

LECTURE SERIES FOR DIGITAL SURVEILLANCE  
SYSTEMS AND APPLICATION

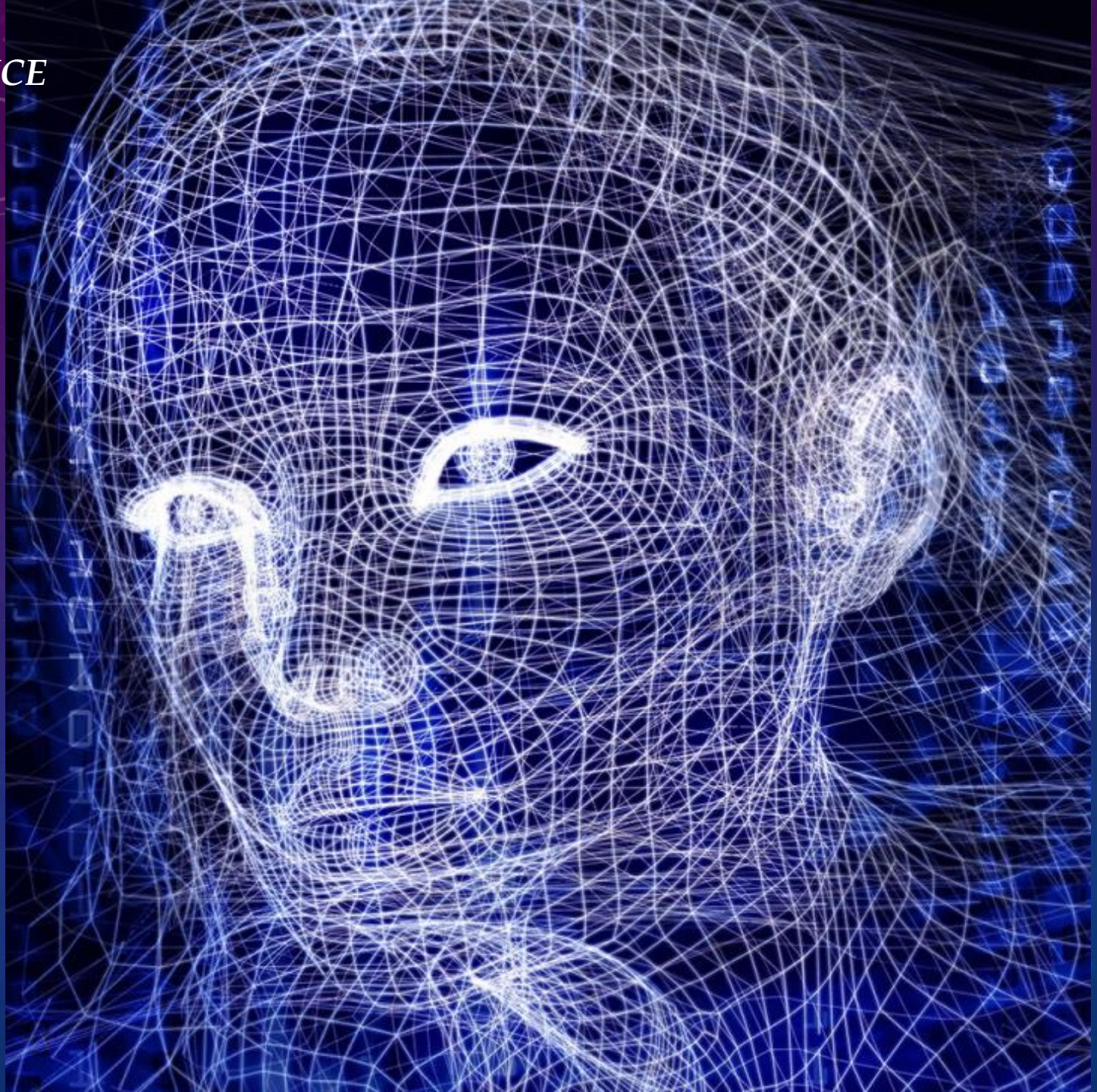
## Chapter 6

# *How a Deep Neural Net Sees the World and Face Detection*

徐繼聖

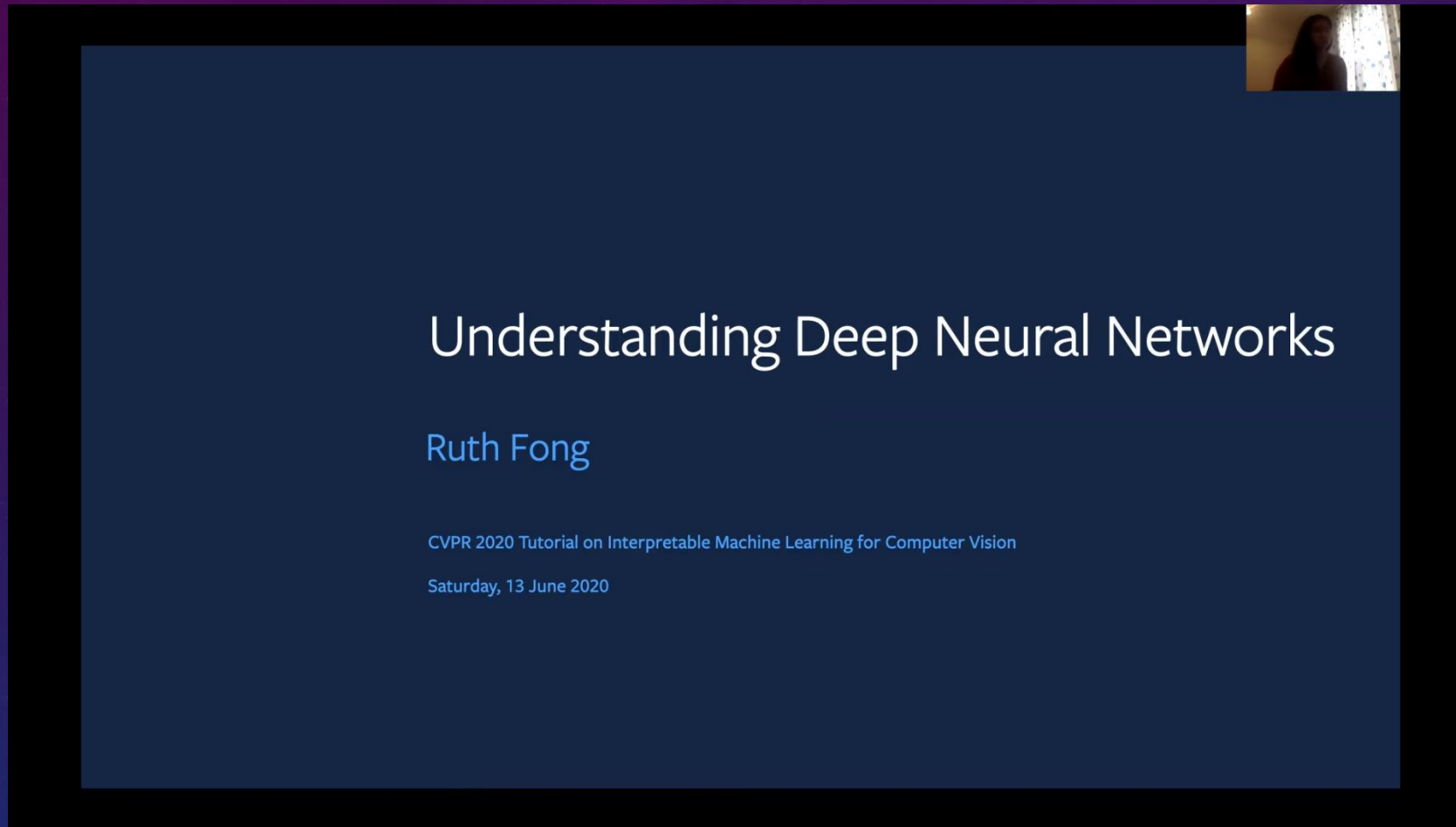
Gee-Sern Jison Hsu

National Taiwan University of Science  
and Technology





# Tutorial From CVPR 2020



(5) CVPR'20 iMLCV tutorial: Understanding Deep Neural Networks by Ruth C. Fong - YouTube

[40:52]

# Contents

- Database for Detection
- Detection Evaluation Method
- Face Detection

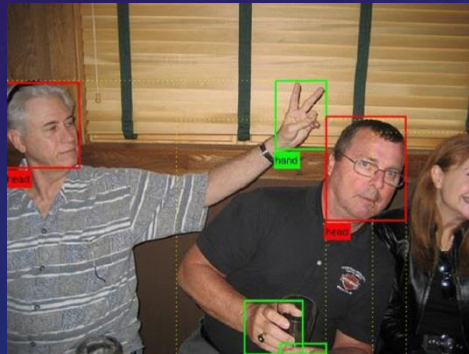
# Popular Database for Detection

# PASCAL

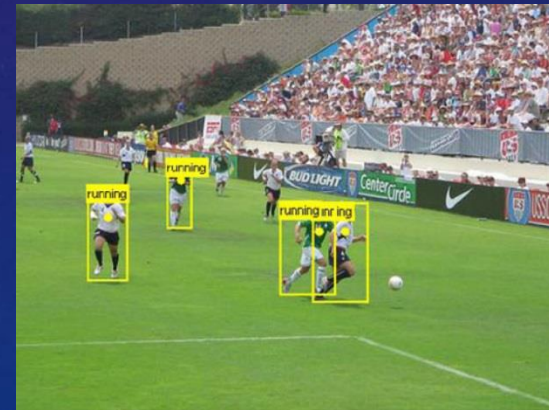
