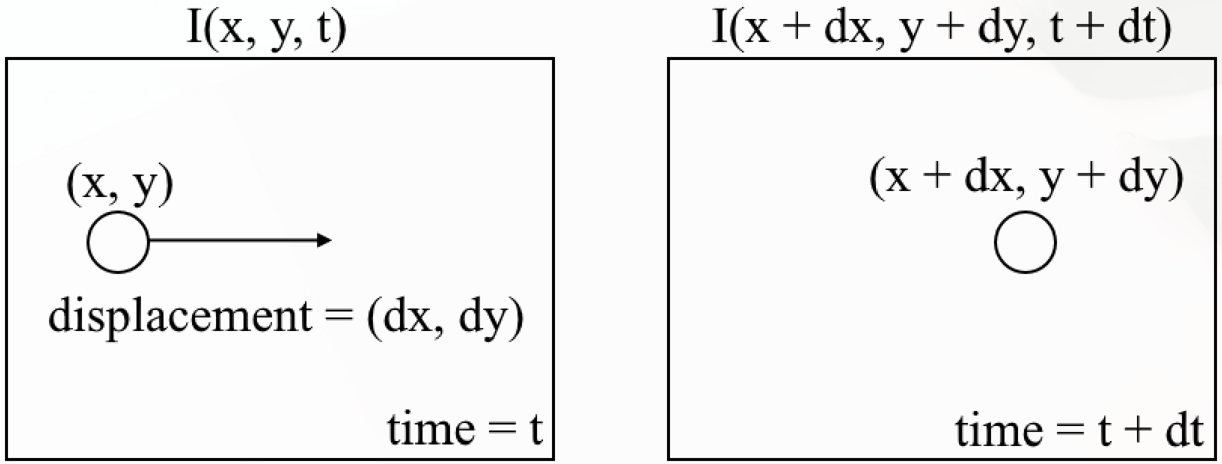
22AIE313 Computer Vision & Image Understanding (2-1-3-4)

Dr. Aiswarya S Kumar
Dept. of Computer Science & Engineering (AI)
Amrita School of Computing

Optical Flow

Optical flow is the motion of objects between consecutive frames of sequence, caused by the relative movement between the object and camera.



We can express the image intensity (I) as a function of space (x,y) and time (t).

Optical Flow

We assume that pixel intensities of an object are constant between consecutive frames.

Optical flow constraint equation:

$$I_x v_x + I_y v_y + I_t = 0$$

where I_x, I_y, I_t are the partial derivative of the image intensity respect to x, y and t, while v_x, v_y are the x and y components of the optical flow.

To estimate v_x and v_y components, we can use Lucas Kanade Method.

Optical Flow

Lucas Kanade Method:

- The Lucas-Kanade optical flow algorithm is a simple technique which can provide an estimate of the movement of interesting features in successive images of a scene.
- The Lucas-Kanade method assumes that the optical flow is constant within a small window of size n x n pixels.
- This method is applicable when the color of an object does not change significantly and significantly in the previous two frames.

Optical Flow

For every point in window, W, linear equations can be formed and they can be summarized as the matrix equality.

$$S\begin{pmatrix} u \\ v \end{pmatrix} = \vec{t}$$

If you consider a 3 × 3 neighborhood around the pixel (x, y), then S will be a 9 × 2 matrix and t is a vector containing the 9 terms.

The above equation can be solved using the Least Squares solution is found by multiplying the equation by S^T.

