MLAI 507
COMPUTER VISION & SPEECH PROCESSING

MLAI – 2023/2024

Dr. Zein Al Abidin IBRAHIM

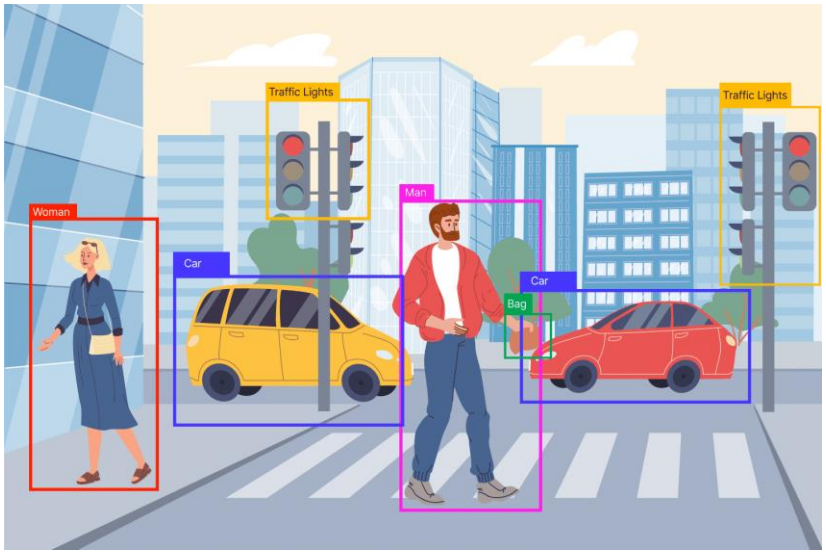
zein.ibrahim@ul.edu.lb

OBJECT DETECTION

Computer vision

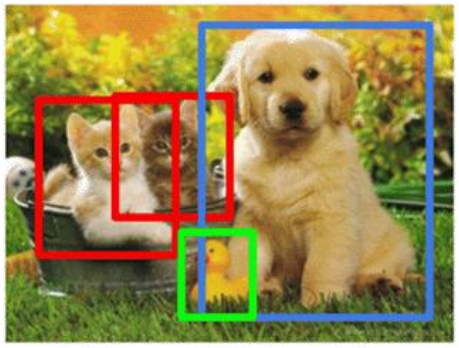
WHAT IS OBJECT DETECTION ?

- Object detection in computer vision (CV) → process of identifying and locating objects within an image or a video frame.
- Image classification → model that predicts a single class label for the entire image
- Object detection algorithms → provides information about the **presence**, **location**, and **class of multiple objects** within the image.



Classification

CAT



Object Detection

CAT, DOG, DUCK

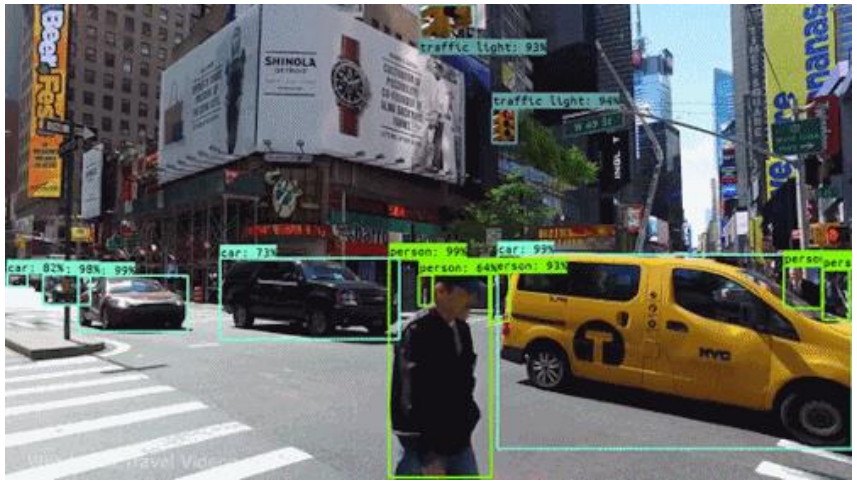
MAIN PROCESS FOR OBJECT DETECTION

1. **Localization**: first localize objects by identifying their bounding boxes in an image.
   - A bounding box is a rectangular box that encloses an object of interest.
   - A bounding box indicates where the object is located in the image.
2. **Classification**: after localization → classification of each localized object into predefined categories or classes.
   - Classes can represent various objects, such as cars, pedestrians, animals, etc.
3. **Detection**: includes the bounding boxes and their corresponding class labels for all detected objects in the image.

APPLICATIONS

- **Surveillance**: Monitoring and detecting objects in security cameras or surveillance footage.
- **Autonomous vehicles**: Identifying pedestrians, vehicles, traffic signs, and other objects in the environment to enable safe navigation.
- **Medical imaging**: Detecting and localizing abnormalities or specific anatomical structures in medical images such as X-rays, MRIs, and CT scans.
- **Retail**: Counting and tracking products on store shelves for inventory management and stock monitoring.
- **Augmented reality**: Placing virtual objects accurately in the real-world environment by detecting and understanding the scene.
- **Object tracking**: Continuously locating and following objects across multiple frames in video sequences.
- Many other domains like agriculture, military, …

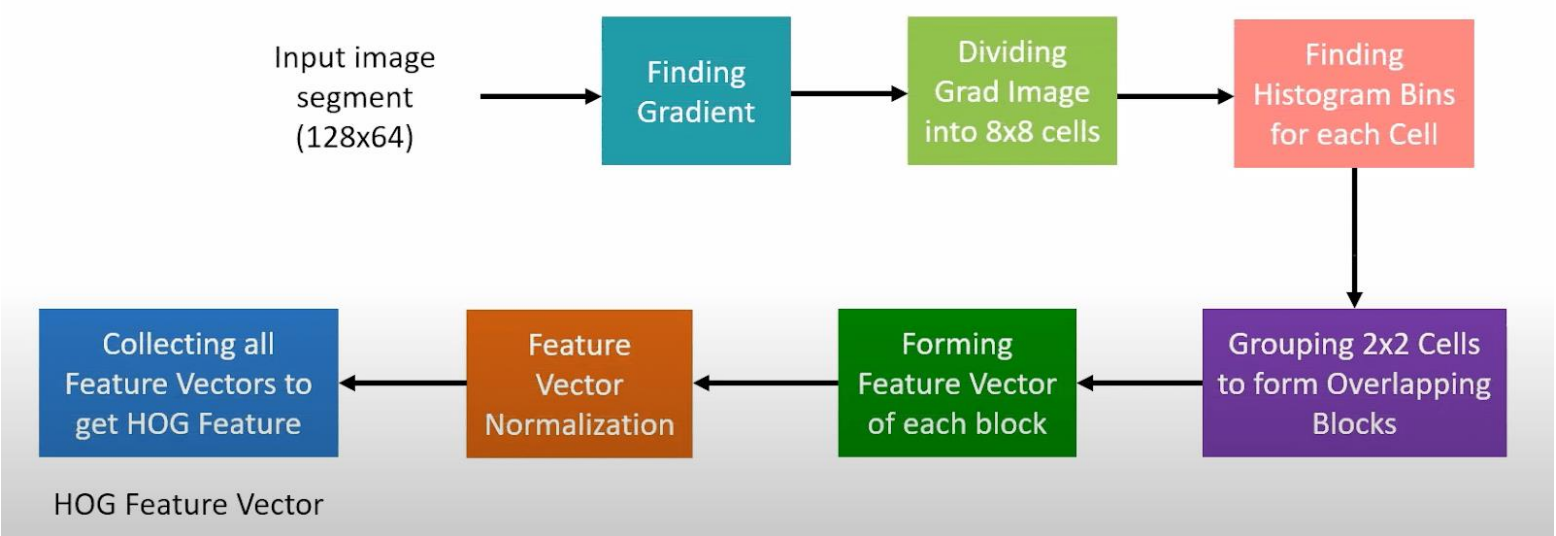
EXAMPLES


traffic light: 93%
traffic light: 94%
car: 82%
car: 73%
person: 99%
person: 64%
car: 99%
person: 93%
SHINOLA
NYC


Imagery Courtesy of Maxar

EARLY METHODS OF OBJECT DETECTION

1. **Histogram of Oriented Gradients (HOG)**:
   - Introduced in 1986 but started to gain popularity in 2005.
   - HOG is a feature descriptor technique used to detect objects in image.
   - It works by computing the distribution of gradient orientations in localized portions of an image.
   - HOG descriptors capture the shape and structure of objects by analyzing the intensity gradients and their directions.
   - Support Vector Machines (SVM) were often used with HOG features for object detection.



