  
0.005823252256959677, 0.0723235160112381,  
0.028528105467557907, -0.002782160881906748,  
-0.06524961441755295, 0.10399515926837921,  
0.01219485979527235, 0.03158007934689522,  
0.046927742660045624, -0.02802014909684658,  
-0.03863518312573433, -0.0190310999751091,  
0.005247218534350395, -0.03423447161912918,  
0.06390134245157242, 0.043098170310258865,  
-0.06691527366638184, 0.019953561946749687,  
0.014163636602461338, 0.00026211352087557316,  
-0.010168898850679398, -0.028404725715517998,  
-0.04332667589187622, -0.01461505051702261,  
0.06261993199586868, 0.0735521912574768,  
0.030010586604475975, 0.061586856842041016,  
-0.0013521158834919333, 0.02855655737221241,  
0.020252032205462456, 0.0172031968832016,  
0.004351011477410793, -0.08569888025522232,  
-0.05994438752532005, -0.06681397557258606,
```

Compare



Learned from demo #2

- ❖ NOT easy to use an model
- ❖ Hard to explain
- ❖ Too much consideration
 - Trouble shotting
 - Accuracy and enhancement
 - Performance

Framework

- Theano (2010)
- Caffe[(2013)
- **TensorFlow** (2015)
- Paddle Paddle (2016)
- **PyTorch** (2017)
- Jittor (2020)

Q&A

Thanks