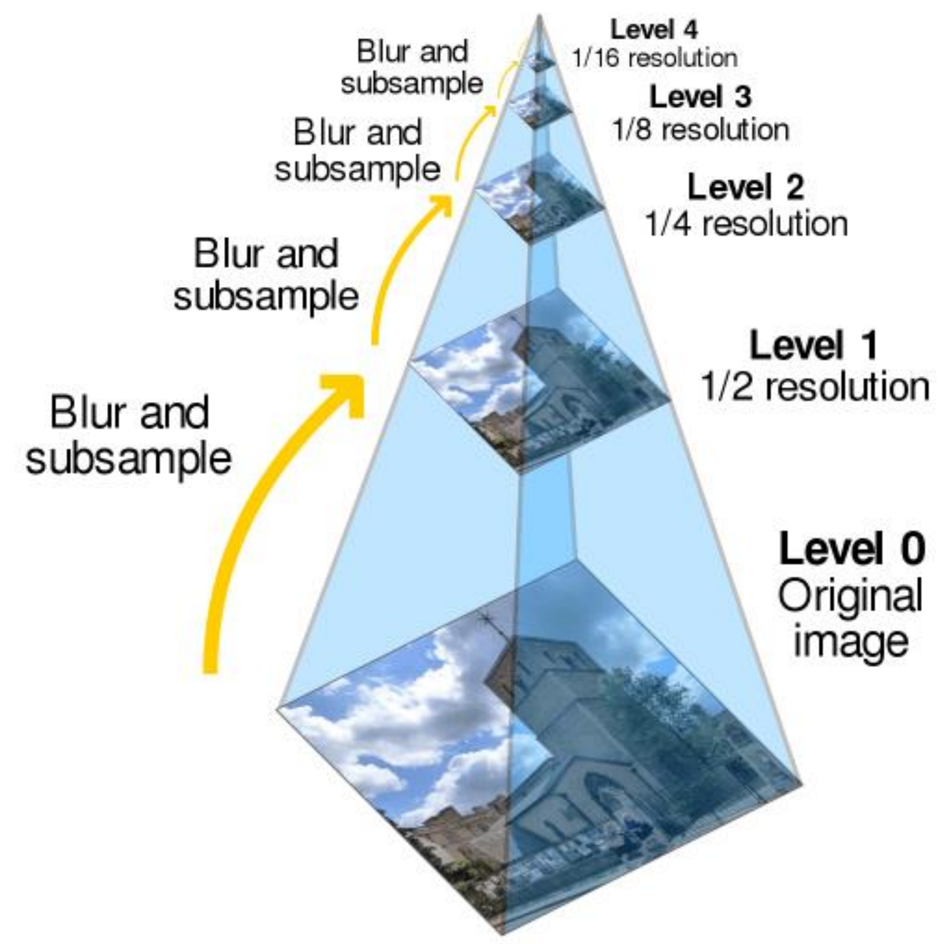
$$S^T S\begin{pmatrix} u \\ v \end{pmatrix} = S^T \vec{t}$$

