# Introduction to Image Processing, Computer Vision and Deep Learning
# (影像處理、電腦視覺及深度學習概論)

# Computer Vision and Deep Learning
# (電腦視覺與深度學習)

# Example of Final Project Report

# Grading

1. (40%) Project presentation
2. (40%) PPT file + additional demo video
3. (20%) Source code

➔ Need to upload: PPT file + demo video file + source code

---

## 1. Project presentation

1) The project presentation will be held at classroom during the class time.
2) You can check the presentation time of your group on Moodle. Remember to bring your notebook to present and demonstrate your project.

## 2. PPT + demo video

1) PPT file: Please check following example slices. Around 15-20 pages
2) Additional demo video file

## 3. Source code

1) The source code of your final project: You can use any programming languages.

# Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks

*Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, Senior Member, IEEE, and Yu Qiao, Senior Member, IEEE*

IEEE SIGNAL PROCESSING LETTERS, VOL. 23, NO. 10, OCTOBER 2016

K. Zhang, Z. Li, and Y. Qiao are with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: kp.zhang@siat.ac.cn; zhifeng.li@siat.ac.cn; yu.qiao@siat.ac.cn).

Z. Zhang is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: zz013@ie.cuhk.edu.hk).

**Keyword:**

Cascaded framework, Three-stage multi-task deep convolutional networks, Face alignment, Face detection

Group 06: 2023/01/02

| | | |
|---|---|---|
| 資訊系 | 機器人實驗室 | 連AA |
| 資訊所 | 機器人實驗室 | 王BB |
| 電機系 | 太空實驗室 | 張CC |
| 機械所 | 飛車實驗室 | 何DD |

# Content:

# 1.1 Introduction: Motivation and Objective

1. Motivation:
   - Face detection and alignment are essential to many face applications, such as face recognition and facial expression analysis.

2. Objective:
   - Detect face bounding boxes and face landmarks from images.
   - Challenge:
     1) Complex face variations (poses, lightings, occlusions)

# 1.1 Introduction: Global Framework

## 3. Global Framework:

Cascaded framework: Three-stage multi-task deep convolutional networks

1. Proposal network (P-Net) - Fast
   1) Obtain the candidate facial bounding box.
   2) Use non-maximum suppression (NMS) to merge highly overlapped candidates.

2. Refine network (R-Net) - Refine
   1) Reject a large number of false candidates
   2) Use NMS to merge highly overlapped candidates.

3. Output network (O-Net) – Bbox + Landmarks
   1) Similar to the second stage.
   2) Output five facial landmarks' positions.

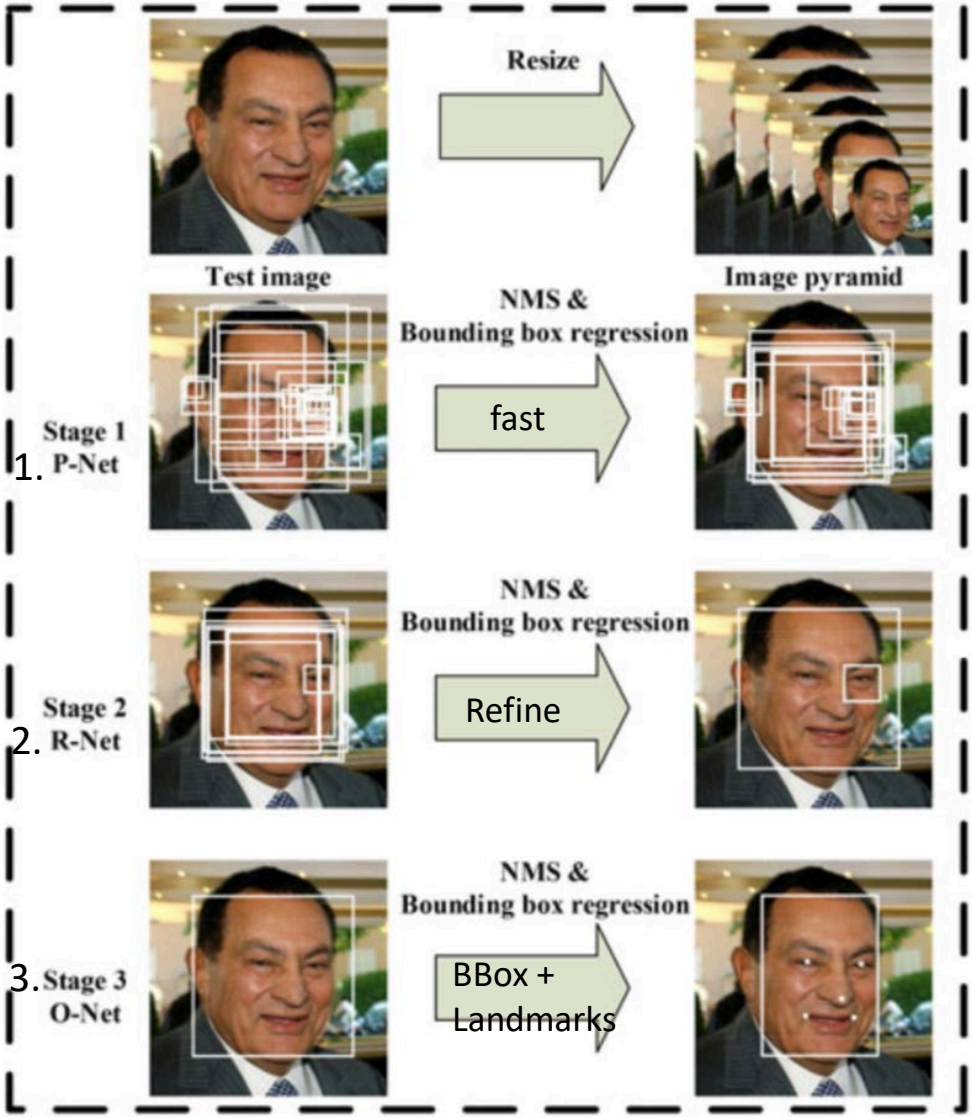



Fig. 1. Pipeline of our cascaded framework that includes three-stage multita
deep convolutional networks. First, candidate windows are produced throu
a fast P-Net. After that, we refine these candidates in the next stage throu
a R-Net. In the third stage, the O-Net produces final bounding box and fac
landmarks position.

# 1.2 Related Works (Example)

1. Camera Localization Methods

| Method Names | Advantages | Disadvantages |
|---|---|---|
| 1.1 **Dsac – Differentiable RANSAC CNN []** | 1) Which can be trained end-to-end<br>2) Using differentiable RANSAC for selecting hypothesis, which improve the accuracy of camera pose prediction | 1) Scoring CNN is porne to overfit<br>2) Requiring RGB-D training data or a 3D model of the scene to generate ground truth for Scene CNN |
| 1.2 **Dsac* - Visual Camera Re-Localization** | 1) The output of this network have a limited receptive field, preserving the patch-based nature of scene prediction<br>2) Selecting hypothesis depends on soft inlier count is more reliable than Score CNN | 1) Based on the RGB images, lose accuracy in some situation with fewer texture features, such as stairs and wall |

# 1.3 Contribution

- Contributions
  1) A new cascaded multitask framework that exploits the inherent correlation between detection and alignment to boost up their performance.
  2) An effective method to conduct online hard sample mining.

| Methods | Problems to be solved | Contribution |
| --- | --- | --- |
| | | |
| | | |
| | | |

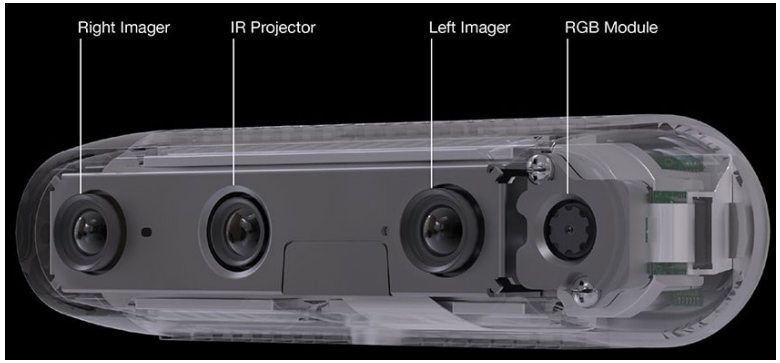
# 2.1 System Setup

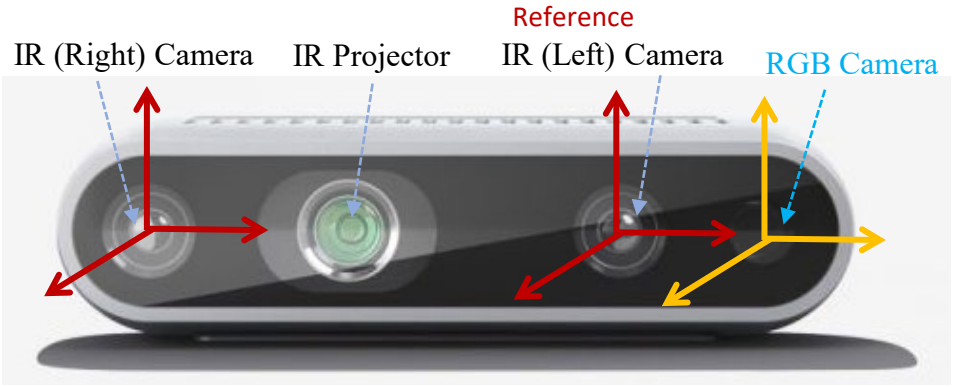
# 2.2 Hardware Specification and Function (Example)

1.1 RGB-D Camera: Intel RealSense Depth Camera-D435

| RGB Sensor | W | H |
|---|---|---|
| Resolution | 1920 | 1080p |
| pixel size | 3 um | 3 um |
| FOV | 69.4° x 42.5° x 77° | |
| Frame Rate | 30 fps | |

| Depth Sensor | W | H |
|---|---|---|
| Resolution (pixel*pixel) | 1280 | 720p |
| FOV | 85.2° x 58° x 94° (+/- 3) | |
| Frame Rate | 30 fps | |
| Z-axis Accuracy | 1 mm (increased by depth) | |
| Depth Technology | Active IR stereo | |
| Minimum Distance | 0.11m | |
| Maximum Distance | Approx. 10 meters | |