HOG

- The x and y derivatives of an image (Gradients) are useful because the magnitude of gradients is large around edges and corners due to abrupt change in intensity and we know that edges and corners pack in a lot more information about object shape than flat regions.

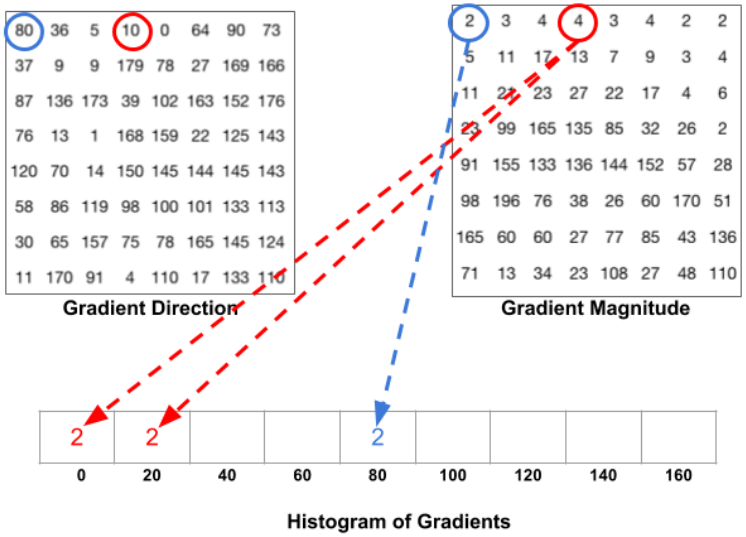
$$G_x = 100 - 50 = 50$$

$$G_y = 120 - 70 = 50$$

$$\theta = arctan(\frac{G_y}{G_x}) = 45^o$$

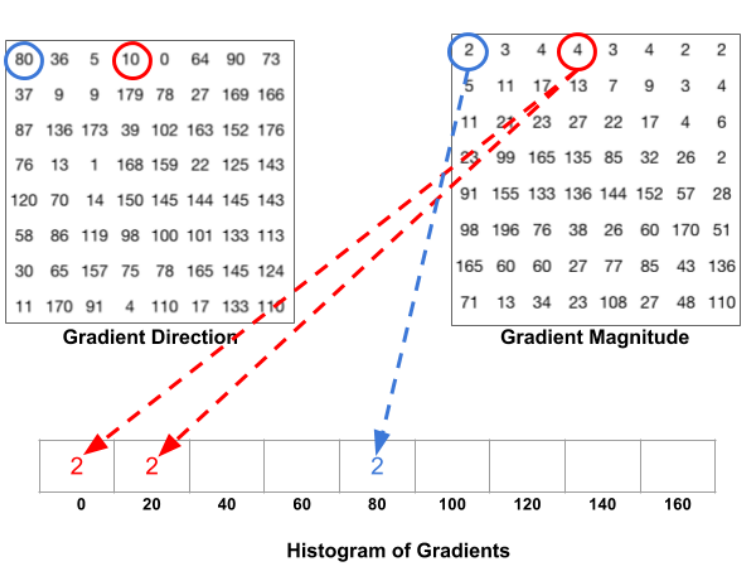
$$G = \sqrt{(G_x)^2 + (G_y)^2} = 70.7$$

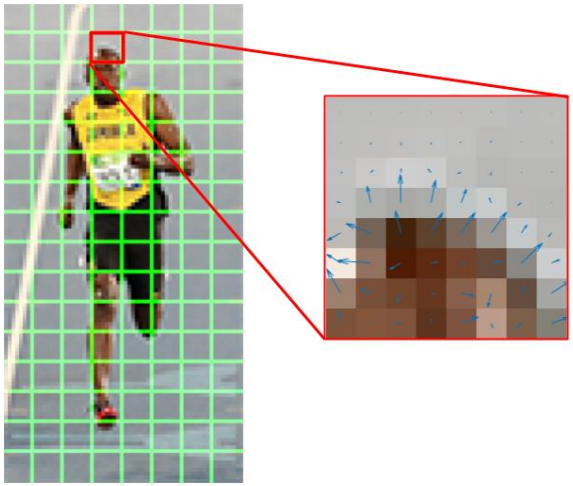
	100	
70	**Q**	120
	50	

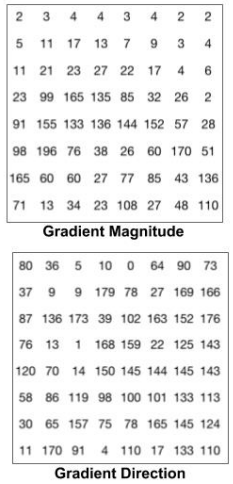


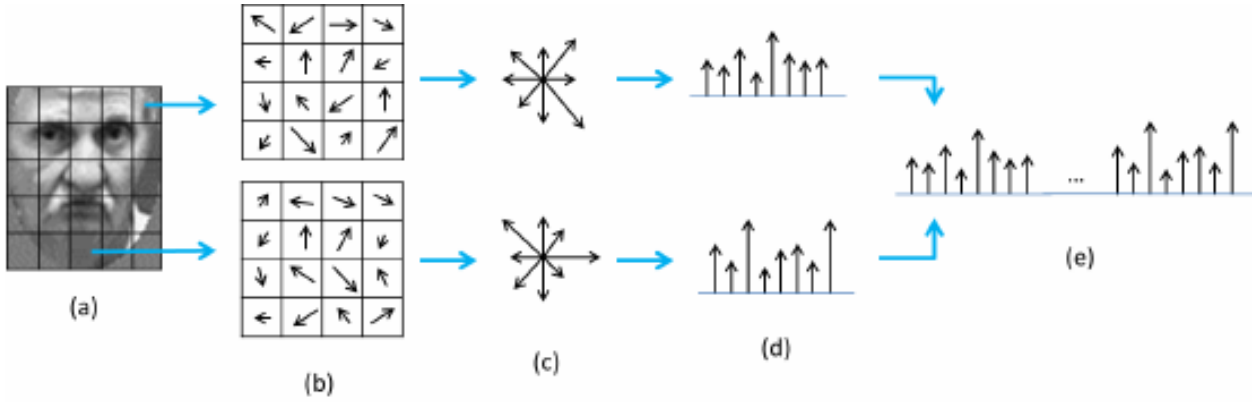
HOG

- Detection is done by feeding patches of different sizes to svm









HOG-BASED: ADVANTAGES

1. **Robustness to Illumination Changes** → relatively invariant to changes in illumination.
2. **Computationally Efficient** → suitable for real-time object detection applications.
3. **Localized Feature Representation** → HOG capture local gradient information within image patches, enabling the detection of objects based on their edge and gradient orientations.
4. **Interpretability** → human-interpretable HOG descriptors of object features so easy to understand and debug the process.
5. **Works Well with Linear Classifiers** → well-suited for use with linear classifiers like SVMs.