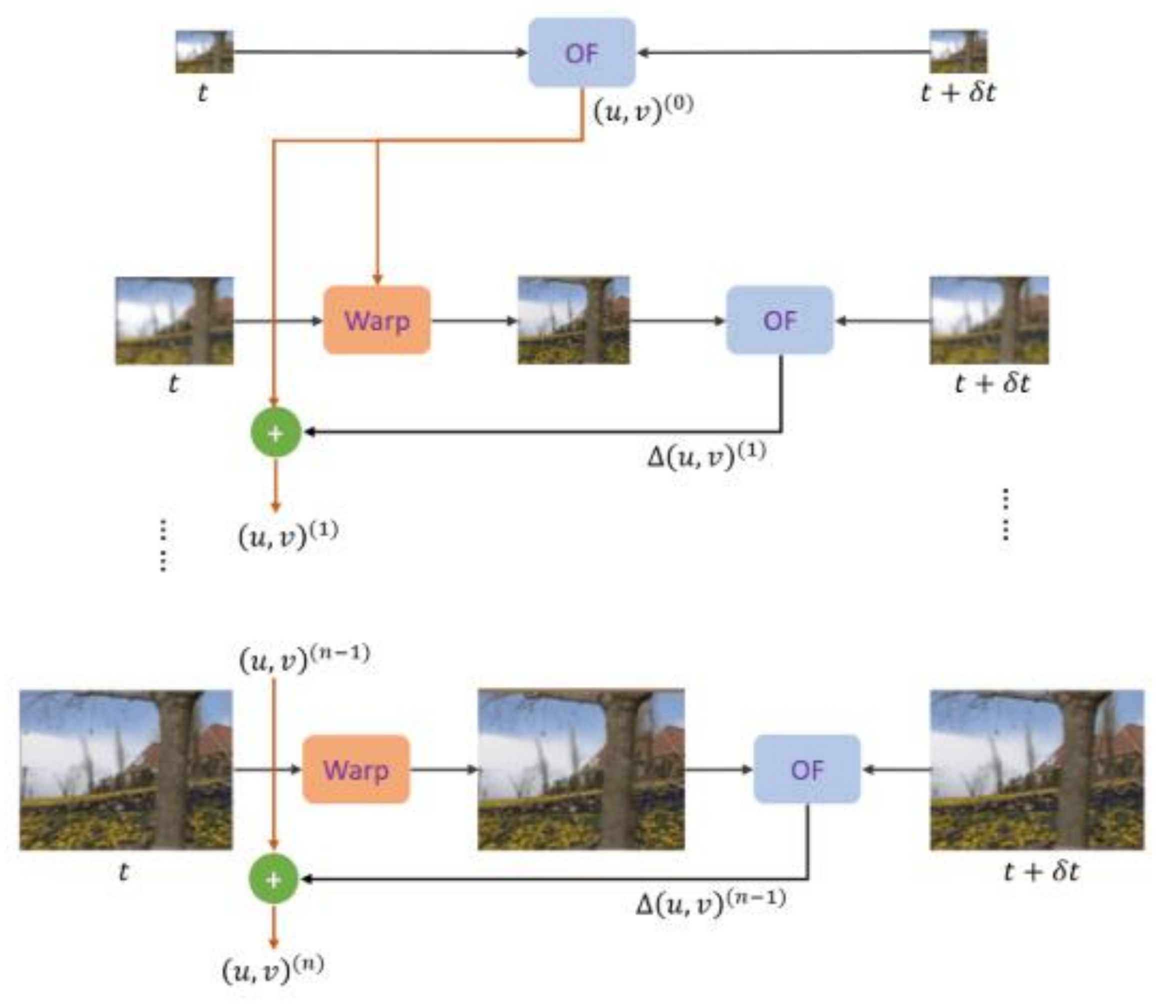
and inverting $S^T S$, so that

$$\begin{pmatrix} u \\ v \end{pmatrix} = (S^T S)^{-1} S^T \vec{t}$$

Resolution pyramids

- Also known as image pyramids.
- A multi-scale representation of an image where each level of the pyramid represents a different resolution of the image.
- Each level of the pyramid is created by downsampling the previous level (typically by a factor of 2, resulting in a pyramid structure where each level is half the size of the one below it).




OF
t
$t + \delta t$
$(u, v)^{(0)}$
Warp
OF
t
$t + \delta t$
+
$\Delta(u, v)^{(1)}$
$(u, v)^{(1)}$
$(u, v)^{(n-1)}$
Warp
OF
t
$t + \delta t$
+
$\Delta(u, v)^{(n-1)}$
$(u, v)^{(n)}$

Coarse-To-Fine Optical Flow Estimation Algorithm

1. Compute lower-resolution versions of the two consecutive images (acquired at time t and $t + \delta t$) by taking each 2×2 window in one of the images, finding the average of those values, and using that value in the new low-resolution image
2. Start with the lowest resolution images and apply the optical flow algorithm, such as Lucas-Kanade's algorithm, to compute the optical flow
3. Use the optical flow resulting from the previous step to warp the image at t in the next higher resolution
4. Compute the optical flow between the warped image and image at $t + \delta t$. Finally, add the result to the previous optical flow estimate
5. Keep repeating steps 3 and 4 until arriving at the highest-resolution images, which gives the final flow

Tasks in CV

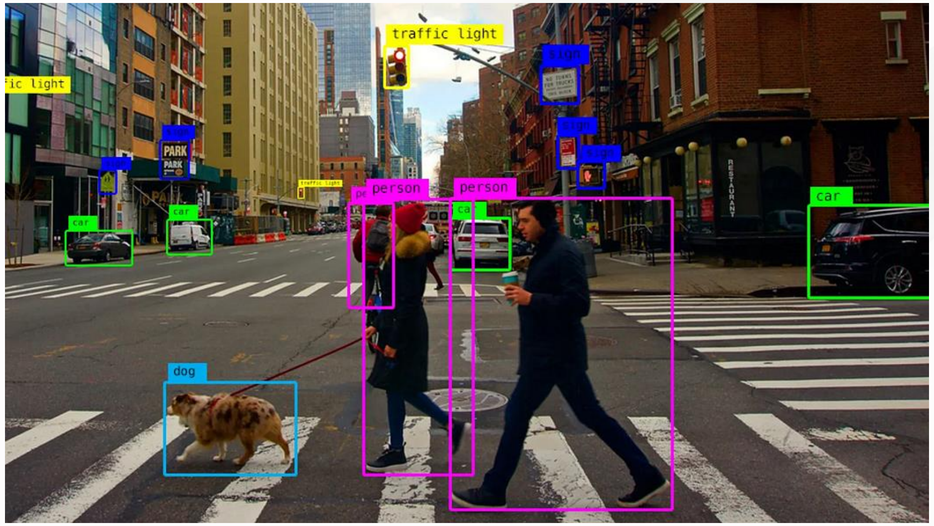
Recommended read : https://viso.ai/computer-vision/image-recognition/

Object Detection

Object detection locates and classifies multiple objects of interest within an image.
Where + What -> bounding boxes + class labels

Example: Detecting all people, cars, and dogs in a street image.