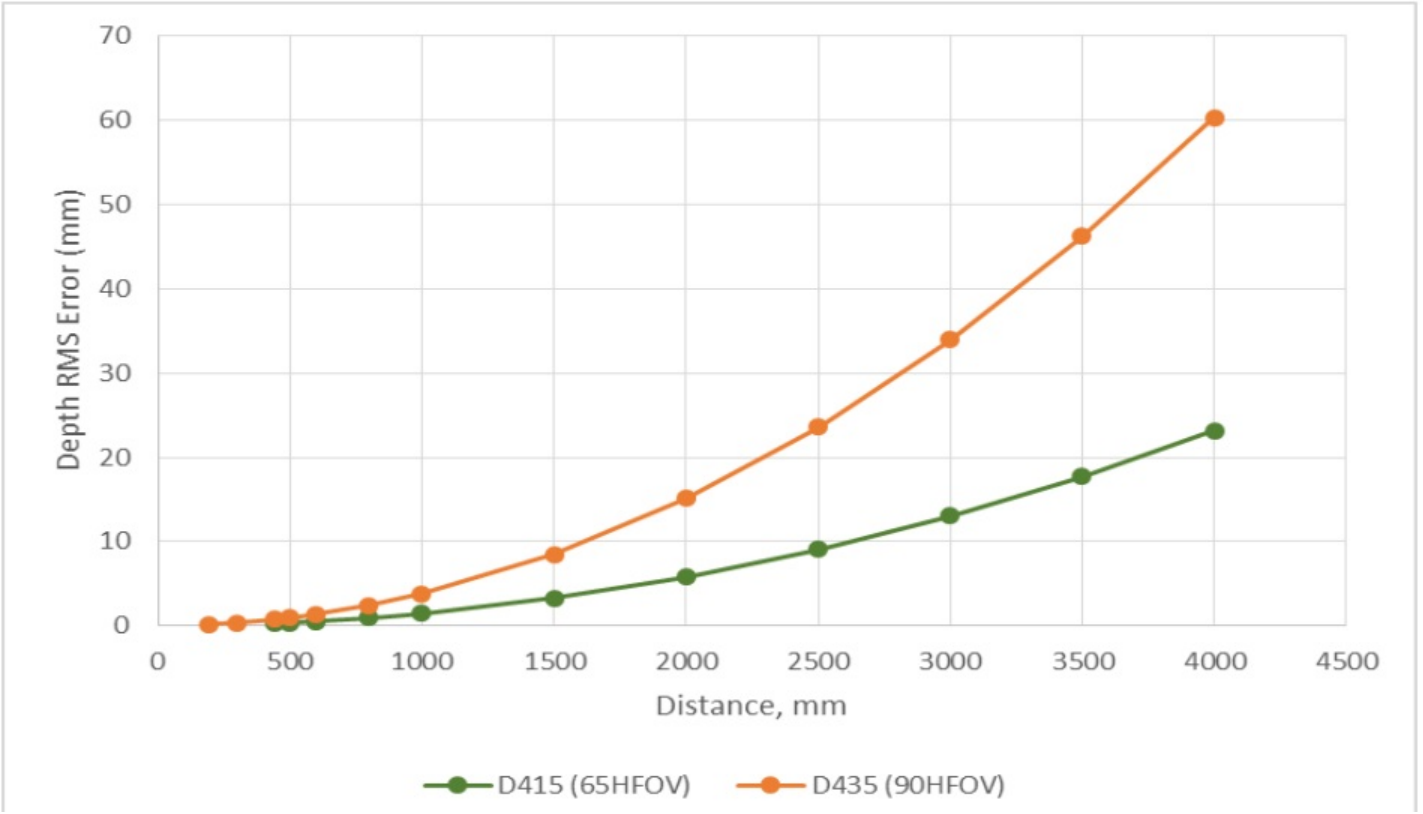

Source : https://click.intel.com/intelr-realsensetm-depth-camera-d435.html

# 2.2 Hardware Specification and Function (Example)

1.2 RGB-D Camera – Intel RealSense D435


## UNDERSTAND THEORETICAL LIMIT

The graph is obtained using subpixel=0.08:
D415 with HFOV=65deg, Xres=1280, and baseline=55mm,
D435 with HFOV=90deg, Xres=848, and baseline=50mm






Reference: https://realsense.intel.com/wp-content/uploads/sites/63/BKM-For-Tuning-D435-and-D415-Cameras-Webinar_Rev3.pdf

# 3.0 System Framework:

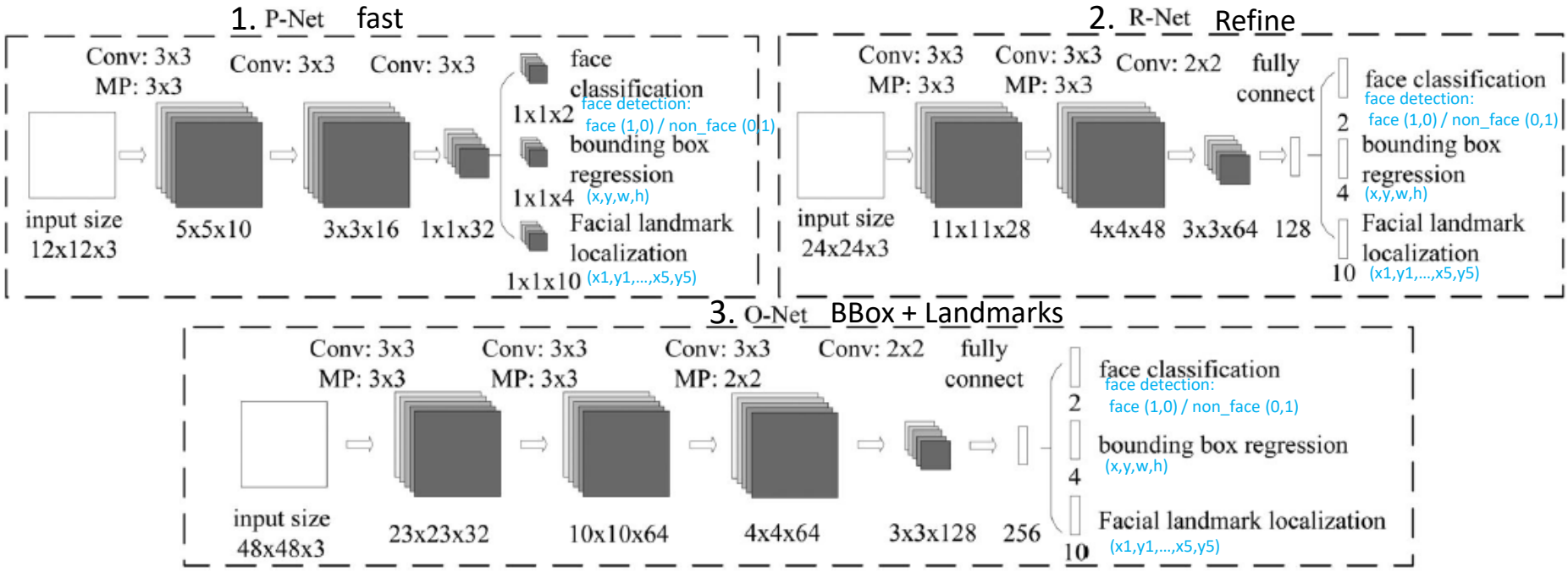



Fig. 2. Architectures of P-Net, R-Net, and O-Net, where "MP" means max pooling and "Conv" means convolution. The step size in convolution and pooling is 1 and 2, respectively.

# 3.1.1 Training Framework: P-Net (Proposal-Net) (1/3)

$M_{PNet}$

## 1. Training P-Net:

### 1.1 Data Collection for P-Net:

1) WIDER FACE
   (1) Number of images:
      - $N_{trainW}$ = 12,880 images
   (2) Generate data: $D_{trainW}$
      - Crop images randomly
      - Calculate IoU
      - 3 Labels
        a) Positives: IoU above 0.65
        b) Part: IoU between 0.4 and 0.65
        c) Negatives: IoU less than 0.3
2) AFLW
   (1) Number of images:
      - $N_{trainA}$ = 24,386 images
   (2) Crop faces as landmark faces $D_{trainA}$
      - Label five landmarks' positions
3) Neg: Pos : Part : Landmark = 3:1:1:2

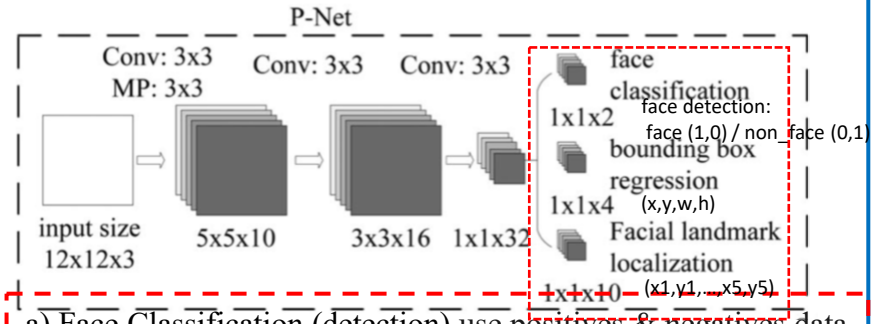
$D_{trainW}$, $D_{trainA}$

### 1.2 Data Resize :

1) Input : $D_{trainW}$, $D_{trainA}$
   - Each cropped images
2) Output : $D'_{train}$
   - resize to 12*12*3

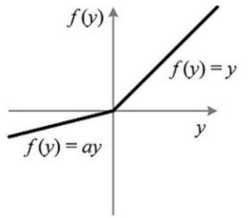
$D'_{train}$

### 1.3 Training P-Net:




a) Face Classification (detection) use positives & negatives data
b) Bounding Box use positives & part data
c) Landmark localization use landmark faces data