HOG-BASED: DISADVANTAGES

1. **Limited Spatial Information** → gradient calculated within localized image patches.
2. **Sensitive to Scale and Aspect Ratio** → Detectors may struggle if scale or aspect ratio vary significantly from the training data.
3. **Limited Discriminative Power** → may not fully capture the complex textures and patterns present in some objects.
4. **Difficulty with Occlusions** → **problems** with detecting partially occluded objects.
5. **Manual Feature Engineering** → Designing effective HOG descriptors often requires manual feature engineering and parameter tuning.
6. **Limited Performance on Complex Scenes** → In scenes with cluttered backgrounds, overlapping objects, or complex visual patterns, detectors may struggle.

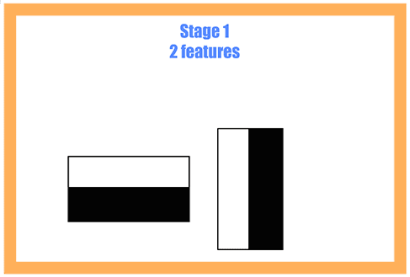
EARLY METHODS OF OBJECT DETECTION

2. Haar Cascades:

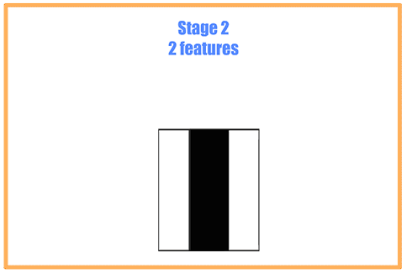
- Haar cascades are based on the Haar-like features, which are simple rectangular filters applied to different regions of an image.
- Viola and Jones introduced Haar cascades for face detection, but they can be used for general object detection tasks.
- Haar cascades involve a series of classifiers organized in a cascade structure, where each classifier progressively filters out regions of the image that are unlikely to contain the object of interest.

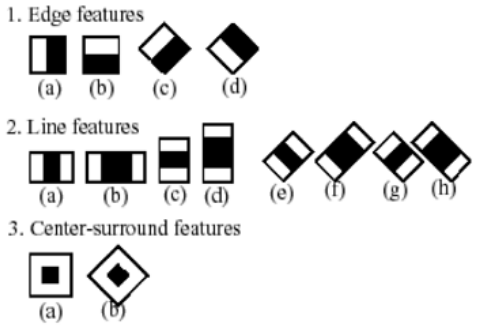
EARLY METHODS OF OBJECT DETECTION

- Each Haar feature is evaluated by subtracting the sum of pixel intensities within the white rectangles from the sum of pixel intensities within the black rectangles.
- The difference between these sums is used as the feature value.
- Haar features are applied at multiple scales and locations across the image to capture variations in object appearance, size, and position.
- Sliding windows are moved across the image, and Haar features are calculated within each window at different scales.



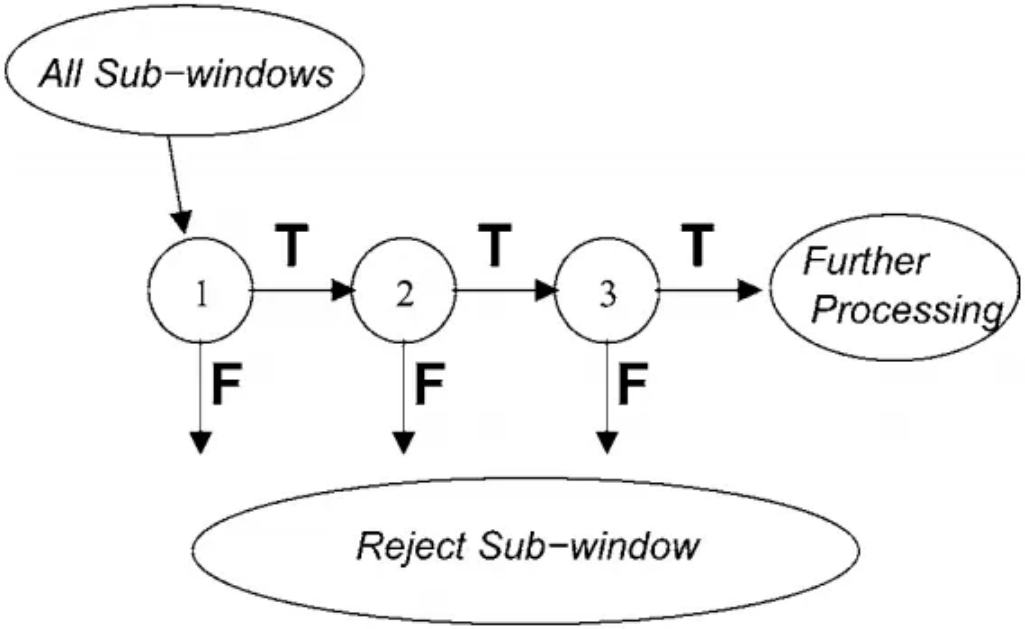
Stage 1 Running



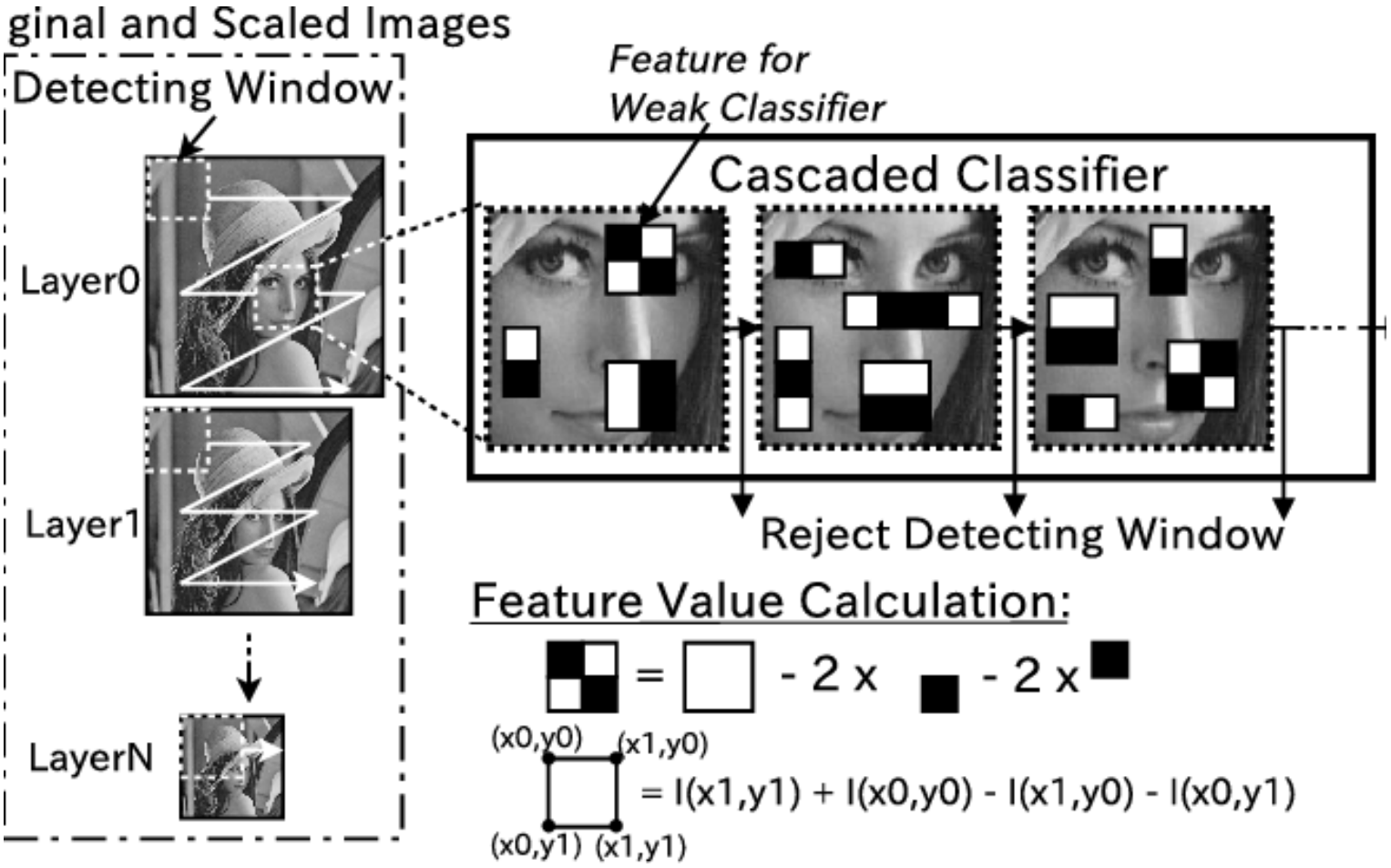


EARLY METHODS OF OBJECT DETECTION

- In the end, the sum of the values of weak classifiers is compared with the threshold of the cascade, and a decision is made whether the object is found or not by this cascade.