- PASCAL Visual Object Classes (VOC): an EU funded project
- PASCAL: pattern analysis, statistical modeling and computational learning
- common ground for measuring and comparing performance of competing algorithms

## Five Challenges

- 1) Classification
- 2) Action classification—“what action is being performed by an indicated person in this image?”
- 3) Detection
- 4) Person layout—“where are the head, hands and feet of people in this image?”
- 5) Segmentation



person layout



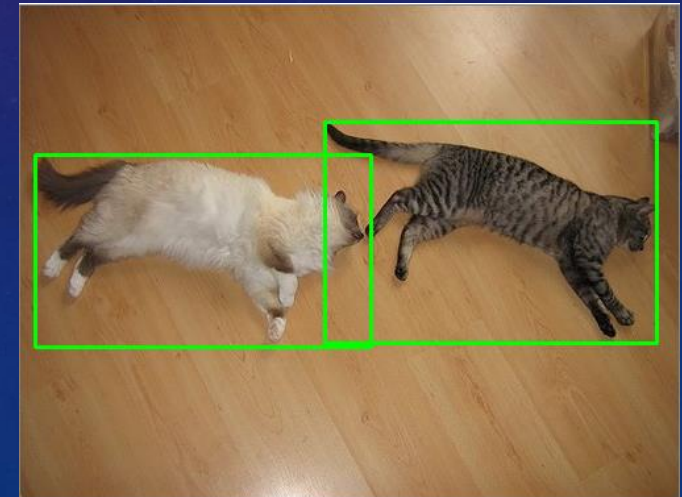
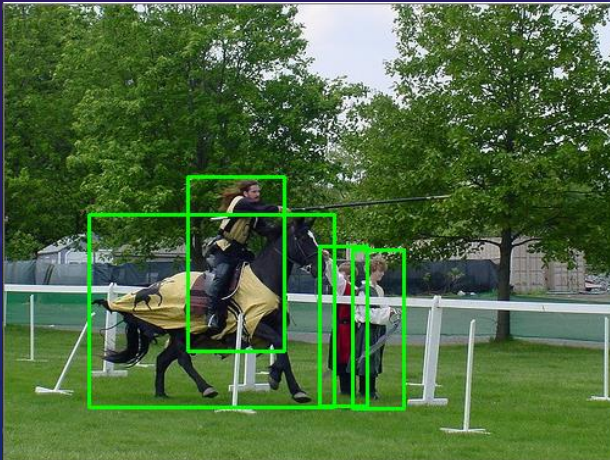
action classification