1) Training model: $M_{PNet}$
   (1) 4 convolutional layers
      - Kernel size = 3*3
   (2) 1 max-pooling layers
      - Kernel size = 3*3
      - Stride = 2
   (3) Activation function : PReLU




$$PReLU(x_i) = \begin{cases} x_i & if\ x_i > 0 \\ a_i x_i & if\ x_i \le 0 \end{cases}$$

   (4) Stochastic gradient descent (SGD)

2) Training process: Loss Function
   (1) Face Classification (Face / Non-Face) — Discrete (Entropy)

$$L_i^{det} = -\left(y_i^{det} \log(p_i) + \left(1 - y_i^{det}\right)(1 - \log(p_i))\right)$$

   (2) Bounding Box Regression — Continue (MSE)

$$L_i^{box} = \left\| \hat{y}_i^{box} - y_i^{box} \right\|_2^2$$

   (3) Facial Landmark Localization — Continue (MSE)

$$L_i^{landmark} = \left\| \hat{y}_i^{landmark} - y_i^{landmark} \right\|_2^2$$

   (4) Combine (1) to (3)

$$\min \sum_{i=1}^{N} \sum_{j \in \{det, box, landmark\}} \alpha_j \beta_i^j L_i^j$$

$\alpha_{det} = 1, \alpha_{box} = 0.5, \alpha_{landmark} = 0.5$

$\beta_i^j \in \{0, 1\}$

# 3.1.1 Training Framework: R-Net (Refine-Net) (2/3)

$M_{RNet}$

## 2. Training R-Net:

$M_{PNet}$

### 2.1 Data Collection for R-Net:

1) WIDER FACE
   (1) Number of images:
   - $N_{trainW}$ = 12,880 images

   (2) Generate data: $D_{trainW}$
   - Output bounding box from $M_{PNet}$
   - Crop bounding box
   - Calculate IoU
   - 3 Labels
     a) Positives : IoU above 0.65
     b) Part : IoU between 0.4 and 0.65
     c) Negatives : IoU less than 0.3

2) AFLW
   (1) Number of images:
   - $N_{trainA}$ = 24,386 images

   (2) Crop faces as landmark faces $D_{trainA}$
   - Label five landmarks' positions

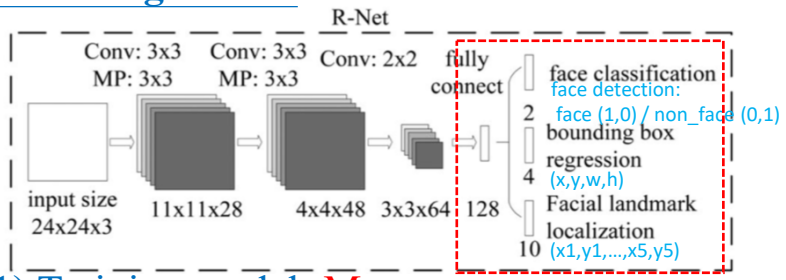
$D_{trainW}$, $D_{trainA}$

$D'_{train}$

### 2.2 Data Resize :

1) Input :$D_{trainW}$, $D_{trainA}$
   - Each cropped images
2) Output :$D'_{train}$
   - resize to 24*24*3

### 2.3 Training R-Net:

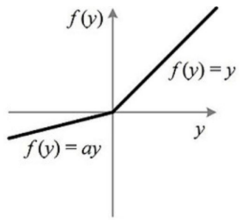



1) Training model: $M_{RNet}$
   (1) 3 convolutional layers
   - Kernel size
     a) Layer 1&2: 3*3
     b) Layer 3: 2*2

   (2) 2 max-pooling layers
   - Kernel size =3*3
   - Stride = 2

   (3) 2 fully connect layers

   (4) Activation function : PReLU




$$PReLU(x_i) = \begin{cases} x_i & if\, x_i > 0 \\ a_i x_i & if\, x_i \le 0 \end{cases}$$

   (5) Stochastic gradient descent (SGD)

2) Training process: Loss Function

   (1) Face Classification (Face / Non-Face) — Discrete (Entropy)

$$L_i^{det} = -\left(y_i^{det} \log(p_i) + (1 - y_i^{det})(1 - \log(p_i))\right)$$

   (2) Bounding Box Regression — Continue (MSE)

$$L_i^{box} = \left\| \hat{y}_i^{box} - y_i^{box} \right\|_2^2$$

   (3) Facial Landmark Localization — Continue (MSE)

$$L_i^{landmark} = \left\| \hat{y}_i^{landmark} - y_i^{landmark} \right\|_2^2$$

   (4) Combine (1) to (3)

$$\min \sum_{i=1}^{N} \sum_{j \in \{det, box, landmark\}} \alpha_j \beta_i^j L_i^j$$

$\alpha_{det} = 1, \alpha_{box} = 0.5, \alpha_{landmark} = 0.5$

$\beta_i^j \in \{0, 1\}$

# 3.1.1 Training Framework: O-Net (Output-Net) (3/3)

$M_{ONet}$

## 3. Training O-Net:

$M_{PNet}$
$M_{RNet}$

### 3.1 Data Collection for O-Net:

1) WIDER FACE
   (1) Number of images:
   - $N_{trainW}$ = 12,880 images

   (2) Generate data: $D_{trainW}$
   - Put images though $M_{PNet}$ and $M_{RNet}$
   - Output bounding box from $M_{RNet}$
   - Crop bounding box and calculate IoU
   - 3 Labels
     a) Positives : IoU above 0.65
     b) Part : IoU between 0.4 and 0.65
     c) Negatives : IoU less than 0.3

2) AFLW
   (1) Number of images:
   - $N_{trainA}$ = 24,386 images

   (2) Crop faces as landmark faces $D_{trainA}$
   - Label five landmarks' positions

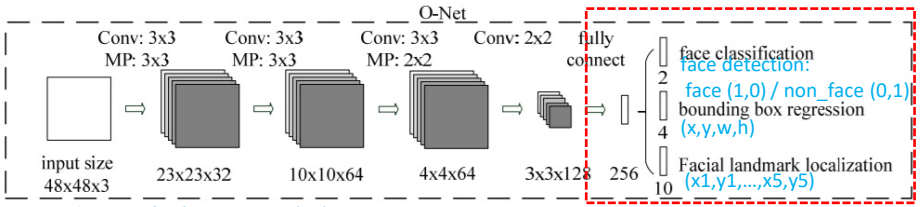
$D_{trainW}$, $D_{trainA}$

### 3.2 Data Resize :

1) Input :$D_{trainW}$, $D_{trainA}$
   - Each cropped images
2) Output :$D'_{train}$
   - resize to 48*48*3

$D'_{train}$

## 3.3 Training O-Net:



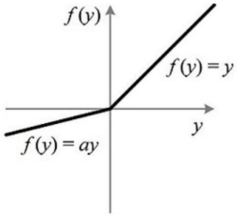

1) Training model: $M_{RNet}$
   (1) 4 convolutional layers
   - Kernel size
     a) Layer 1~3: 3*3
     b) Layer 4: 2*2

   (2) 3 max-pooling layers
   - Kernel size
     a) Layer 1&2: 3*3
     b) Layer 3: 2*2
   - Stride = 2

   (3) 2 fully connect layers

   (4) Activation function : PReLU

   (5) Stochastic gradient descent




$$PReLU(x_i) = \begin{cases} x_i & if\, x_i > 0 \\ a_i x_i & if\, x_i \le 0 \end{cases}$$

2) Training process: Loss Function

(1) Face Classification (Face / Non-Face) — Discrete (Entropy)

$$L_i^{det} = -\left(y_i^{det} \log(p_i) + \left(1 - y_i^{det}\right)(1 - \log(p_i))\right)$$

(2) Bounding Box Regression — Continue (MSE)

$$L_i^{box} = \left\| \hat{y}_i^{box} - y_i^{box} \right\|_2^2$$

(3) Facial Landmark Localization — Continue (MSE)

$$L_i^{landmark} = \left\| \hat{y}_i^{landmark} - y_i^{landmark} \right\|_2^2$$

(4) Combine (1) to (3)

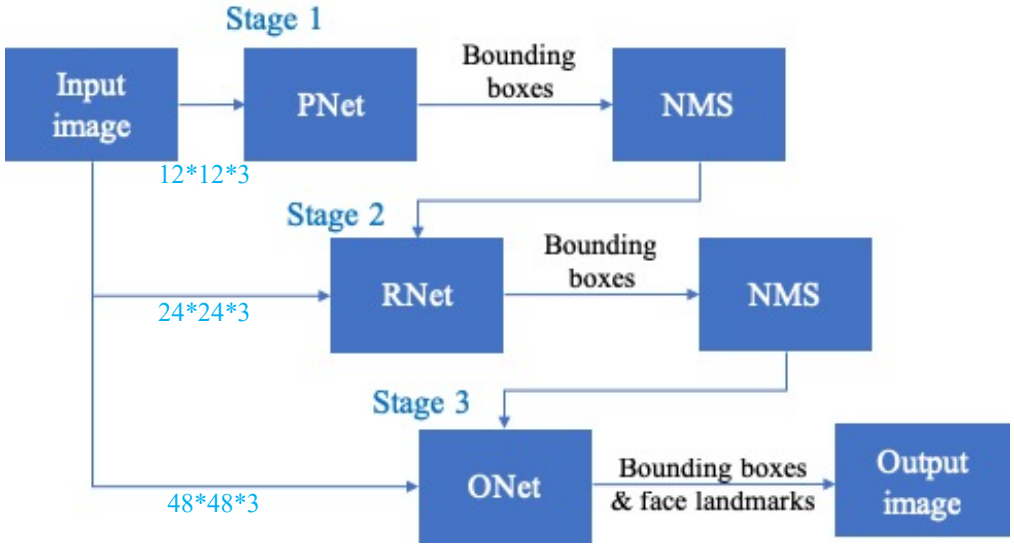
$$\min \sum_{i=1}^{N} \sum_{j \in \{det, box, landmark\}} \alpha_j \beta_i^j L_i^j$$