EARLY METHODS OF OBJECT DETECTION



HAAR-CASCADE-BASED: ADVANTAGES

1. **Fast Detection Speed** → computationally efficient and can perform real-time object detection on resource-constrained devices.
2. **Simple Implementation** → relatively straightforward compared to more complex machine learning algorithms.
3. **Robustness to Illumination Changes** → Haar features are designed to capture local intensity variations so relatively robust to changes in illumination and contrast.
4. **Minimal Training Data Requirements:** Haar cascade classifiers can achieve reasonable performance with relatively small training datasets.
5. **Suitable for Detecting Simple Objects:** Haar cascade classifiers are well-suited for detecting simple objects with distinctive features, such as faces, eyes, and frontal objects, in controlled environments.

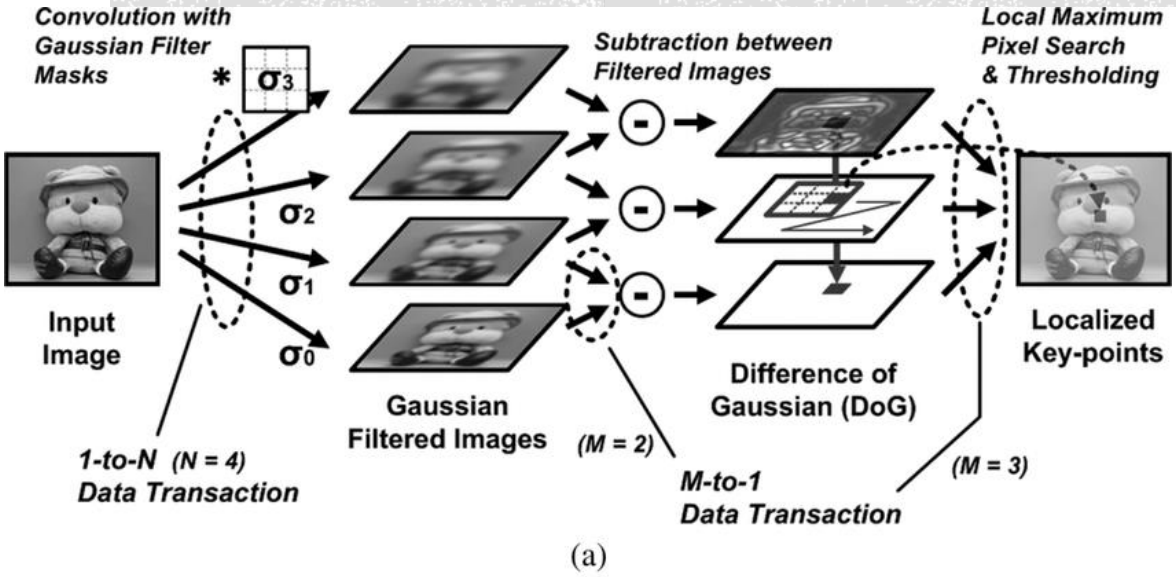
HAAR-CASCADE-BASED: DISADVANTAGES

1. **Limited Accuracy** → may struggle to achieve high accuracy in complex object detection tasks, especially when objects exhibit variations in scale, rotation, and viewpoint.
2. **Difficulty with Occlusions** → may fail to detect objects that are partially occluded or obscured by other objects in the scene.
3. **Sensitive to Background Clutter** → may produce false positive detections in scenes with high background clutter or complex visual textures.
4. **Limited Ability to Capture Spatial Relationships** → Haar features capture local intensity variations but may not effectively capture spatial relationships between different parts of an object, limiting their ability to handle objects with complex geometric structures.
5. **Limited Flexibility in Feature Representation** → Haar features have limited expressive power compared to more advanced feature representations used in deep learning-based approaches, which may restrict their ability to capture subtle object characteristics and variations.
6. **Manual Feature Engineering** → Designing effective Haar cascade classifiers often requires manual feature engineering and parameter tuning.
7. **Difficulty with Non-Frontal Views** → may struggle to detect objects in non-frontal views or under significant pose variations.

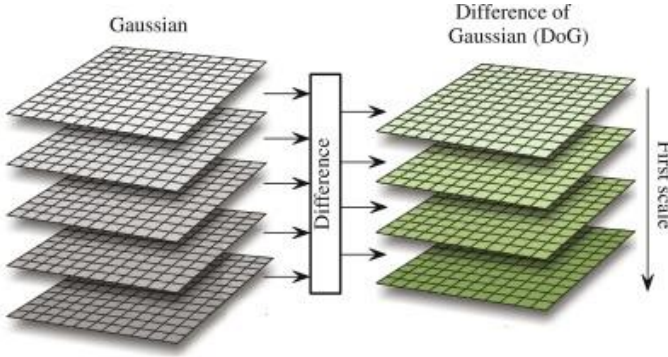
EARLY METHODS OF OBJECT DETECTION

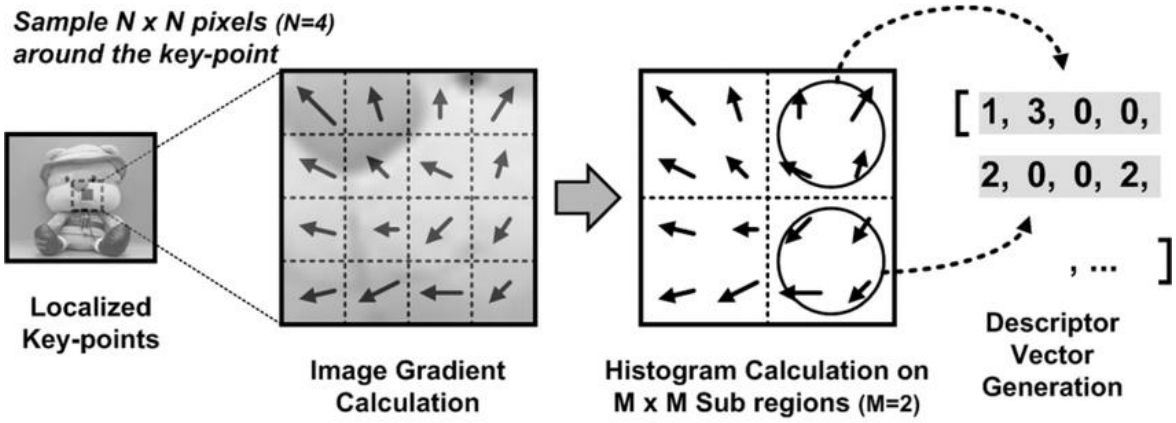
3. **Scale-Invariant Feature Transform (SIFT)**:
   - SIFT is a method for detecting and describing local features in images.
   - It identifies key points (interest points) in an image that are invariant to scale, rotation, and illumination changes.
   - SIFT extracts feature descriptors based on the local image gradients and histograms of gradient orientations.
   - These descriptors are robust and can be matched across different images for tasks like object recognition and image stitching.
   - **The major advantage of SIFT features, over-edge features, or hog features is that they are not affected by the size or orientation of the image.**