Popular models:
- YOLO (You Only Look Once)
- Faster R-CNN
- SSD (Single Shot Detector)
- DETR (Detection Transformer)

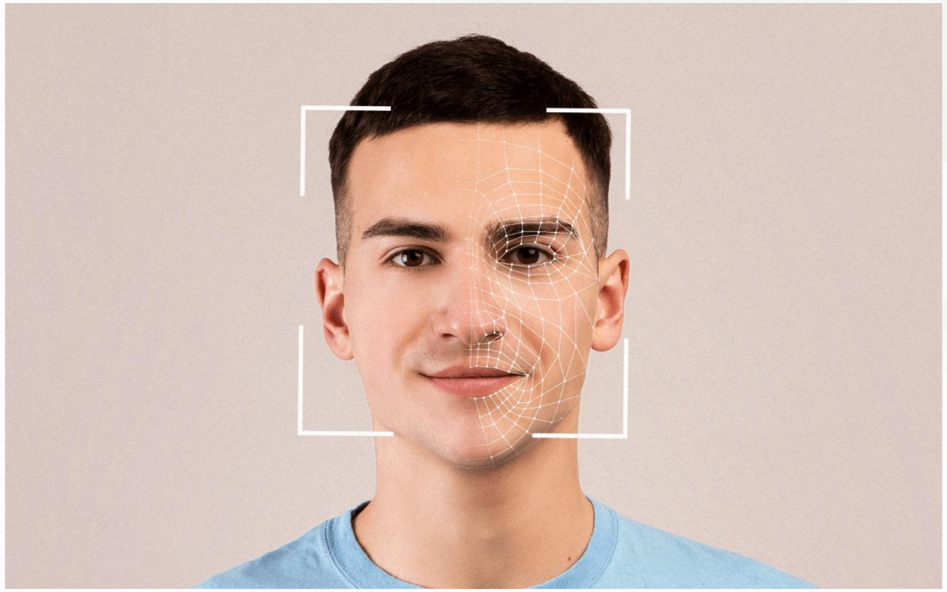


Face Detection

Locating human faces in images using bounding boxes.
Where -> bounding boxes

Popular models:

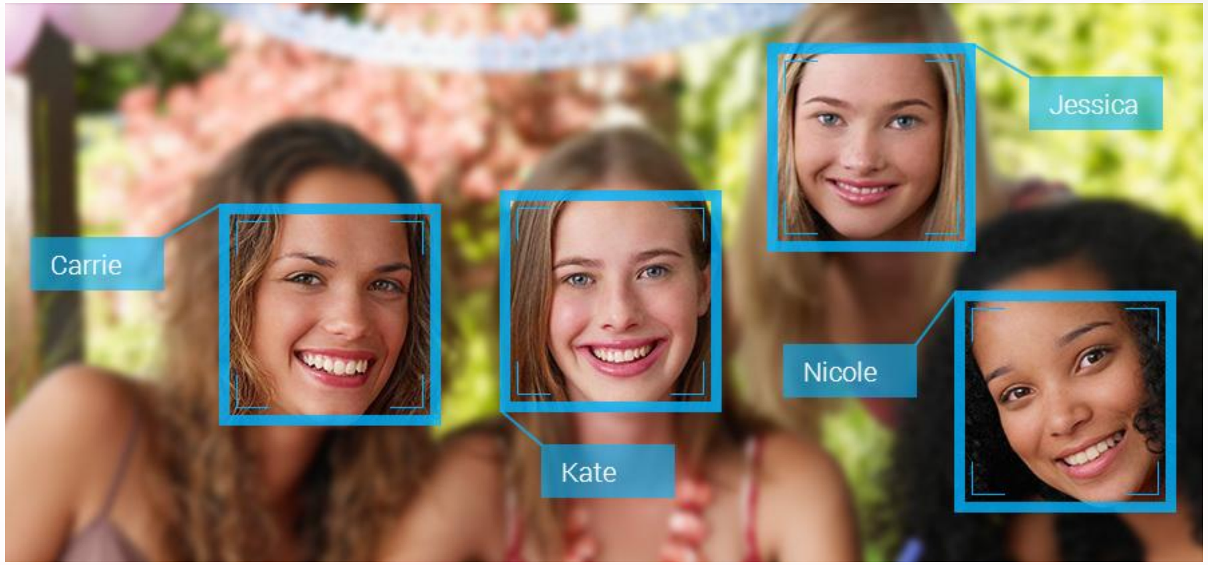
- Haar cascades
- MTCNN
- RetinaFace



Face Recognition

Identifying individuals in images by analyzing facial features.
Where + Who -> bounding boxes + class labels

Popular models:
- FaceNet
- DeepFace
- ArcFace



Instance (Level) Recognition

Identifying a specific object instance rather than just a category.