$\alpha_{det} = 1, \alpha_{box} = 0.5, \alpha_{landmark} = 1$

$\beta_i^j \in \{0, 1\}$


15

# 3.1.2 Inference Framework:



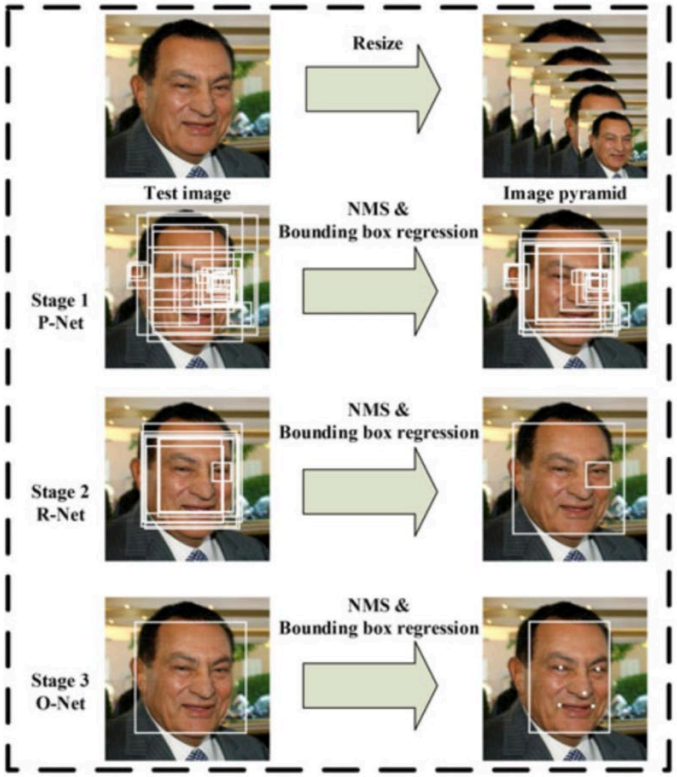
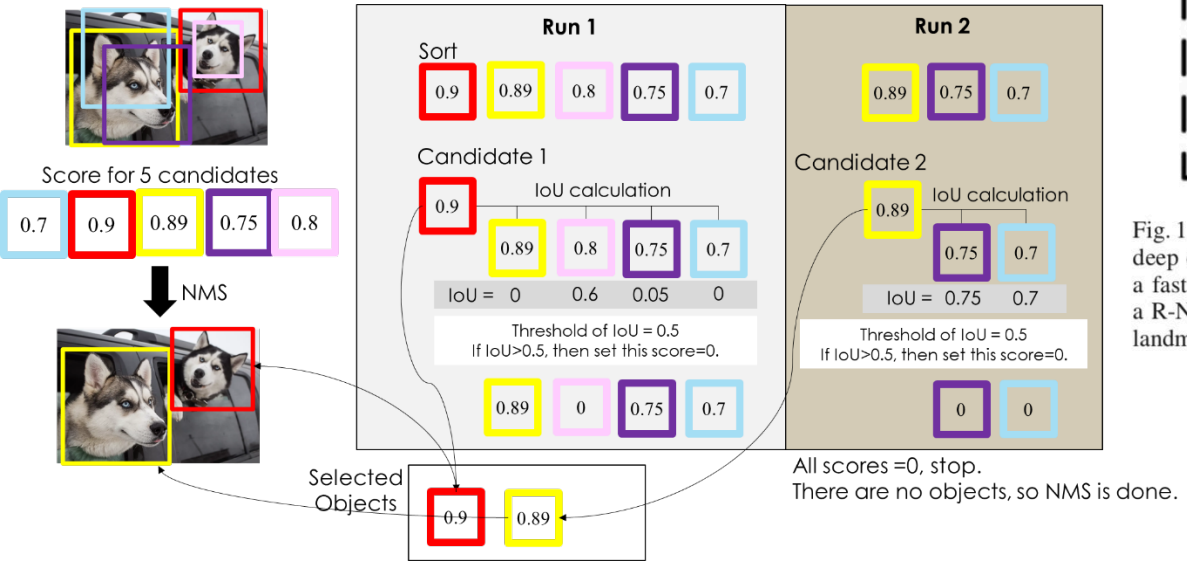




Fig. 1. Pipeline of our cascaded framework that includes three-stage multitask deep convolutional networks. First, candidate windows are produced through a fast P-Net. After that, we refine these candidates in the next stage through a R-Net. In the third stage, the O-Net produces final bounding box and facial landmarks position.

Continue (mse)

## Non-Maximum Suppression (NMS)



# 3.1.3 Loss Function:

# 3.2 YoloV4 (1/2) (Example)

- **YOLOv4 (One-stage Object Detector)**

(0) **Input** (input image):
- Image.

(1) **Backbone** (extracts the feature maps):
- CSPDarknet53 [81]. **Cross-Stage-Partial-connections (CSP):**

(2) **Neck** (enhance the feature discriminability and robustness):
- Additional blocks: Spatial Pyramid Pooling SPP [25].
- Path-aggregation blocks: modified PANet [49].

(3) **Head** (handles the Bboxes prediction and Categories classification):
- Dense Prediction (one-stage): same as YOLOv3 [61]

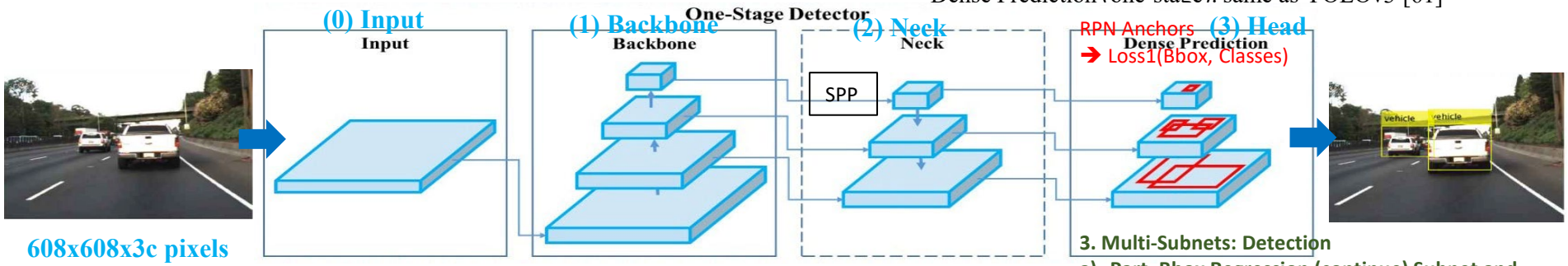



*Fig. Overview of one-stage object detector*

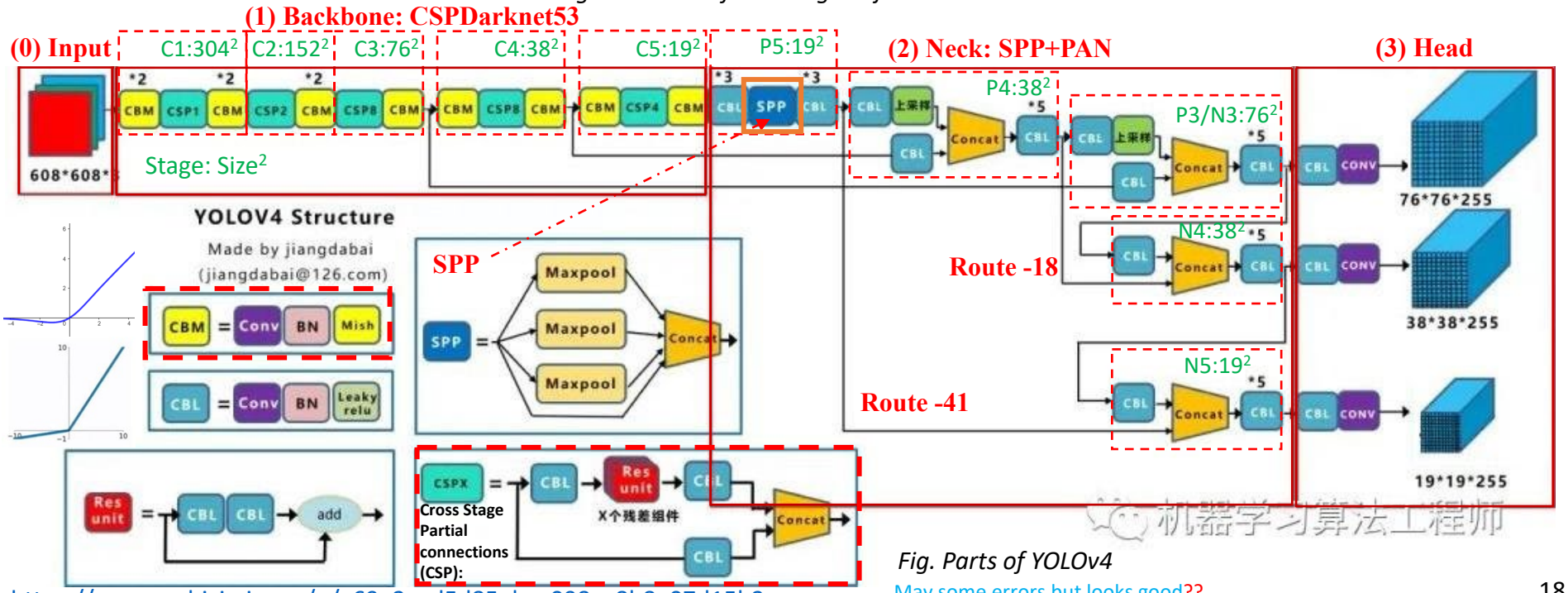


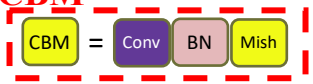

*Fig. Parts of YOLOv4*

May some errors but looks good??

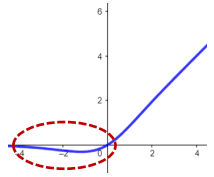
https://www.ershicimi.com/p/e69e2ecd5d85ebec098ce8b8e07d15b3

# 3.2 YoloV4 (2/2) (Example)

## 1) CBM

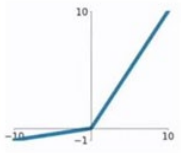


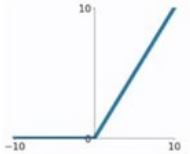

- Conv: Convolution Layer. It is used to extract features in images.
- BN: Batch Normalization. It normalizes each layer's inputs by fixing means and variances. It solves the gradient vanishing problem and achieves better convergence speed.
- Mish: Mish activation function. It is a smooth function and preserves small negative inputs, which is easier to optimize and hence the network generalizes better.