# VOC (Visual Object Classes)

Each object is represented by –

- a label (20 classes)
- a bounding box in absolute boundary coordinates
- a perceived detection difficulty (either 0, meaning not difficult, or 1, meaning difficult)



# VOC (Visual Object Classes)

20 Classes :

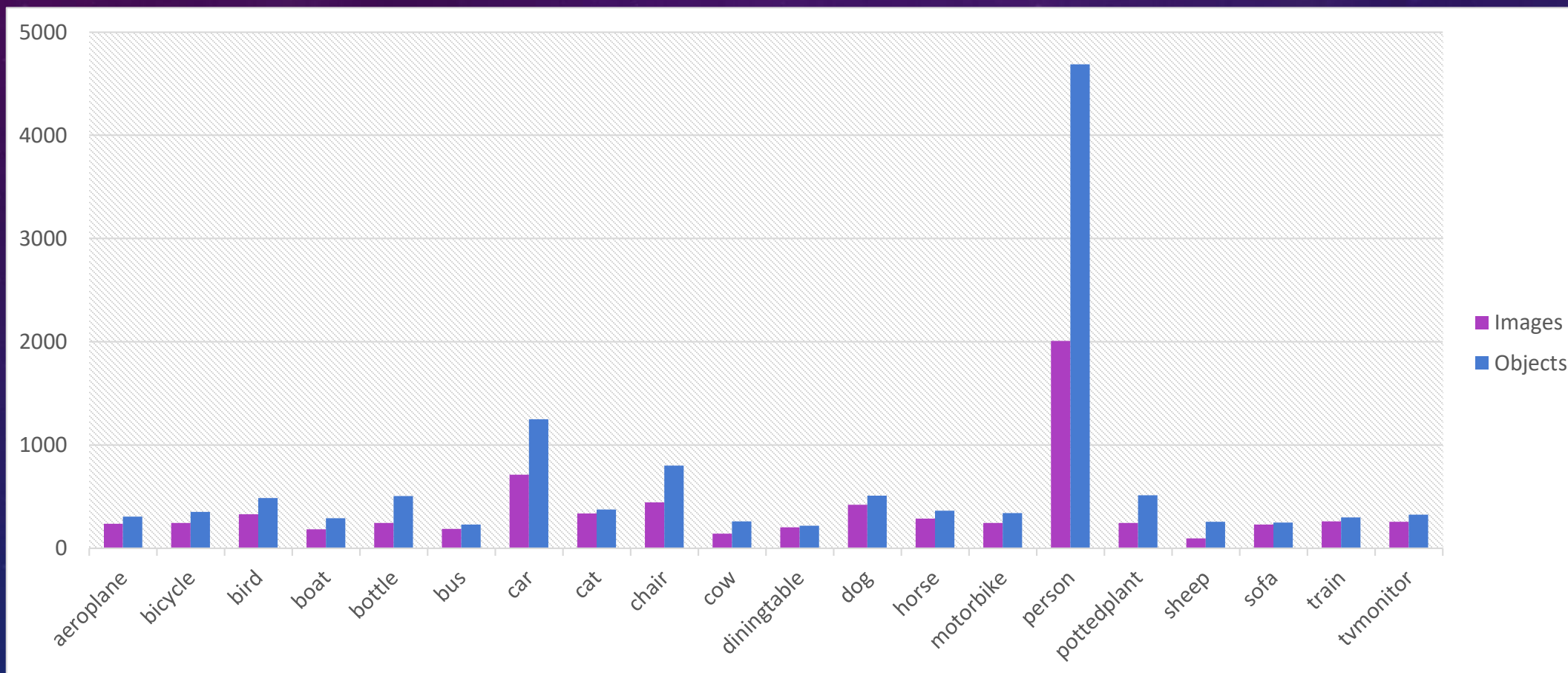
1. Person : person
2. Animal : bird, cat, cow, dog, horse, sheep
3. Vehicle : aeroplane, bicycle, boat, bus, car, motorbike, train
4. Indoor : bottle, chair, dining table, potted plant, sofa, tv/monitor

The number of images and objects in VOC 2007/2012:

	train		val		trainval		test	
	Images	Objects	Images	Objects	Images	Objects	Images	Objects
VOC 2007	2501	6301	2510	6307	5011	12608	4952	12032
VOC 2012	5717	13609	5823	13841	11540	27450	-	-
Total	8218	18810	8333	20148	16551	40058	4952	12032