PIPELINE

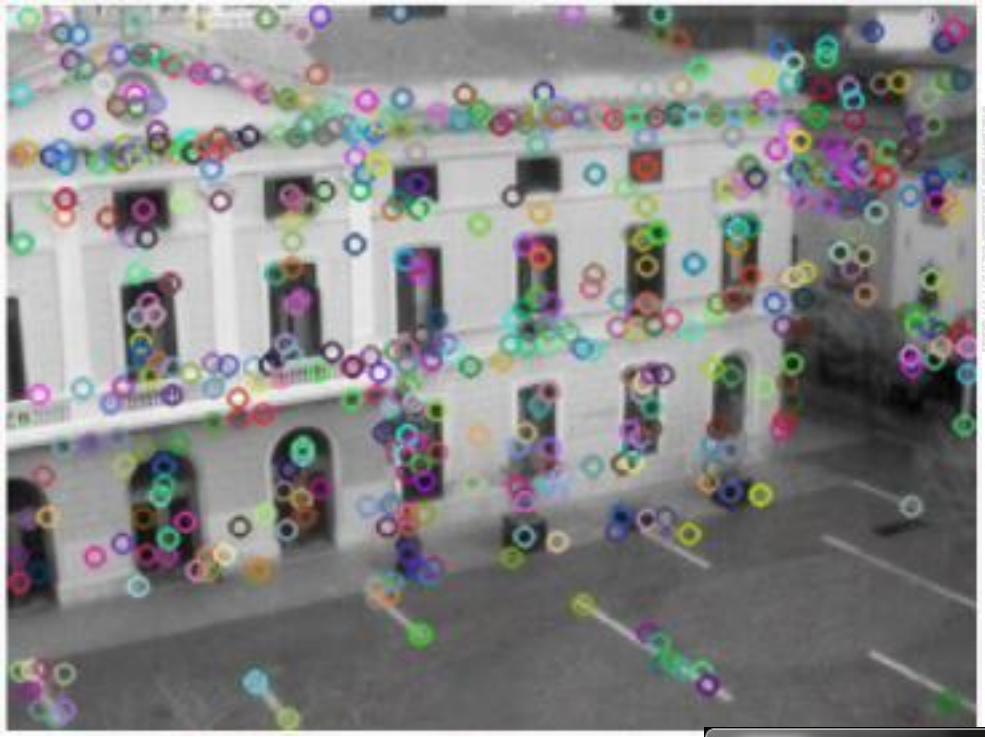

Convolution with Gaussian Filter Masks
σ3
σ2
σ1
σ0
Input Image
Gaussian Filtered Images
Subtraction between Filtered Images
Difference of Gaussian (DoG)
Local Maximum Pixel Search & Thresholding
Localized Key-points
1-to-N (N = 4) Data Transaction
(M = 2)
M-to-1 Data Transaction
(M = 3)

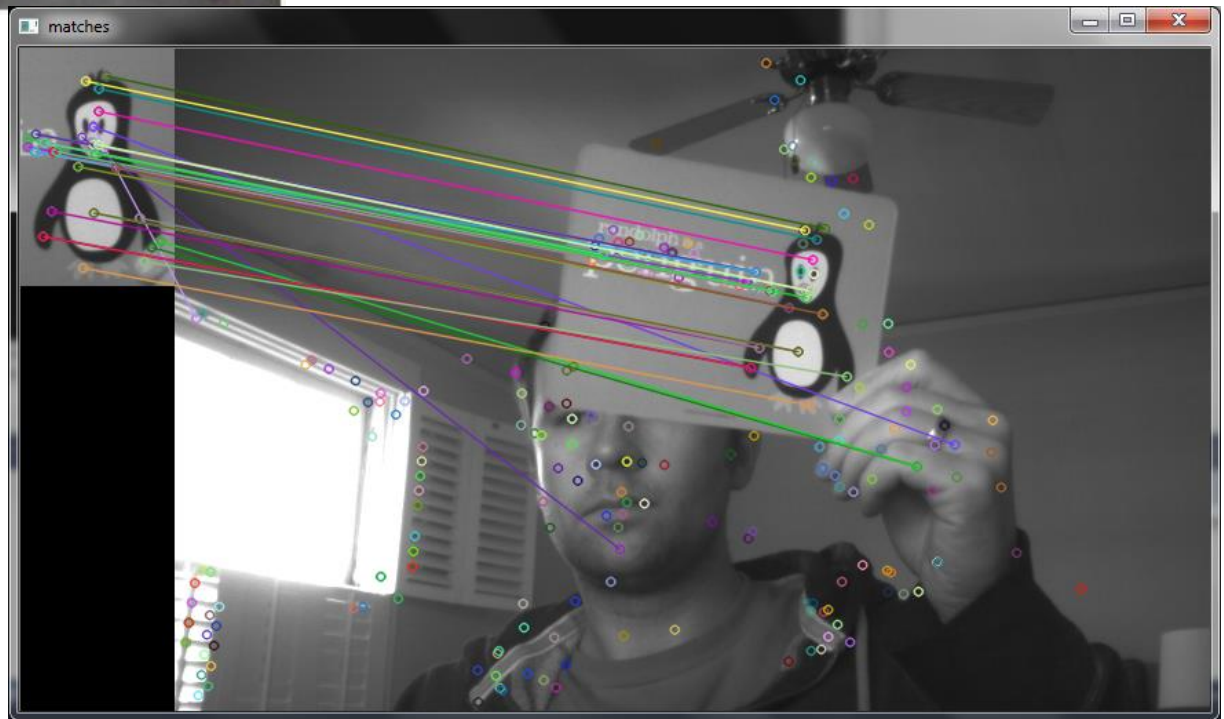
(a)