Mish

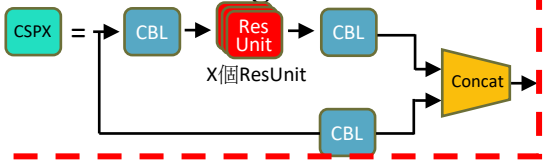
## 2) CBL: similar to CBM, but the activation function is Leaky ReLU instead of Mish.


- Leaky ReLU: It is a modified version of ReLU. It preserves negative inputs and solves gradient vanishing problem.


Leaky ReLU

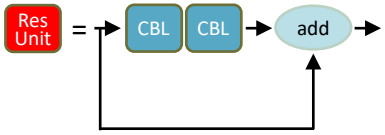

ReLU

## 3) CSPX: Cross Stage Partial Connections ("X" stands for the number of ResUnits)



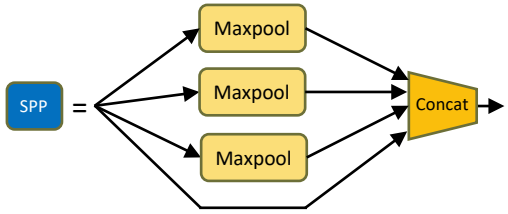

- Input feature maps will be divided into 2 parts. The first part goes through CBLs and ResUnits. The second part goes through a CBL and becomes part of the input to the next layer.
- It not only retains the details on the image but also reduces the computational complexity.

## 4) ResUnit: Residual Unit (the building block of ResNet)




- Input feature maps will go through 2 CBLs and a shortcut (not divided into 2 parts).
- It solves gradient vanishing problem and allows the network to get deeper.

## 5) SPP: Spatial Pyramid Pooling




- It is composed of multiple max-pooling layers.
- It increases receptive fields to get more features and separates out the most significant context features.

# 3.2.1 YoloV4: Backbone (1/2) (Example)

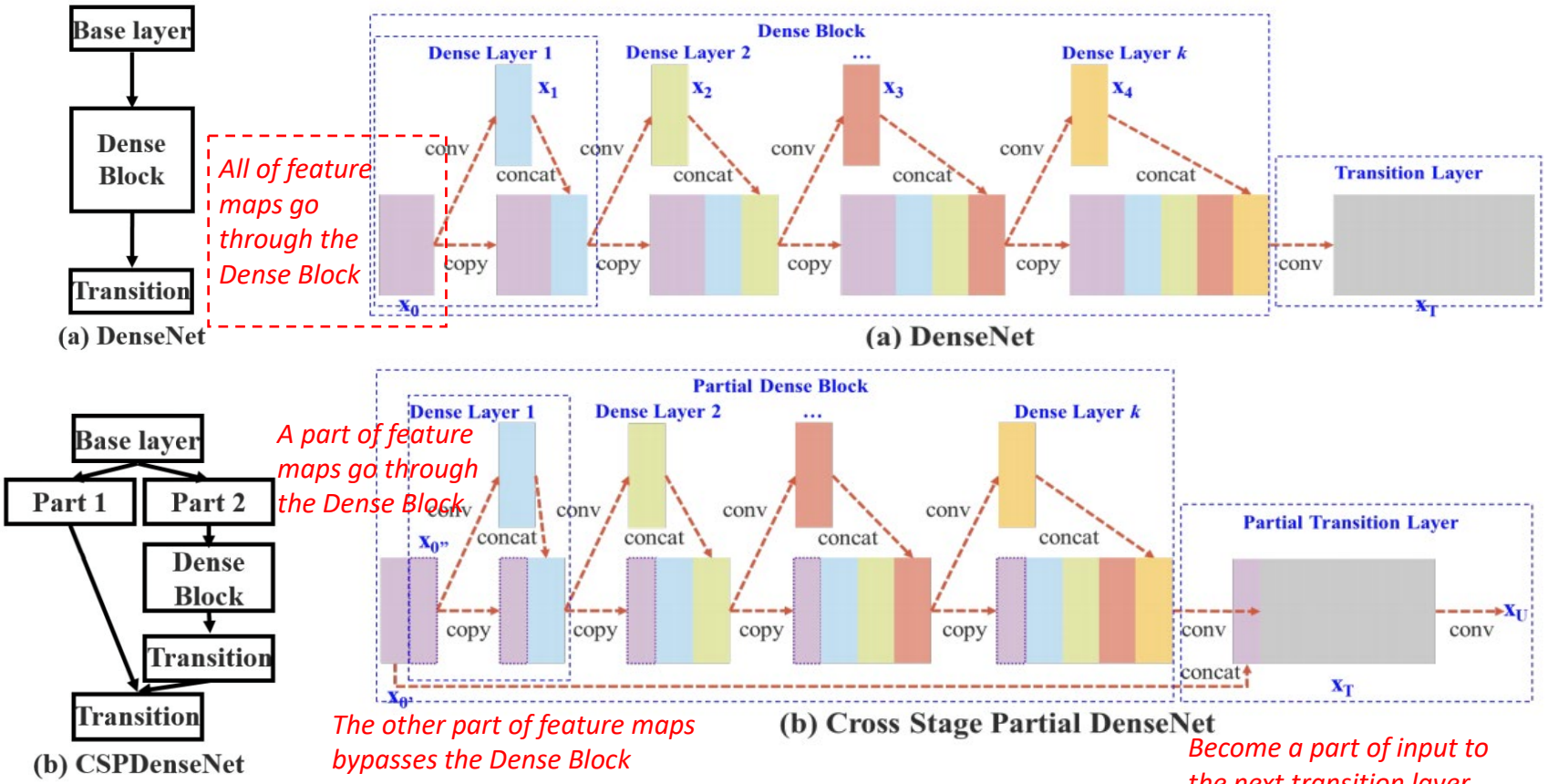
## 1) Cross Stage Partial connections (CSP)

- A **Dense Block:**
  - ❖ Each layer *Hi* takes the output of all previous layers as well as the original as its input.
  - ❖ At each layer, the number of feature maps is increased by a growth rate.
- **CSPNet** separates the input feature maps of the Dense Block into two parts.

(1) The first part $x_0'$ bypasses the Dense Block and becomes part of the input to the next transition layer.

(2) The second part $x_0''$ will go through the Dense block as below

- This new design reduces the computational complexity with only one part going through the Dense Block.



# 3.2.1 YoloV4: Backbone (2/2) (Example)

## 2) CSPDarknet53 model:

- Darknet-53 using the CSP connections.
- Has higher accuracy in object detection compared with ResNet-based designs.
- The classification accuracy of CSPDarknet53 can be improved (Mish and other techniques).

| | Type | Filters | Size | Output |
|---|---|---|---|---|
| | Convolutional | 32 | 3 × 3 | 256 × 256 |
| | Convolutional | 64 | 3 × 3 / 2 | 128 × 128 |
| 1× | Convolutional | 32 | 1 × 1 | |
| | Convolutional | 64 | 3 × 3 | |
| | Residual | | | 128 × 128 |
| | Convolutional | 128 | 3 × 3 / 2 | 64 × 64 |
| 2× | Convolutional | 64 | 1 × 1 | |
| | Convolutional | 128 | 3 × 3 | |
| | Residual | | | 64 × 64 |
| | Convolutional | 256 | 3 × 3 / 2 | 32 × 32 |
| 8× | Convolutional | 128 | 1 × 1 | |
| | Convolutional | 256 | 3 × 3 | |
| | Residual | | | 32 × 32 |
| | Convolutional | 512 | 3 × 3 / 2 | 16 × 16 |
| 8× | Convolutional | 256 | 1 × 1 | |
| | Convolutional | 512 | 3 × 3 | |
| | Residual | | | 16 × 16 |
| | Convolutional | 1024 | 3 × 3 / 2 | 8 × 8 |
| 4× | Convolutional | 512 | 1 × 1 | |
| | Convolutional | 1024 | 3 × 3 | |
| | Residual | | | 8 × 8 |
| | Avgpool | | Global | |
| | Connected | | 1000 | |
| | Softmax | | | |

Table 1. **Darknet-53.**

# 3.2.2 YoloV4: Neck (1/3) (Example)

文

- **Objectives: For the Neck, select additional blocks to increase the receptive field and the best method for parameter aggregation from different backbone levels for different detector levels.**
  - Additional Blocks: Add the SPP block over the CSPDarknet53
    - ❖ Increases the receptive field.
    - ❖ Separates out the most significant context features.
    - ❖ No reduction of the network operation speed.
  - Parameter aggregation: Use PANet as the method instead of the FPN.
    - ❖ Better than FPN in YOLOv3.

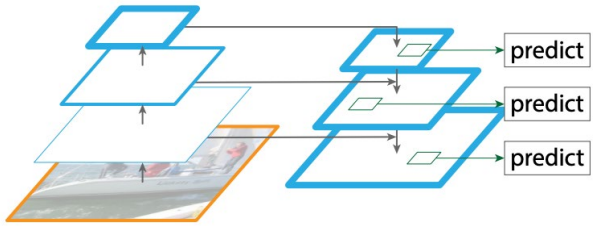



*Fig. FPN*

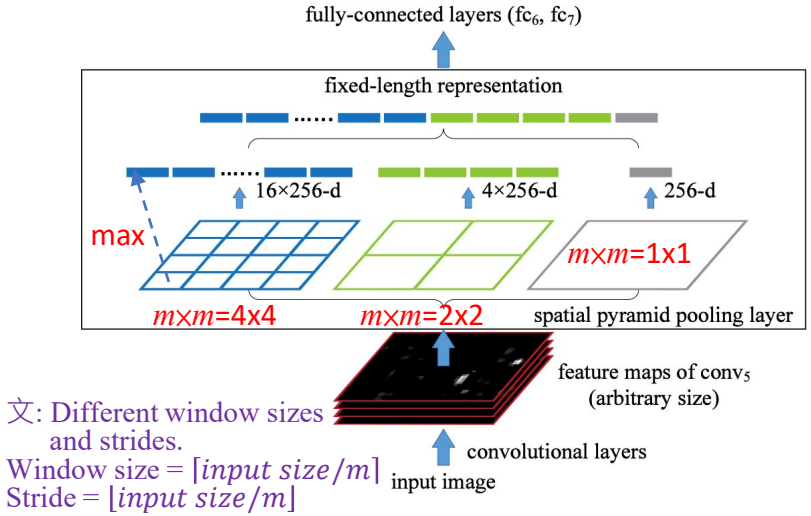



*Fig. Traditional Spatial Pyramid Pooling (SPP) Block.*

- A spatial pyramid pooling layer is added at the last set of feature maps
- These feature maps are spatially divided into $m \times m$ bins with $m$ equals 4, 2, and 1 respectively.
- Then a maximum pool is applied to each bin.
- This forms a fixed-length representation that can be further analyzed with FC-layers.