Example: Recognizing a specific coffee mug from a collection of mugs.

Popular models:

- SIFT + descriptor matching
- NetVLAD
- DELF

Photo by Sylwia Bartyzel on Unsplash

Basic Recognition	Building, Architecture, Monument, Dome
Instance-level Recognition	Taj Mahal, India

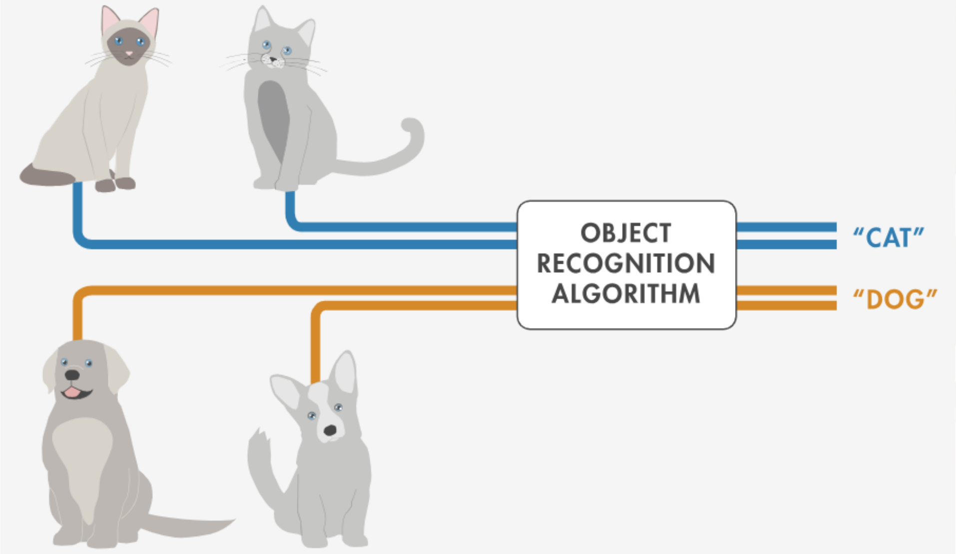
Category Recognition

Identifying what type or category an object belongs to.

Example: Classifying an animal as a "cat", "dog", or "horse“.

Popular models:

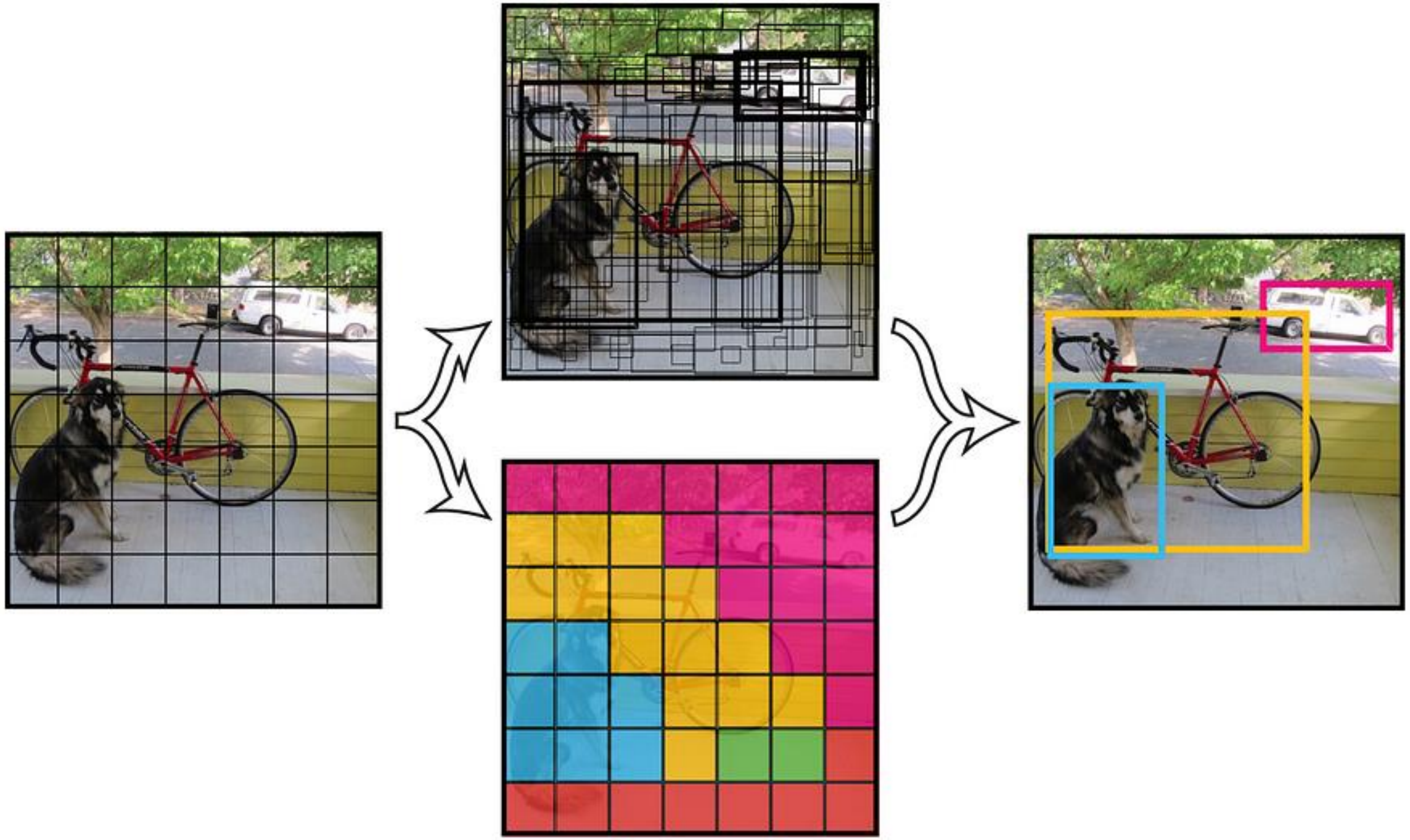
- AlexNet
- VGG
- ResNet



YOLO (You Only Look Once)

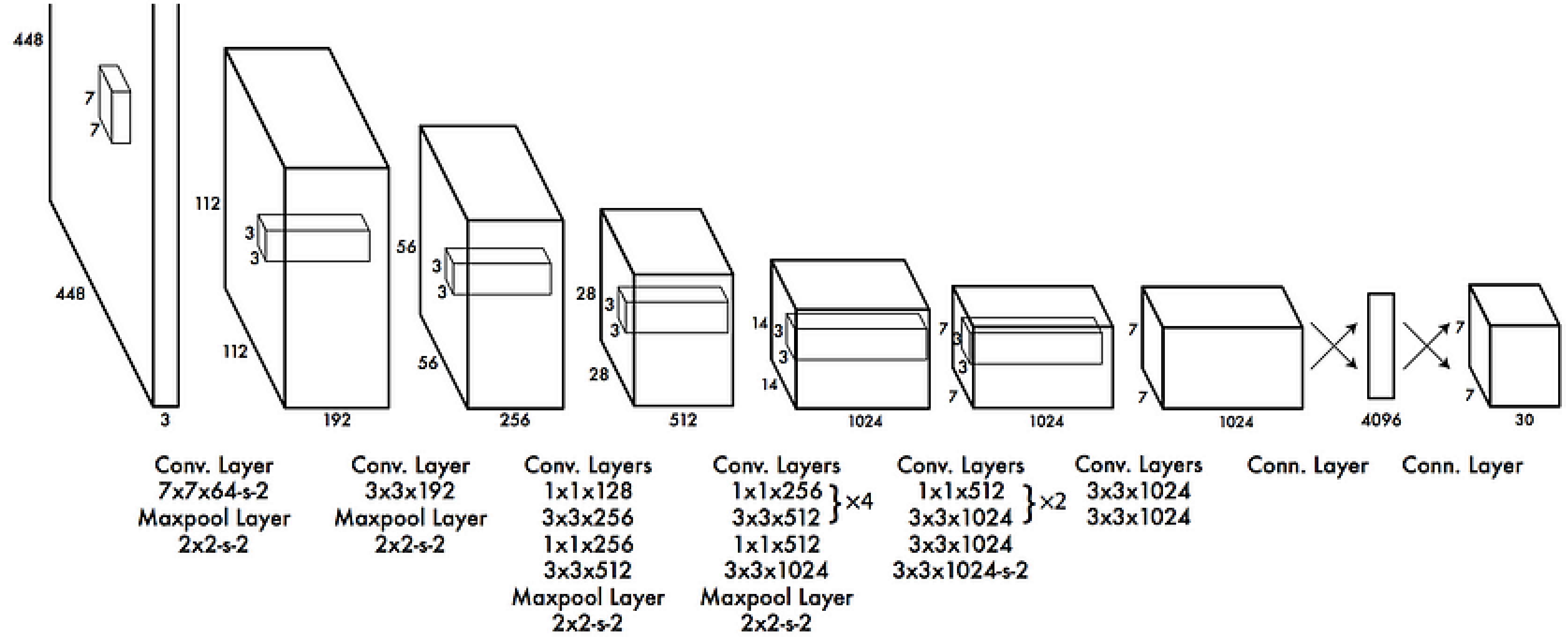
- In 2015, Joseph Redmon came up with an architecture called YOLO.
- YOLO uses a single neural network to predict bounding boxes and associate class probabilities.
- The image is divided into grids and each gird produces bounding boxes and their confidence scores.
- Each bounding box consists of 5 predictions: x, y, w, h, and confidence.