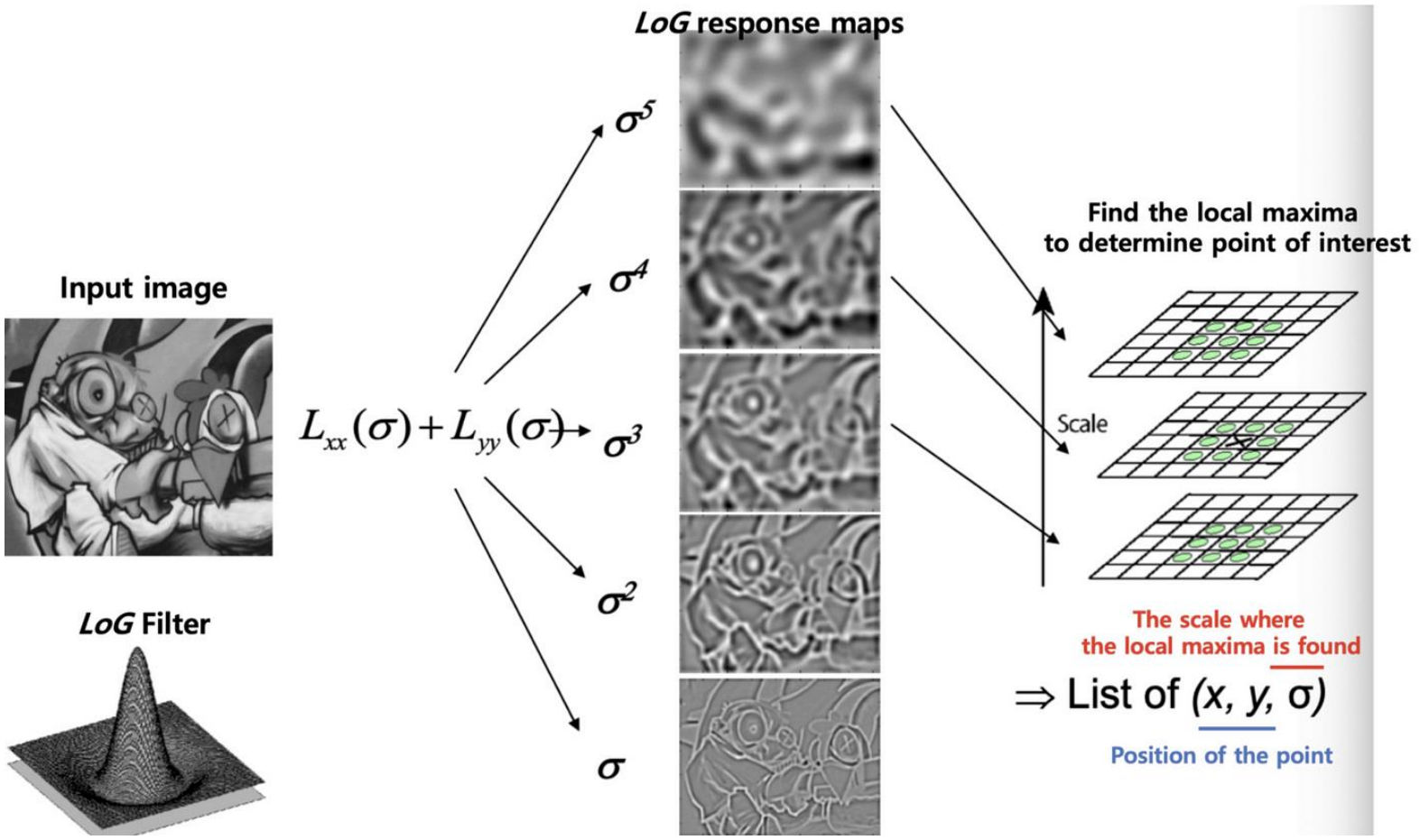

Gaussian
Difference of Gaussian (DoG)
Difference
First scale
Difference
Second scale

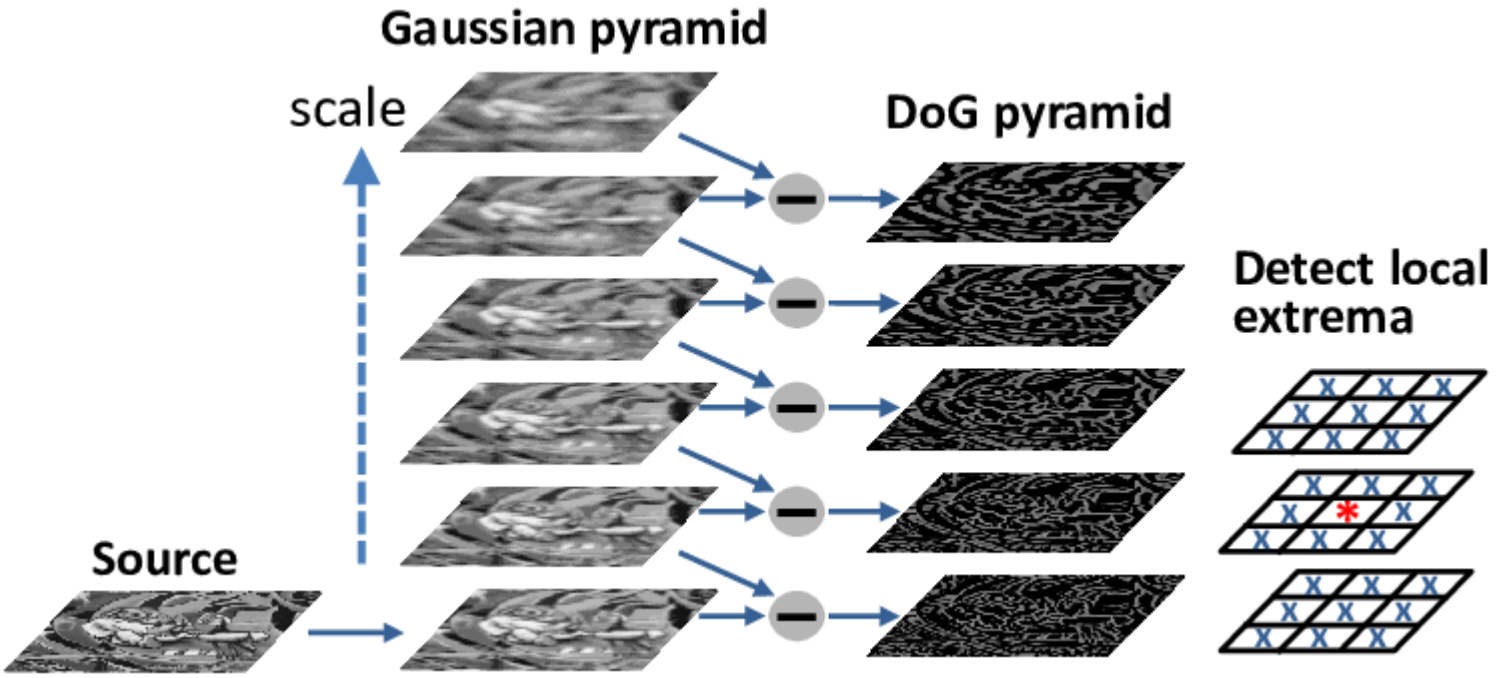

Sample N x N pixels (N=4) around the key-point
Localized Key-points
Image Gradient Calculation
Histogram Calculation on M x M Sub regions (M=2)
[1, 3, 0, 0,
2, 0, 0, 2,
, ...]
Descriptor Vector Generation

(b)