# VOC (Visual Object Classes)

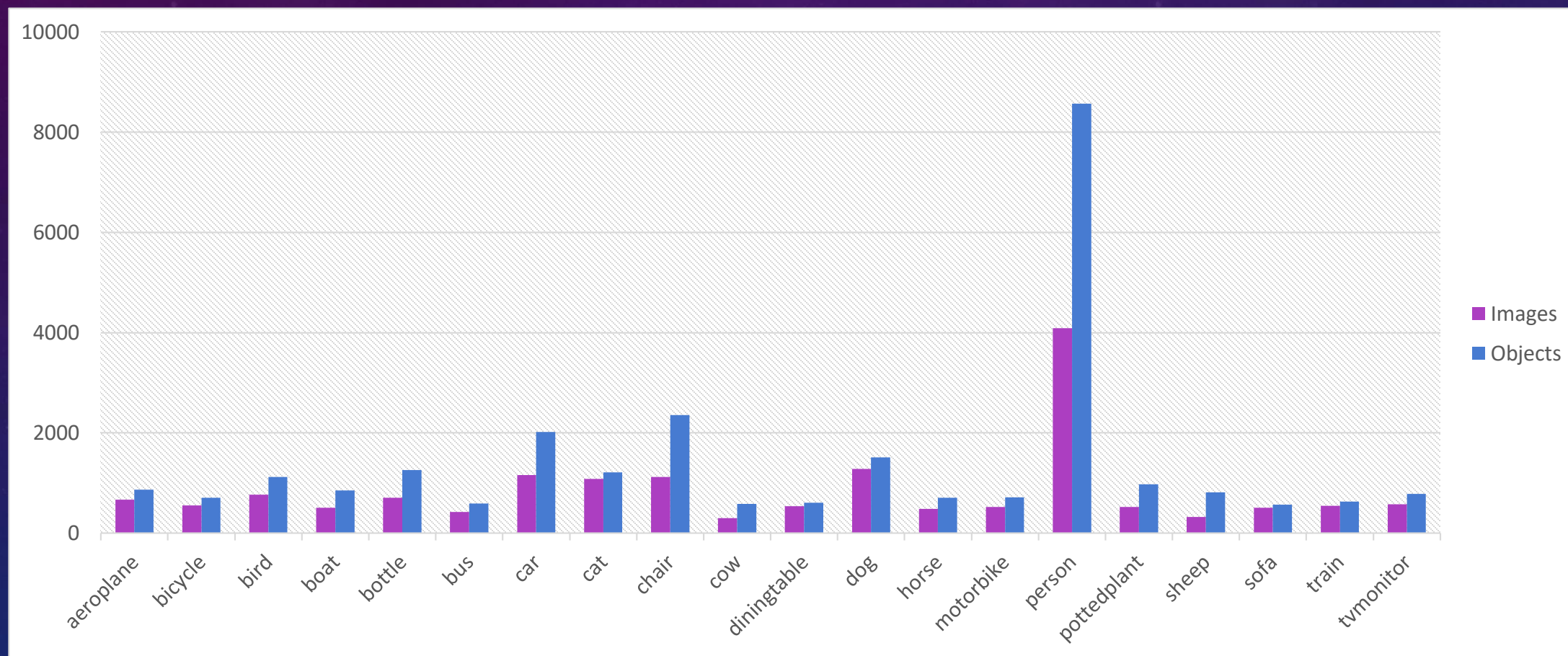
VOC 2007 (train + val)





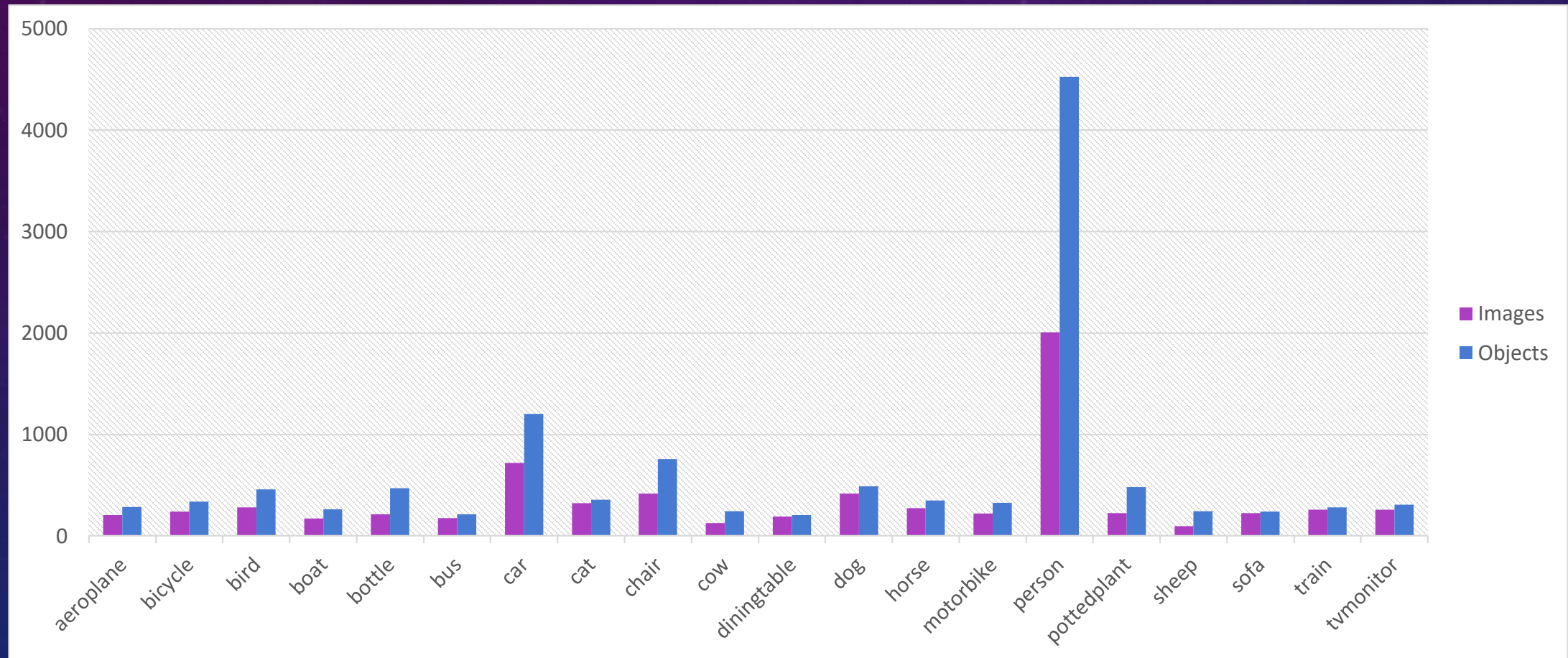
# VOC (Visual Object Classes)