Original source: https://arxiv.org/pdf/1506.02640



Simple explanation on YOLOv1: https://www.youtube.com/watch?v=NAwleAip-9Q

YOLO v1 Architecture



Viola Jones for Face Detection

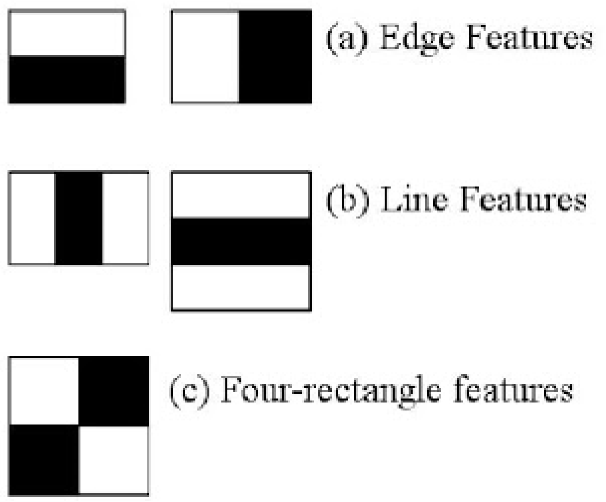
- Developed by Paul Viola and Michael Jones.
- Relatively slow to train, but can quickly detect objects accurately.
- Works well to detect human faces in real time.

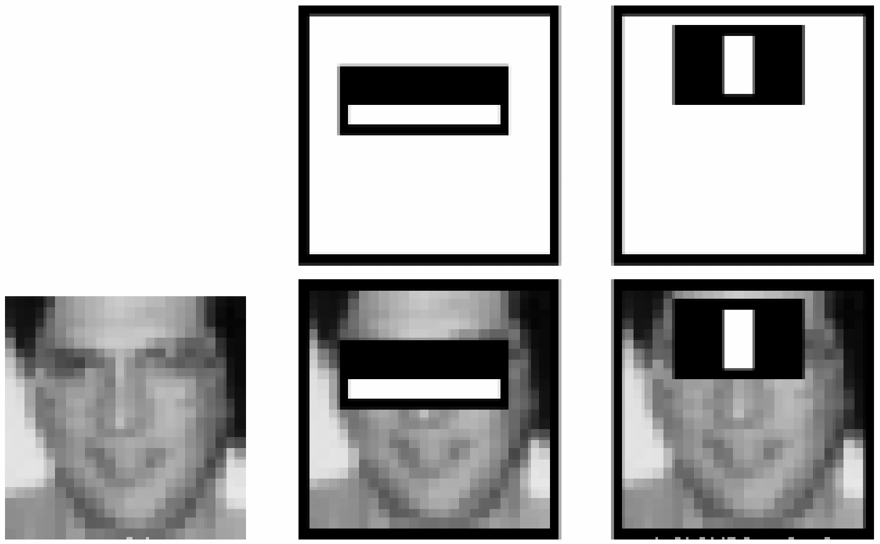
Steps in Viola Jones

1. Selecting Haar-like features
2. Creating an integral image
3. Running AdaBoost training
4. Creating classifier cascades

Viola Jones for Face Detection

1. Selecting Haar features





Viola Jones for Face Detection

2. Creating an integral image

- An integral image gives a fast and simple way to calculate the value of any haar-like feature.
- The value for location (x, y) on the integral image is the sum of the pixels above and to the left of the (x, y) on the original image plus itself.
- After constructing the integral image, it is used to efficiently calculate the sum of dark/light rectangular regions.