matches

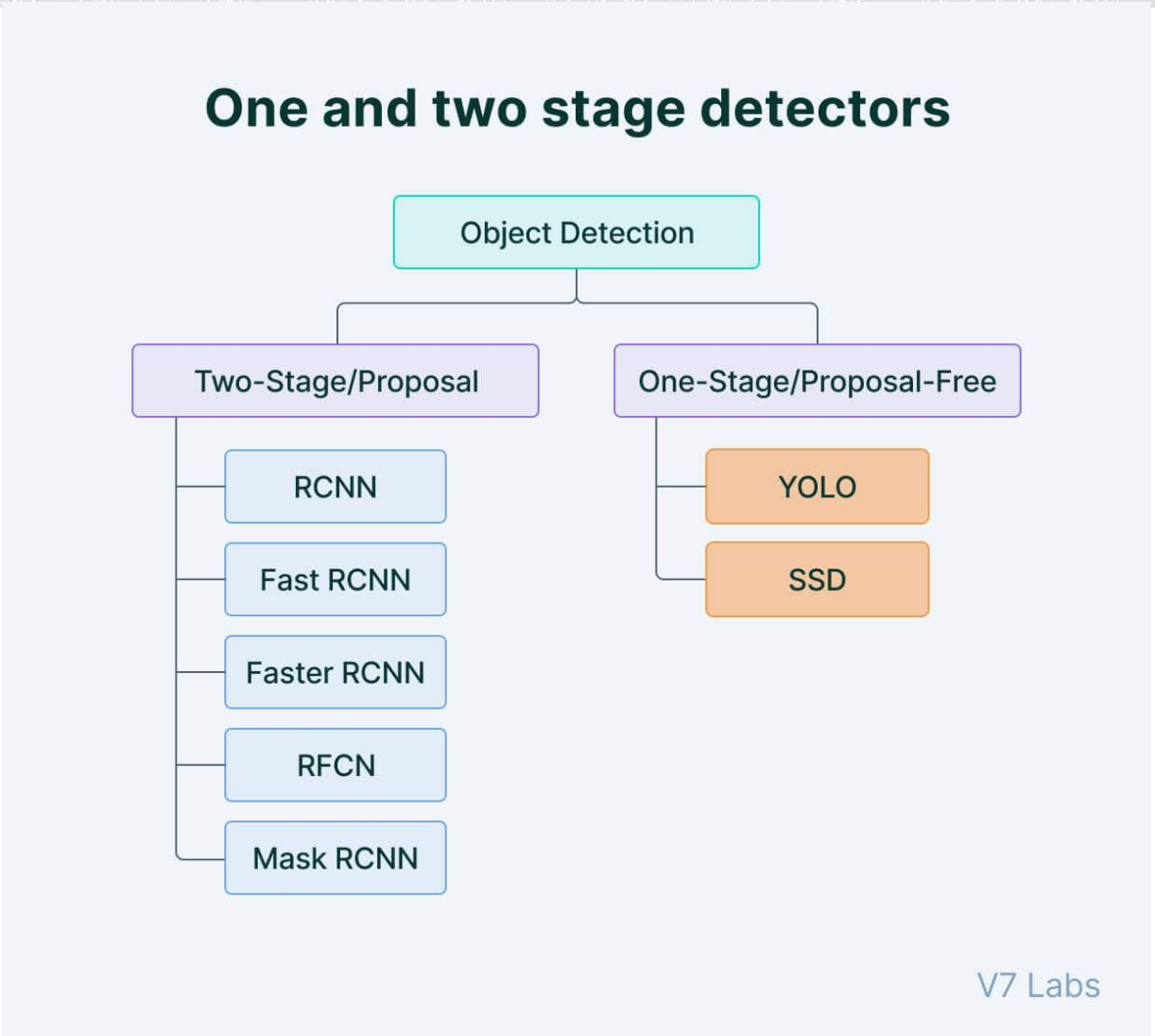

LoG response maps
Input image
LoG Filter
$L_{xx}(\sigma)+L_{yy}(\sigma)$
σ^5
σ^4
σ^3
σ^2
σ
Find the local maxima
to determine point of interest
Scale
The scale where
the local maxima is found
⇒ List of (x, y, σ)
Position of the point


Gaussian pyramid
scale
DoG pyramid
Detect local extrema
Source

OTHER ALGORITHMS

- **Speeded-Up Robust Features (SURF)**:
  - SURF is a feature detection and description algorithm similar to SIFT but computationally more efficient.
  - It uses integral images and box filters to approximate the convolutions in the computation of image gradients.
  - SURF descriptors capture the local image structures and are used for matching and recognizing objects in images.
- **Template Matching**:
  - Template matching involves comparing a template image (a small image patch representing the object) with different regions of the input image.
  - It measures the similarity between the template and the image patches using methods like correlation.
  - Template matching can be effective for detecting objects with well-defined and consistent visual patterns, but it is sensitive to changes in scale, rotation, and occlusion.

CATEGORIES OF METHODS



SINGLE SHOT OBJECT DETECTION

- Single-shot object detection uses a single pass of the input image to make predictions
- It processes an entire image in a single pass, making them computationally efficient.
- Single-shot object detection is generally less accurate than other methods, and it's less effective in detecting small objects.
- YOLO is a single-shot detector that uses a fully convolutional neural network (CNN) to process an image.

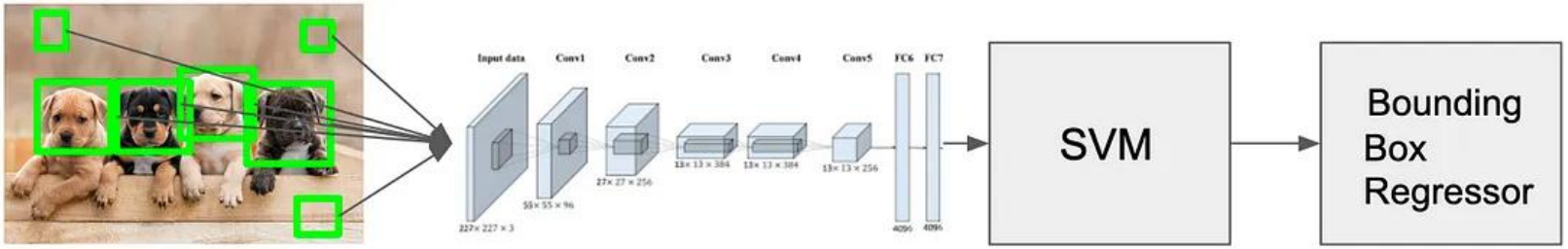
TWO SHOT OBJECT DETECTION

- Two-shot object detection uses two passes of the input image to make predictions about the presence and location of objects.
  - The first pass is used to generate a set of proposals or potential object locations
  - The second pass is used to refine these proposals and make final predictions.
  - This approach is more accurate than single-shot object detection but is also more computationally expensive.
- Choice between single-shot and two-shot object detection depends on the specific requirements and constraints of the application
- Single-shot object detection is better suited for real-time applications, while two-shot object detection is better for applications where accuracy is more important.

NEW METHODS: DEEP-LEARNING-BASED

- **Region-based Convolutional Neural Networks (R-CNN) Family:**
  - **R-CNN**
    - one of the pioneering methods that used deep learning for object detection.
    - generates region proposals using a selective search algorithm and then extracts features using a pre-trained CNN.
    - features are later classified using SVMs.
  - **Fast R-CNN:**
    - improved upon R-CNN by sharing convolutional features across region proposals, making it faster and more efficient.
  - **Faster R-CNN**
    - introduced the Region Proposal Network (RPN), which shares convolutional features with the object detection network.
    - unified the region proposal and object detection into a single network, leading to further speed improvements.

STEPS



1) Region proposals with selective search

2) Feature extraction with CNN

3) Classify features with a SVM

4) Improve the bounding box

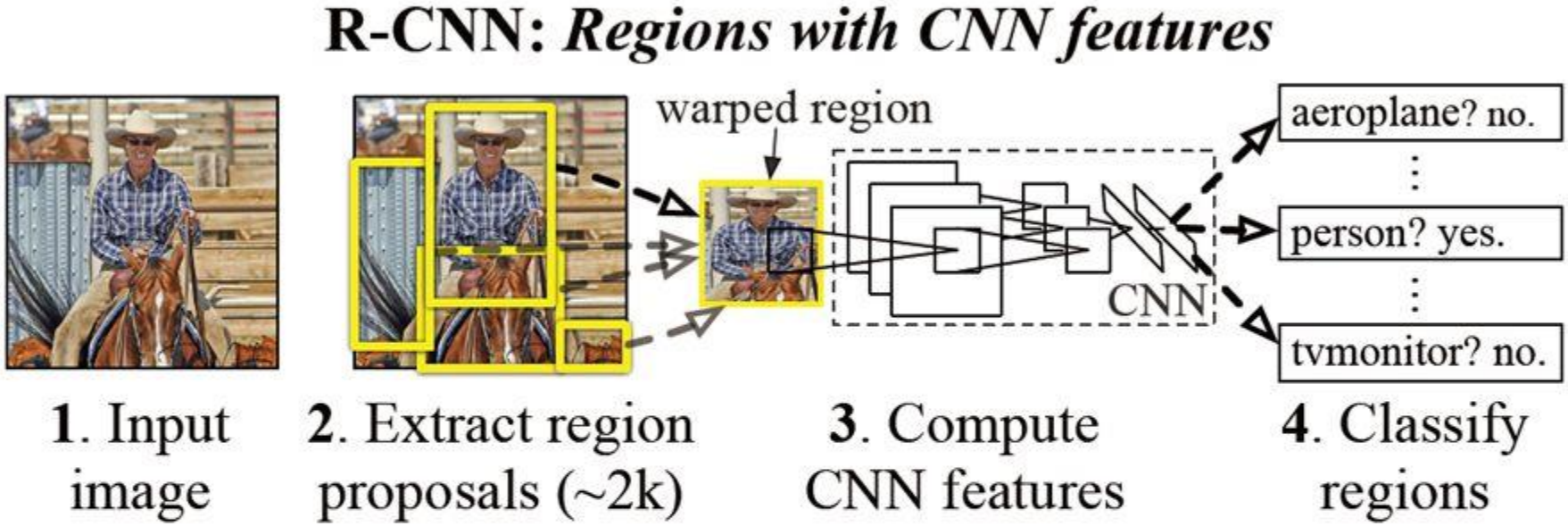
PIPELINE IN DETAILS

1. **Selective Search for Region Proposals** → use a selective search algorithm to generate a set of region proposals
2. **Region Proposal Pooling** → resize to a fixed size to create region of interest (RoI) boxes
3. **Feature Extraction:** → Each RoI box is passed through a pre-trained CNN (AlexNet or VGGNet, ..) to extract feature vectors.
   - CNN pre-trained on a large dataset like ImageNet for generic feature learning.
4. **SVM Classification** → feature vectors extracted from the RoI boxes are fed into SVMs for object classification.