VOC 2012 (train + val)



# VOC (Visual Object Classes)

VOC 2007 (test)





# COCO Dataset

- COCO dataset, meaning “Common Objects In Context”, literally implies that the images in the dataset are everyday objects captured from everyday scenes. COCO dataset is a set of challenging, high quality datasets for computer vision. And being one of the most popular image datasets out there, with applications like object detection, segmentation, and captioning.



## COCO dataset include:

- 330K images (>200K labeled)
- 1.5 million object instances
- 80 object categories

# Summary of Popular Dataset for Object Detection



Dataset Name	Total Images	Categories	Image per Category	Object per Image	Image Size	Started year
MNIST	60000	10	6000	1	28*28	1998
CIFAR-10	60000	10	6000	1	32*32	2009
PASCAL VOC	11540	20	303~4087	2.4	470*380	2005
ImageNet	14 millions+	21841	x	1.5		2009
MS COCO	328,000+	91	x	7.3	640*480	2014



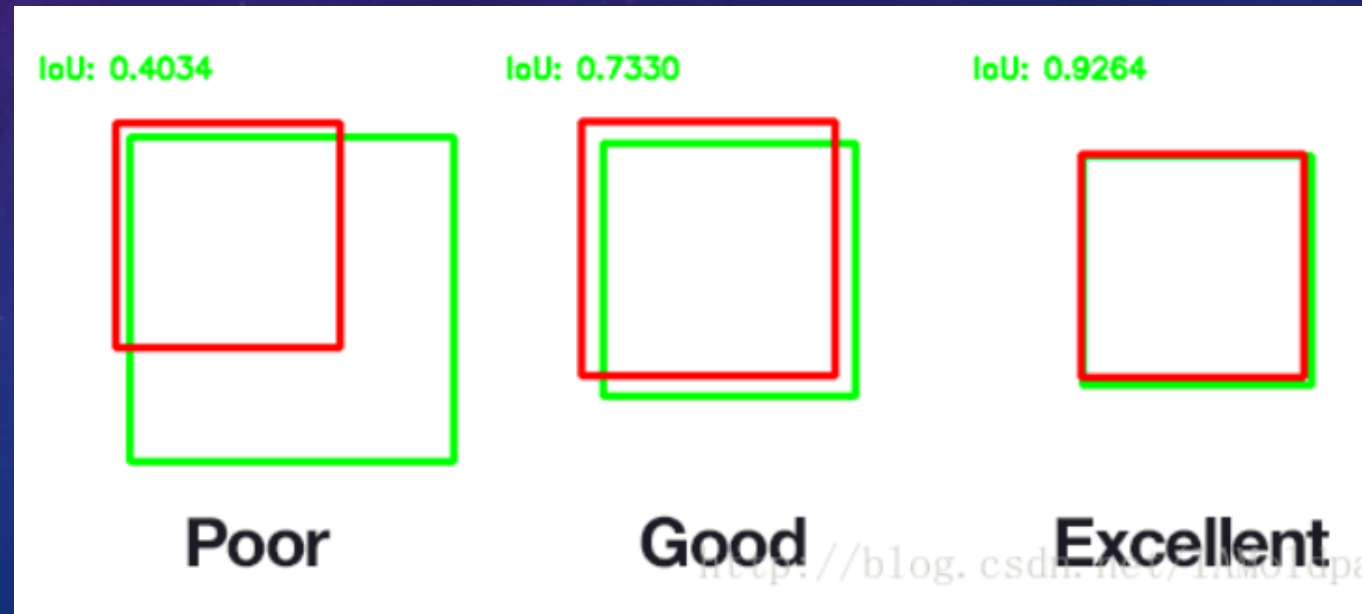
# Detection Evaluation Method

# IOU

- Formally we define confidence as  $Pr(\text{Object}) * IOU(\text{pred}, \text{truth})$ . If no object exists in that cell, the confidence score should be zero. Otherwise we want the confidence score to equal the intersection over union (IOU) between the predicted box and the ground truth.
- IOU