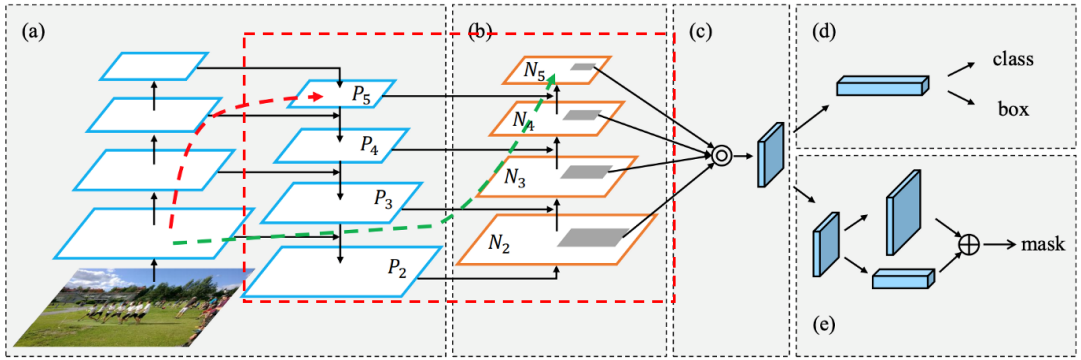


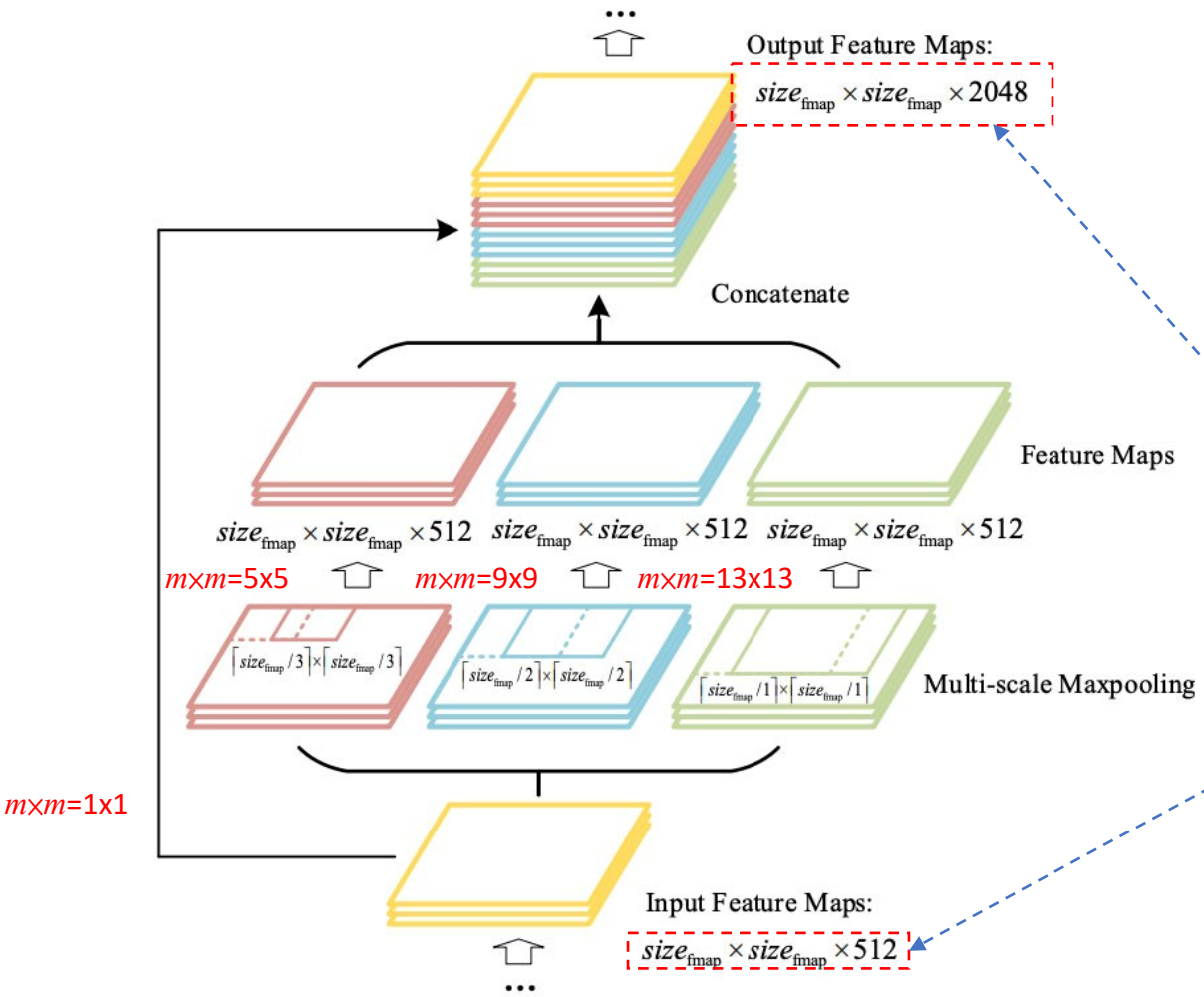

*Fig. PANet*

- A bottom-up path (b) is augmented to make low-layer information easier to propagate to the top.
- PAN introduced a short-cut path (the green path) which only takes about 10 layers to go to the top $N_5$ layer.
- This short-circuit concepts make fine-grain localized information available to top layers.

# 3.3.2 YoloV4: Neck (2/3) (Example)

**1) SPP in YOLO:**

- In YOLO, the SPP is modified to retain the output spatial dimension.
- A maximum pool (stride 1) is applied to a sliding kernel of size, 1×1, 5×5, 9×9, 13×13.
- The features maps from different kernel sizes are then concatenated together as output.
- The padding is utilized to keep a constant size of the output feature maps. 文




The spatial dimension is preserved but the ~~receptive filed~~ dimension is increased from 512 to 2,048.

# 3.2.3 YoloV4: Head (Example)

1) **YOLOv3 Head**
   - 9 anchor ratios (scale) by K-means for 3 stages (3 ratios/ stage)
   - Input channel size = (4 coordinates + 1 confidence score + 4 categories) x 3 = 27 channels/ stage

- 4 predicted Bbox offsets: $t_x$, $t_y$, $t_w$, $t_h$
- Confident score: c
- 4 detect type of class probability: $pc_0$ ,$pc_1$, $pc_2$, $pc_3$
- Pre-defined anchor box size: $p_w$, $p_h$
- 4 output Bbox coordinates: $b_x$, $b_y$, $b_w$, $b_h$