Original

5	**2**	**3**	4	1
1	**5**	**4**	2	3
2	2	1	3	4
3	5	6	4	5
4	1	3	2	6

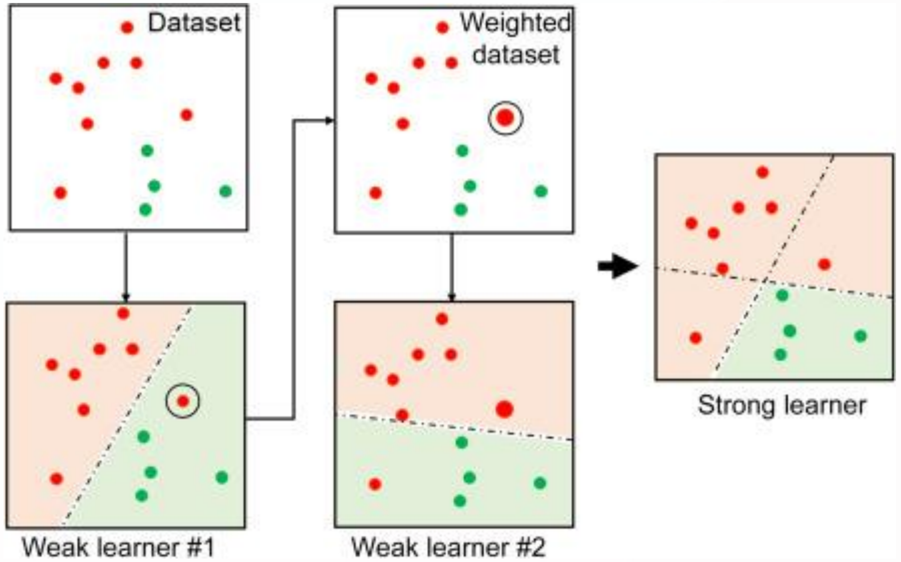
Integral

5	7	10	14	15
6	13	**20**	26	30
8	17	25	34	42
11	25	39	52	65
15	30	47	62	81

$$5 + 2 + 3 + 1 + 5 + 4 = 20$$

Viola Jones for Face Detection

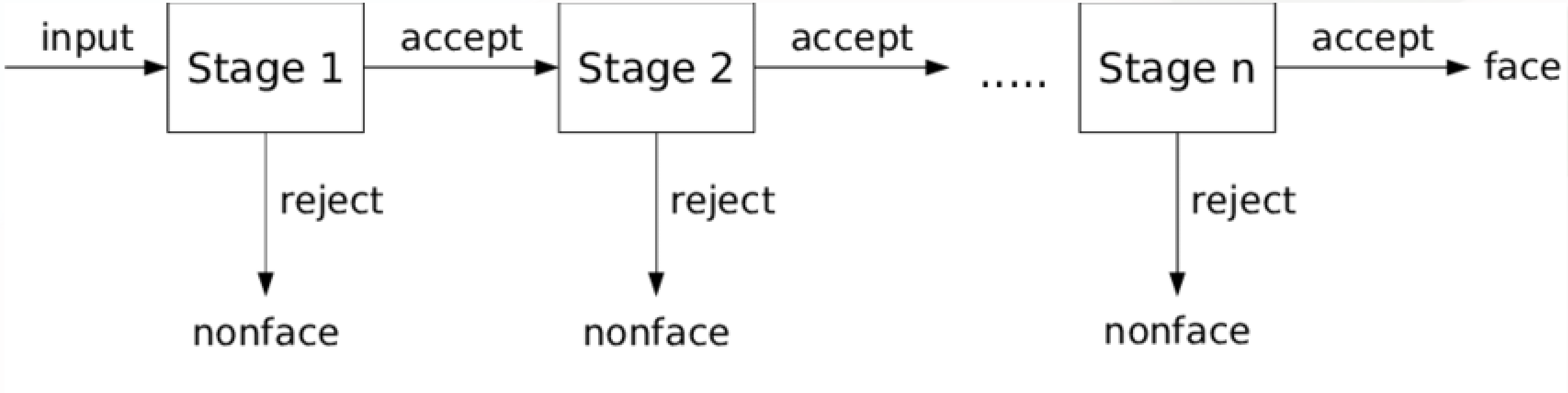
3. Running AdaBoost – for feature selection



- Selecting a subset of prominent features from the huge set of features.
- The algorithm sets a minimum threshold for determining a useful feature.
- ‘N’ no. of weak classifiers together make a strong classifier.

Viola Jones for Face Detection

4. Cascading



Thank you