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


<http://blog.csdn.net/IAmoldpau>

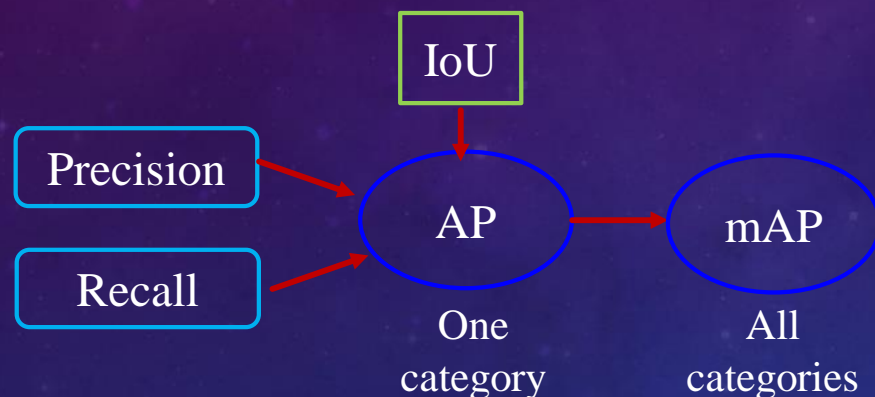




# mAP

- mAP: mean Average Precision over different categories

$$Precision = \frac{TP}{TP + FN} \quad Recall = \frac{TP}{TP + TN}$$



		True condition	
		Positive	Negative
Predicted	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

**AP@[0.5:.095]** corresponds to the average AP for IoU from 0.5 to 0.95 with a step size of 0.05.

For the COCO competition, AP is the average over 10 IoU levels on 80 categories (AP@[0.50:.005:0.95]: start from 0.5 to 0.95 with a step size of 0.05).

# FPS(Frame per second)

- FPS (Frame Per Second) defines how fast your object detection model process your video and generate the desired output.
- For example, at 30 fps, 30 distinct images would appear in succession within one second.



# Face Detection

# What is face detection?

- Face detection is a computer vision problem that involves finding faces in photos.
- It is a trivial problem for humans to solve and has been solved reasonably well by classical feature-based techniques, such as the cascade classifier.
- More recently deep learning methods have achieved state-of-the-art results on standard benchmark face detection datasets.
- One example is the Multi-task Cascade Convolutional Neural Network, or MTCNN for short.

From <https://machinelearningmastery.com/how-to-perform-face-detection-with-classical-and-deep-learning-methods-in-python-with-keras/>

# Face Detection Overview

- Face Detection
- Face Detection With OpenCV
- Face Detection With Deep Learning



# Face Detection

- Face detection is a problem in computer vision of locating and localizing one or more faces in a photograph.
- Locating a face in a photograph refers to finding the coordinate of the face in the image, whereas localization refers to demarcating the extent of the face, often via a bounding box around the face.

# Face Detection with OpenCV

- Feature-based face detection algorithms are fast and effective and have been used successfully for decades.
- A modern implementation of the Classifier Cascade face detection algorithm is provided in the OpenCV library.
- This is a C++ computer vision library that provides a python interface.
- The benefit of this implementation is that it provides pre-trained face detection models, and provides an interface to train a model on your own dataset.

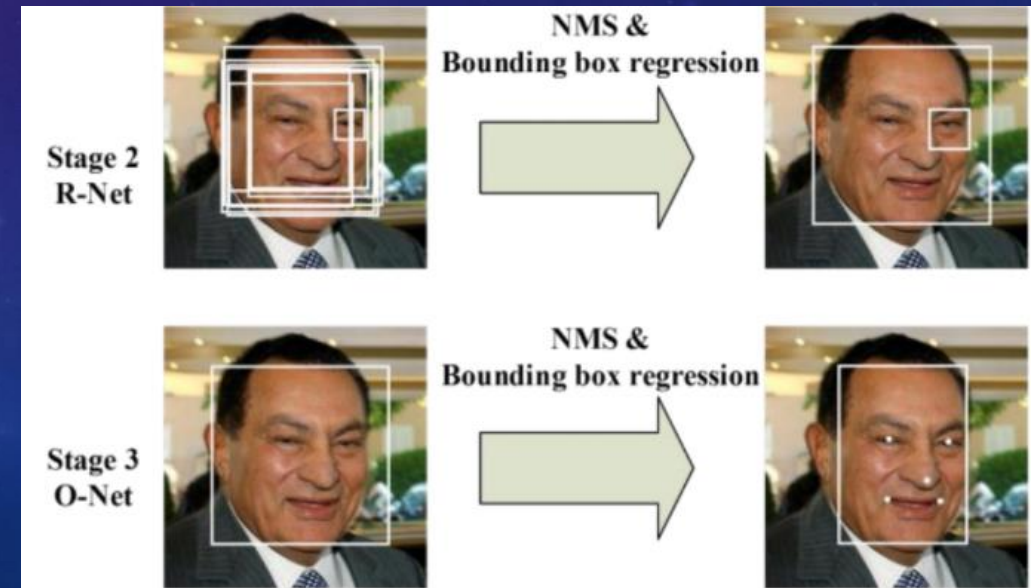
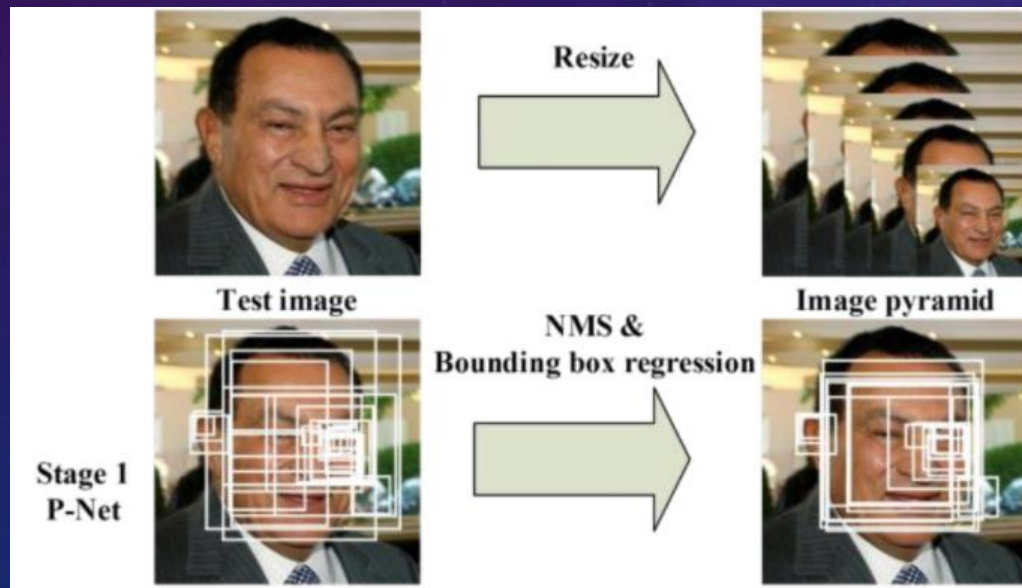
# Face Detection with Deep Learning - MTCNN