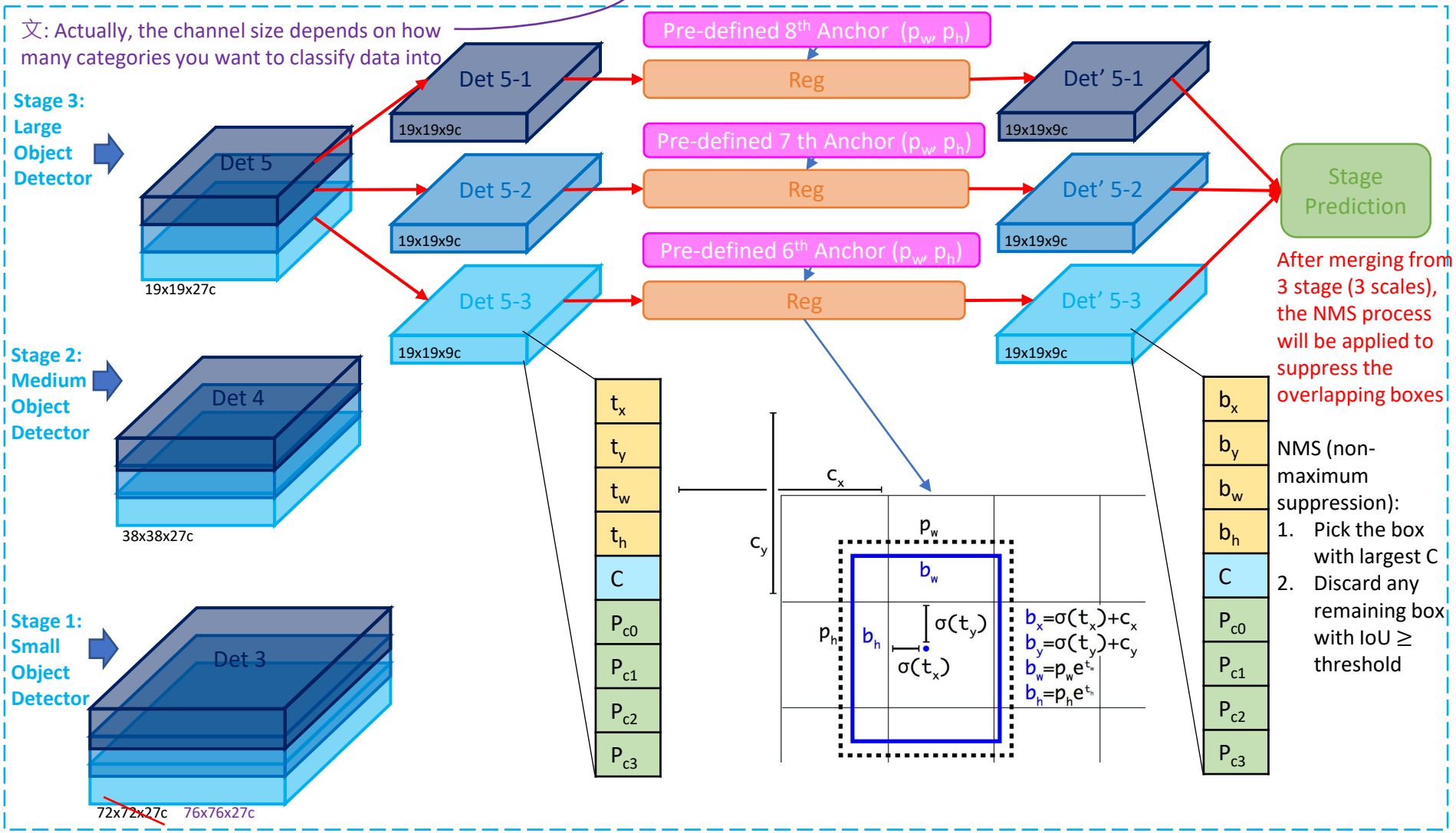



***Figure***. YOLOv4 output prediction

# 3.3 Online Hard Sample Mining

1. Traditional hard sample mining
   1) Doing after training original classifier.
2. Online hard sample mining
   1) Doing while training face/nonface classification task
   2) Method :
      (1) In each minibatch, sort the losses from all samples
      (2) Select the top 70% as hard samples
      (3) Compute the gradients from hard samples in the backward propagation
   3) Ignore the easy samples that are less helpful.

# 3.4 OOO

1. xxx

# 4.1.1 Data Collection: Face Detection

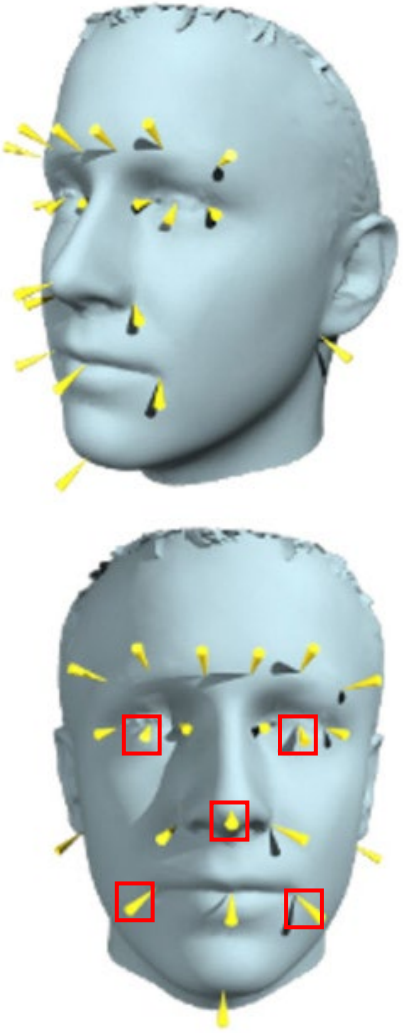
1. WIDER FACE
   - Consists of 393,703 labeled face bounding boxes in 32,203 images
     1) 80% (32,097 images) for training
     2) 10% ( 3,880 images) for validation
     3) 10% ( 3,226 images) for inference





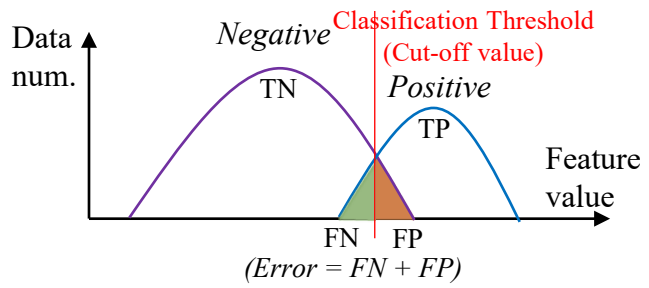
# 4.1.1 Data Collection: Face Landmark

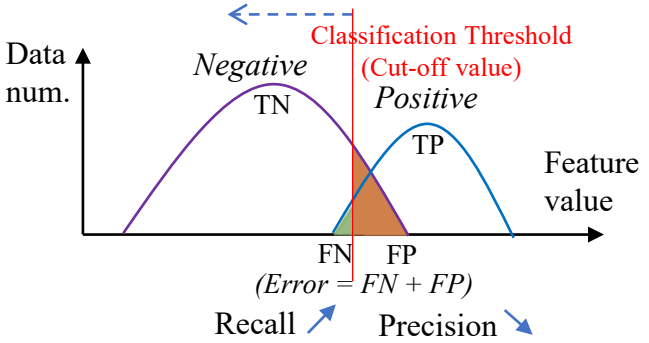
2. Annotated Facial Landmarks in the Wild (AFLW)
   1) Total 24,386 images
   2) Each image has one face with 21 landmarks
   3) Only take 5 landmarks for training (2 eyes, 1 nose, 2 mouth corners)



# 4.1.2 Metrics: Maximum F-measure (MF)

https://en.wikipedia.org/wiki/F1_score By Phu



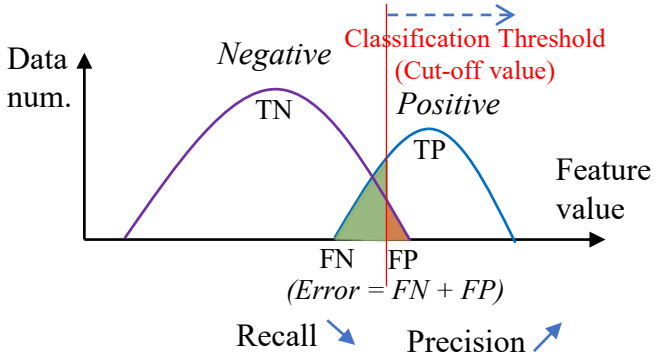
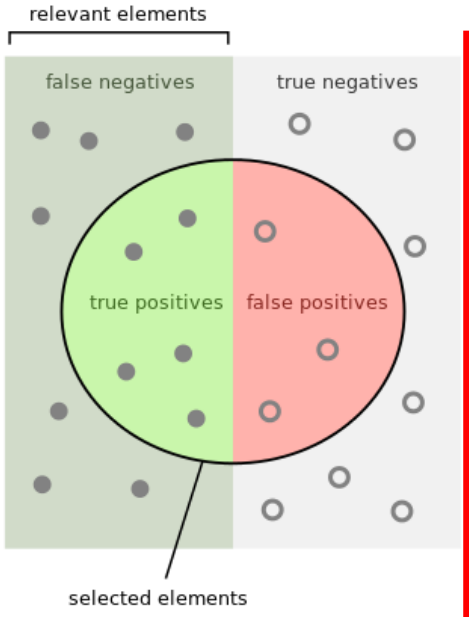




***Figure***. 3 types of Gaussian Distribution at different Classification Threshold (Cut-off)




How many selected items are relevant?

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{True\ Positive}{Total\ Detection}$$

How many relevant items are selected?

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{True\ Positive}{Total\ GroundTruth}$$

1. Classification Threshold move to the left (Reduction)
   - *FP* ⬆ FN ⬇ Recall ⬆ Precision ⬇
2. Classification Threshold move to the Right (Increasement)
   - *FP* ⬇ FN ⬆ Recall ⬇ Precision ⬆

Because of the **trade-off** between Recall and Precision, F1 score is utilized to define which is the most suitable confident score:

❖ ***F1 Score (DICE):*** The mean harmonic between Precision and Recall

$$F_1 = \left( \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} \right) = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

# 4.2 Experimental Result:

- Effectiveness of Online Hard Sample Mining
  1. Compare two P-Nets (with and without online hard sample mining)
  2. Online hard sample mining is beneficial to improve performance.
     1) Bring about 1.5% overall performance improvement.

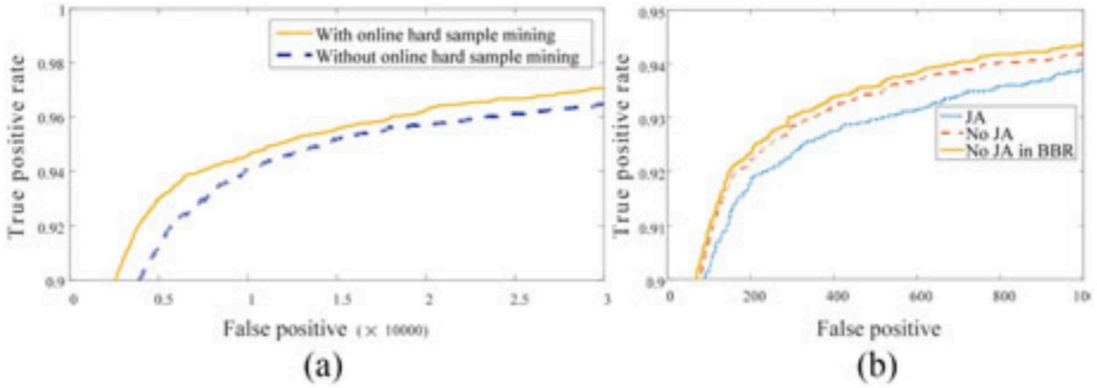



Fig. 3. (a) Detection performance of P-Net with and without online hard sample mining. (b) "JA" denotes joint face alignment learning in O-Net while "No JA" denotes do not joint it. "No JA in BBR" denotes use "No JA" O-Net for bounding box regression.