- A number of deep learning methods have been developed and demonstrated for face detection.
- One of the more popular approaches is called the “*Multi-Task Cascaded Convolutional Neural Network*,” or MTCNN for short, described by Kaipeng Zhang, et al. , 2016 IEEE Signal Processing Letters, titled “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks.”
- The MTCNN is popular because it achieved then state-of-the-art results on a range of benchmark datasets, and because it is capable of also recognizing other facial features such as eyes and mouth, called landmark detection.



# Face Detection with Deep Learning - MTCNN

- The network uses a cascade structure with three networks; first the image is rescaled to a range of different sizes (called an image pyramid), then the first model (Proposal Network or P-Net) proposes candidate facial regions, the second model (Refine Network or R-Net) filters the bounding boxes, and the third model (Output Network or O-Net) proposes facial landmarks.
- The image below taken from the paper provides a helpful summary of the three stages from top-to-bottom and the output of each stage left-to-right.

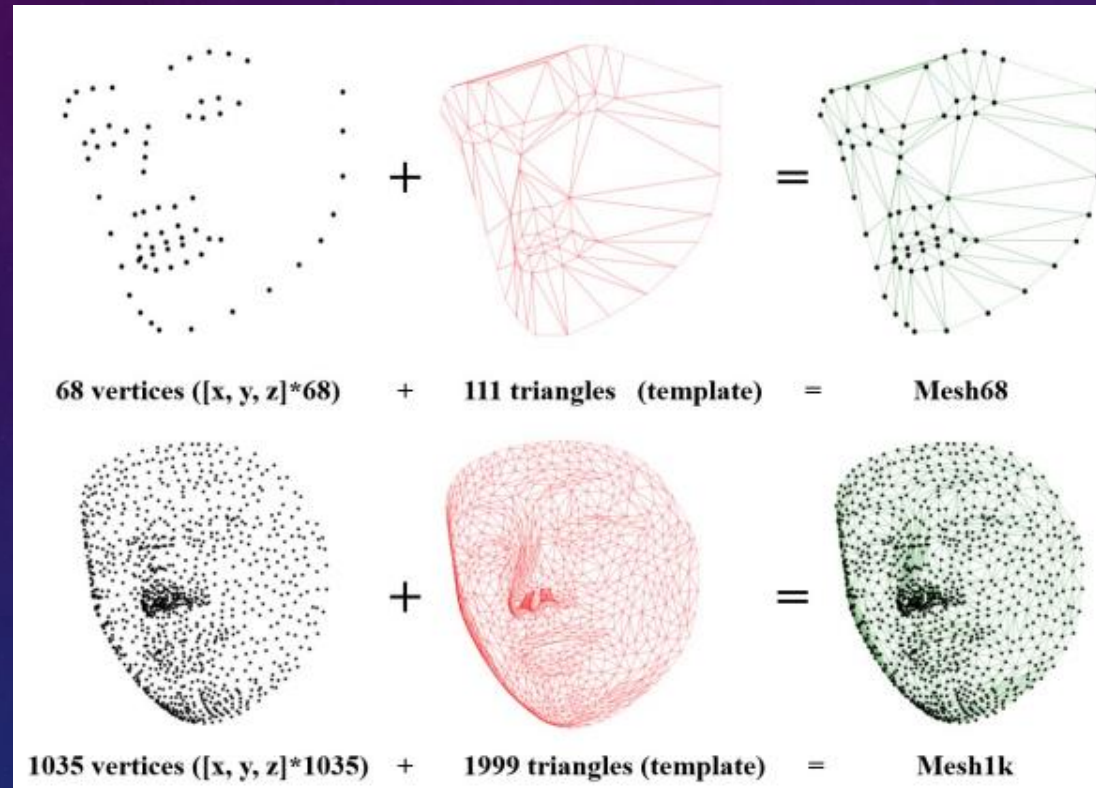


# Face Detection with Deep Learning - RetinaFace

- RetinaFace performs three different face localization tasks together, that are face detection, 2D face alignment and 3D face reconstruction based on a single shot framework.
- All the three targets are solved keeping in mind only one common target that all the points regressed for the above three tasks should lie on the image plane.

# Face Detection with Deep Learning - RetinaFace

## Approach - 3D Face Reconstruction



- For creating a 3D face from the 2D image, using a predefined triangular face with N vertices as shown in the above figure.



# Face Detection with Deep Learning - RetinaFace

## Approach - 3D Face Reconstruction

- The vertices shares the same semantic meaning across different faces and with the fixed triangular topology each face pixel can be indexed by barycentric coordinates and the triangle index making pixel wise correspondence with the 3D face.

# Face Detection with Deep Learning - RetinaFace

## Approach - 3D Face Reconstruction

- For regressing the 3D vertices on the 2D image plane, they are using 2 loss functions:
- Here, N is the total vertices i.e. 1103(68+1035) and V is predicted point and V\* is ground-truth point.

$$\mathcal{L}_{vert} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{V}_i(x, y, z) - \mathbf{V}_i^*(x, y, z)\|_1,$$

- It is the edge length loss, as it is a triangular topology. Here, M is the number of triangles i.e. 2110(111+1999) and E is predicted edge length and E\* is ground truth edge length.

$$\mathcal{L}_{edge} = \frac{1}{3M} \sum_{i=1}^M \|\mathbf{E}_i - \mathbf{E}_i^*\|_1,$$

- So the total loss for regressing 3D points becomes:

$$\mathcal{L}_{mesh} = \mathcal{L}_{vert} + \lambda_0 \mathcal{L}_{edge},$$

# Face Detection with Deep Learning - RetinaFace

## Approach – Multi-Level Face Localization

- The complete loss function for an anchor  $i$  becomes :

$$\mathcal{L} = \mathcal{L}_{cls}(p_i, p_i^*) + \lambda_1 p_i^* \mathcal{L}_{box}(t_i, t_i^*) \\ + \lambda_2 p_i^* \mathcal{L}_{pts}(l_i, l_i^*) + \lambda_3 p_i^* \mathcal{L}_{mesh}(v_i, v_i^*).$$

- The loss function has 4 parts:
  - a) Softmax loss for binary classes (face / not face), where,  $p$  is the predicted probability that anchor  $i$  is face and  $p^*$  is ground truth.
  - b) Regression loss of bounding box.
  - c) Regression loss of five landmarks
  - d) Regression loss of 3D points as discussed above.



# Face Detection with Deep Learning - RetinaFace

## Approach – Multi-Level Face Localization

- All the coordinates are normalized as:

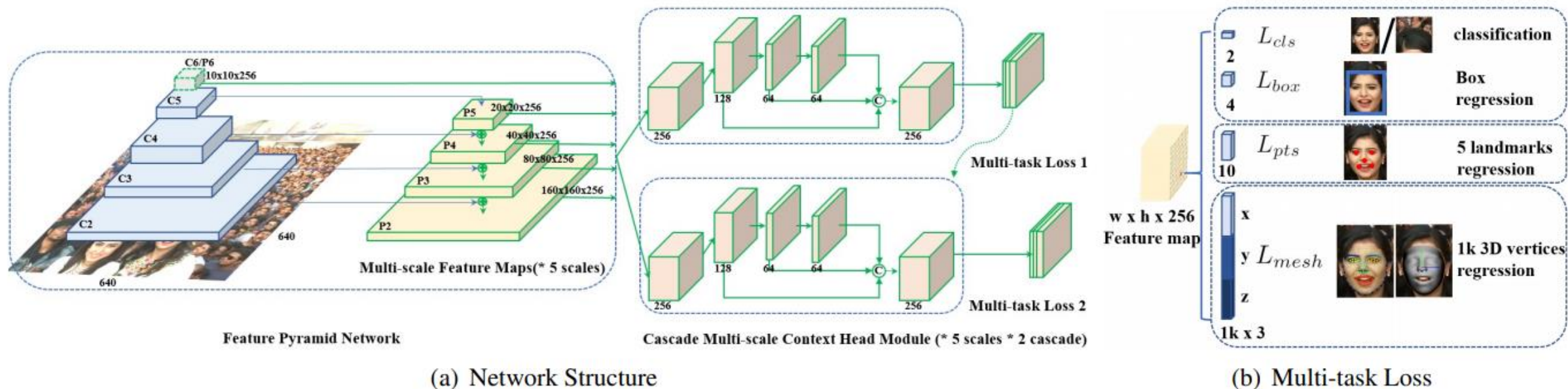
$$\begin{aligned} & (x_j^* - x_{center}^a) / s^a, \\ & (y_j^* - y_{center}^a) / s^a, \\ & (z_j^* - z_{nose-tip}^*) / s^a, \end{aligned}$$

# Face Detection with Deep Learning - RetinaFace

## Approach – Single-shot Multi-level Face Localization

The model consists of four main components:

- a) Feature Pyramid Network
- b) Context Head Module
- c) Cascade Multi Task Loss
- d) Matching Strategy



# Face Detection with Deep Learning - RetinaFace

Approach – Single-shot Multi-level Face Localization

## Feature Pyramid Network

- It takes the input image and outputs five feature maps of different scales.
- First four feature map in above figure is calculated using ResNet which was pre-trained on imagenet-11k dataset.
- The top most feature map was by the convolution of 3x3 with stride 2 on C5.

# Face Detection with Deep Learning - RetinaFace

Approach – Single-shot Multi-level Face Localization

## Context Module

- To strengthen the context modelling capacity deformation convolutional network(DCN) is used in this module over the feature maps other than normal 3x3 convolution.



# Face Detection with Deep Learning - RetinaFace

Approach – Single-shot Multi-level Face Localization

## Cascade Multi-task Loss

- To improve face localization cascade regression is used along with multi-task loss as described above.
- The first context module predicts the bounding box using the regular anchors and then subsequent modules predicts more accurate bounding box using the regressed anchors.

# Face Detection with Deep Learning - RetinaFace

Approach – Single-shot Multi-level Face Localization

## Matching Strategy

- From the first context head module, anchors are matched to ground truth boxes if their IOU is greater than 0.7 and to background if it is less than 0.3 and for the second context head module, anchors are matched to ground truth boxes if their IOU is greater than 0.5 and to background if it is less than 0.4.
- Positive and negative training examples are balanced using OHEM.