- Effectiveness of Joint Detection and Alignment
  1. Compare with two different O-Nets (same P-Net and R-Net)
     1) Joint facial landmarks regression learning
     2) Do not joint facial landmarks regression learning
  2. Joint landmark localization task help to enhance the performance

# 4.2.1 Experimental Result 1:

- Evaluation on Face Detection

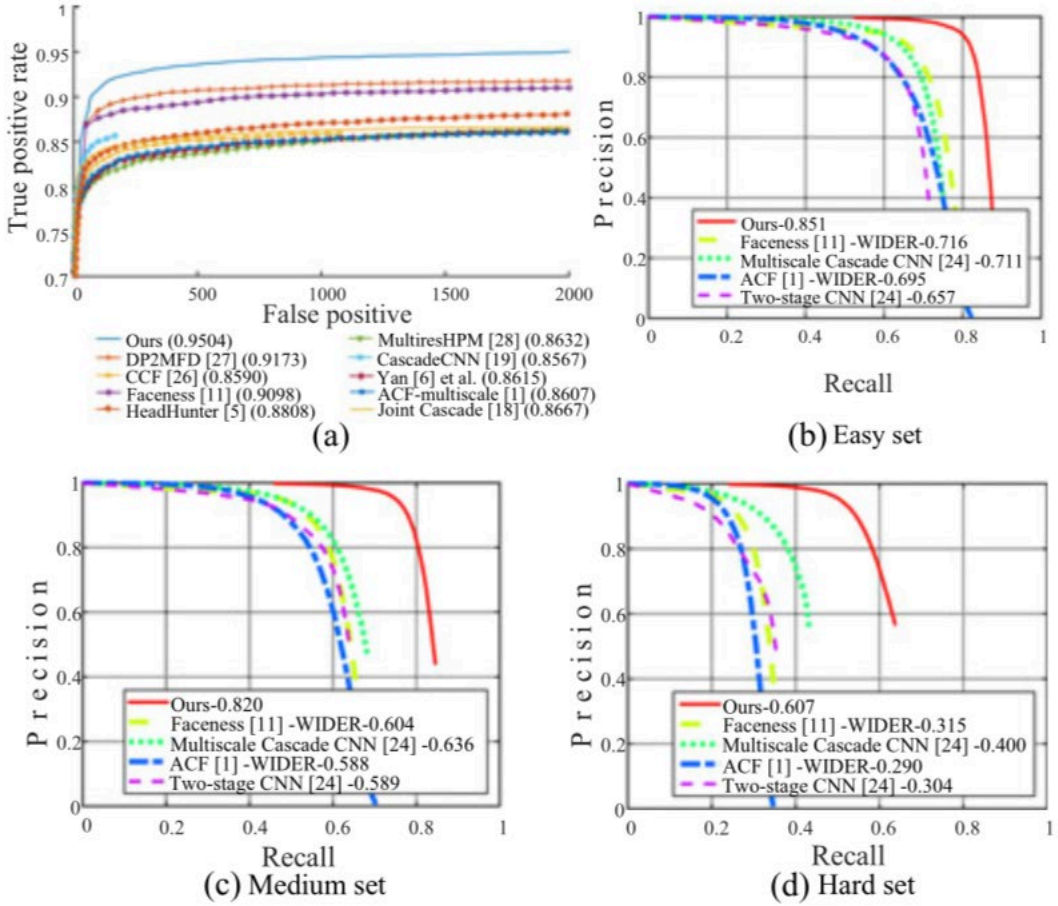



Fig. 4. (a) Evaluation on FDDB. (b)–(d) Evaluation on three subsets of WIDER FACE. The number following the method indicates the average accuracy.

# 4.2.2 Experimental Result 2:

- Evaluation on Face Alignment
  1. Mean error
     1) The distances between the estimated landmarks and the ground truths.
     2) normalized with respect to the interocular distance.

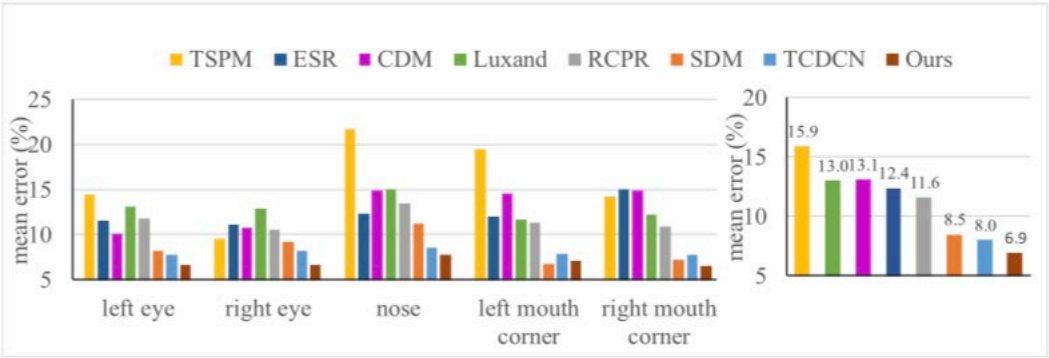



Fig. 5. Evaluation on AFLW for face alignment.

- Runtime Efficiency
  1. This method can achieve high speed.
  2. Compare with others on GPU.

TABLE II
SPEED COMPARISON OF OUR METHOD AND OTHER METHODS

| Method | GPU | Speed |
|---|---|---|
| Ours | Nvidia Titan Black | 99 FPS |
| Cascade CNN [19] | Nvidia Titan Black | 100 FPS |
| Faceness [11] | Nvidia Titan Black | 20 FPS |
| DP2MFD [27] | Nvidia Tesla K20 | 0.285 FPS |

# 4.3.1 Experimental Result: (Example)

IOU_Threshold = 0.2

| Database Ln Detection (Unit: 3D Nodule) (Testing set) | | | |
|---|---|---|---|
| Method | Recall (%) | Precision (%) | F1-score (%) |
| 1) Unet + Dice Loss | 23 / 27 = 85.18 | 23 / 177 = 12.99 | 22.54 |
| 2) 1) + Lung Lobe Segm | 23 / 27 = 85.18 | 23 / 158 = 14.56 | 24.87 |
| 3) 2) + 3D Nodule Size + Mask Confidence (last) | 22 / 27 = 81.48 | 22 / 35 = 62.86 | 70.96 |
| 4) Unet + Attention Gate + Dice Loss ?? | | | |
| 5) Unet + Attention Gate + Focal Loss ?? | | | |

**Lung Segm**: Labeling the Lung region (ignore nodule outside lung)

**3D Nodule Size**: Number of 3D nodule pixel (ignore small size nodule)

**Mask Confidence**: Average heapmap of nodule region (ignore small confidence nodule)

# 4.3.2 Experimental Result: (Example)

| Process Time | | |
|---|---|---|
| Process | Data Type | Time |
| Training Model | Ln Database (Training set) | 1.8 hr / epoch |
| | Me Database (Training set) | 3.5 hr / epoch |
| Inference Model | Ln Database (Testing set) | 1.5 hr |
| | Me Database (Testing set) | 3.0 hr |
| Evaluate Model | Ln Database (Testing set) | 1.2 hr |
| | Me Database (Testing set) | 3.0 hr |

**Ps. Evaluate Model -> Every threshold "t" (Not include "IOU Threshold" 、"Nodule Size Threshold"、"Confidence Threshold")**

# 5. Conclusion and Future Works

# 6.0 References (1/2)


[1] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Aggregate channel features for multi-view face detection," in *IEEE Int. Joint Conf. Biometrics*, 2014, pp. 1–8.

[2] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.

[3] M. T. Pham, Y. Gao, V. D. D. Hoang, and T. J. Cham, "Fast polygonal integration and its application in extending Haar-like features to improve object detection," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 942–949.

[4] Q. Zhu, M. C. Yeh, K. T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *IEEE Comput. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1491–1498.

[5] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool, "Face detection without bells and whistles," in *Eur. Conf. Comput Vis.*, 2014, pp. 720–735.

[6] J. Yan, Z. Lei, L. Wen, and S. Li, "The fastest deformable part model for object detection," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2497–2504.

[7] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2879–2886.

[8] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2011, pp. 2144–2151.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[10] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Adv. Neural Inf. Process. Syst.*, 2014, pp. 1988–1996.

[11] S. Yang, P. Luo, C. C. Loy, and X. Tang, "From facial parts responses to face detection: A deep learning approach," in *IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3676–3684.

[12] X. P. Burgos-Artizzu, P. Perona, and P. Dollar, "Robust face landmark estimation under occlusion," in *IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1513–1520.

[13] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face alignment by explicit shape regression," *Int. J. Comput. Vis.*, vol. 107, no. 2, pp. 177–190, 2012.

[14] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, Jun. 2001.

[15] X. Yu, J. Huang, S. Zhang, W. Yan, and D. Metaxas, "Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model," in *IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1944–1951.

[16] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment," in *Eur. Conf. Comput. Vis.*, 2014, pp. 1–16.

[17] Luxand Incorporated: Luxand face SDK. [Online]. Available: http://www.luxand.com/

[18] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun, "Joint cascade face detection and alignment," in *Eur. Conf. Comput. Vis.*, 2014, pp. 109–122.

[19] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5325–5334.

[20] C. Zhang and Z. Zhang, "Improving multiview face detection with multi-task deep convolutional neural networks," in *IEEE Winter Conf. Appl. Comput. Vis.*, 2014, pp. 1036–1041.

[21] X. Xiong and F. Torre, "Supervised descent method and its applications to face alignment," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 532–539.

[22] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *Eur. Conf. Comput. Vis.*, 2014, pp. 94–108.

[23] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3730–3738.

[24] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A Face detection benchmark," arXiv:1511.06523.

[25] V. Jain and E. G. Learned-Miller, "FDDB: A benchmark for face detection in unconstrained settings," Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. UMCS-2010-009, 2010.

[26] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Convolutional channel features," in *IEEE Int. Conf. Comput. Vis.*, 2015, pp. 82–90.

[27] R. Ranjan, V. M. Patel, and R. Chellappa, "A deep pyramid deformable part model for face detection," in *IEEE Int. Conf. Biometrics Theory, Appl. Syst.*, 2015, pp. 1–8.

# 6.0 References (2/2)


[28] G. Ghiasi and C. C. Fowlkes, "Occlusion coherence: Detecting and localizing occluded faces," arXiv:1506.08347.

[29] S. S. Farfade, M. J. Saberian, and L. J. Li, "Multi-view face detection using deep convolutional neural networks," in *ACM Int. Conf. Multimedia Retrieval*, 2015, pp. 643–650.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.