PIPELINE IN DETAILS

5. **Bounding Box Regression** → refinement of the location and size of the bounding boxes around the objects
6. **Non-maximum Suppression (NMS)** → filter out redundant detections by removing overlapping bounding boxes and retaining only the highest-scoring bounding boxes while discarding others that have significant overlap with them.
7. **Post-processing** → may include thresholding to remove low-confidence detections, filtering based on object size or aspect ratio, and applying additional constraints to improve the accuracy of the final detections.

R-CNN: Pipeline Overview



Step1. Input an image
Step2. Use selective search to obtain ~2k proposals
Step3. Warp each proposal and apply CNN to extract its features
Step4. Adopt class-specified SVM to score each proposal
Step5. Rank the proposals and use NMS to get the bboxes.
Step6. Use class-specified regressors to refine the bboxes' positions.

Ross Girshick et al. Rich feature hierarchies for accurate object detection and semantic segmentation, CVPR14

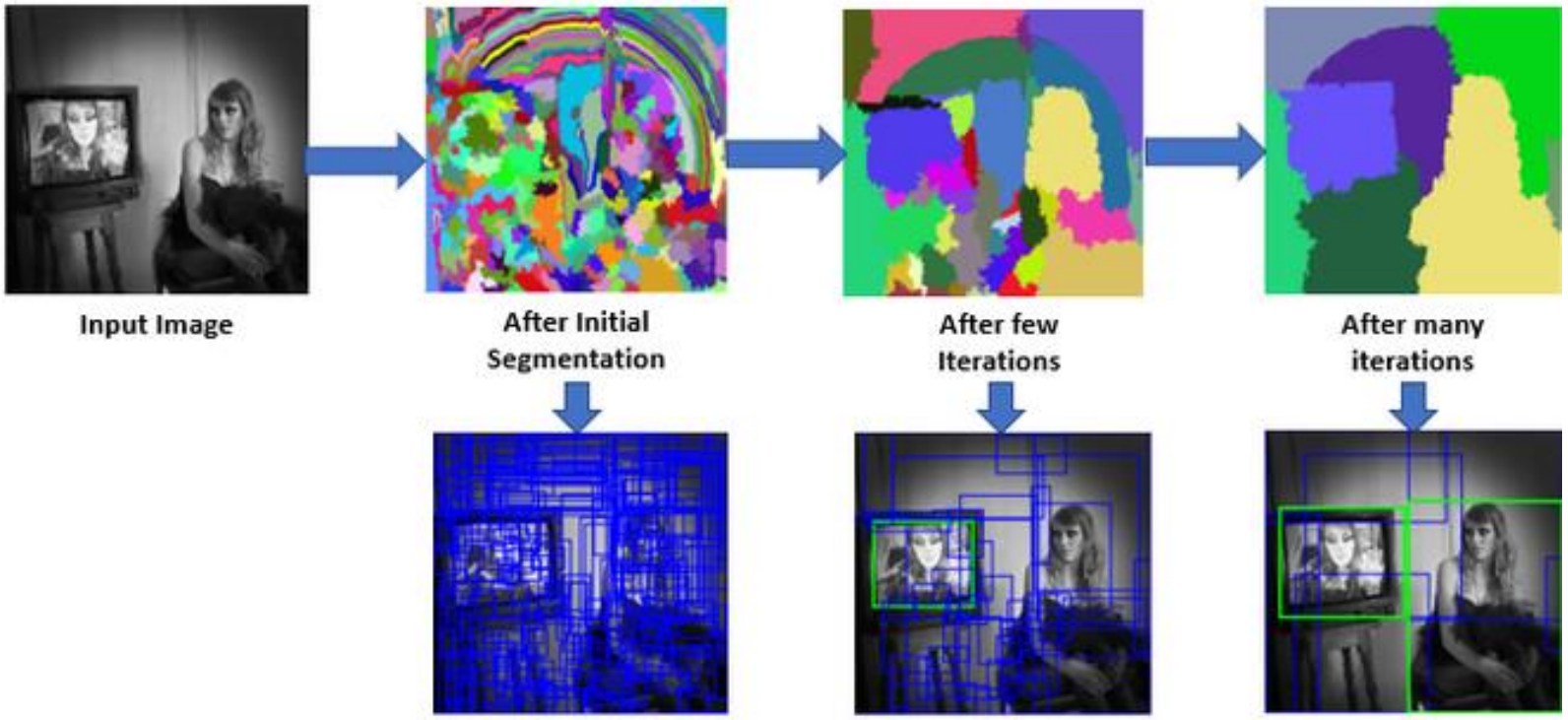
SELECTIVE SEARCH ALGORITHM

- **Selective Search** was first introduced in 2012 that combines the strength of both exhaustive search and segmentation.
  - exhaustive search → aims at finding all the possible locations by systematically enumerating all the possible candidates.
  - segmentation → uses the image structure to cluster pixels into different regions
- **Other region proposal algorithms**
  - Objectness
  - Constrained Parametric Min-Cuts for Automatic Object Segmentation
  - Category Independent Object Proposals
  - Randomized Prim
  - **Details: https://learnopencv.com/selective-search-for-object-detection-cpp-python/**

SELECTIVE SEARCH ALGORITHM

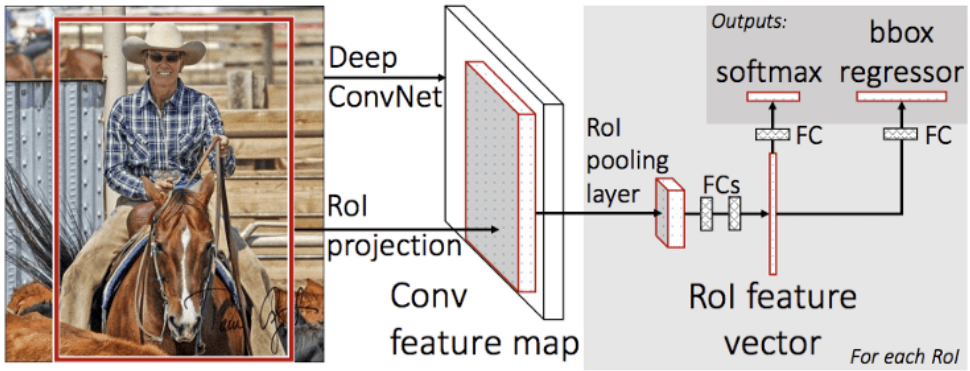
- **Selective Search Steps**:
  - **Image segmentation** → segment the image into smaller segments or regions based on color, texture, intensity, and other low-level features.
  - **Grouping Segments** → merge segment that share similar features. helps in capturing objects that may span multiple segments.
  - **Region Merging** → iteratively merges similar regions to form larger candidate regions until the entire image is represented by a small number of candidate regions
  - **Ranking Region Proposals** → region proposals are ranked based on their likelihood of containing objects.
    - Selective Search uses a combination of similarity metrics, such as color histograms, texture features, and shape information, to score the region proposals.

SELECTIVE SEARCH ALGORITHM



FAST R-CNN

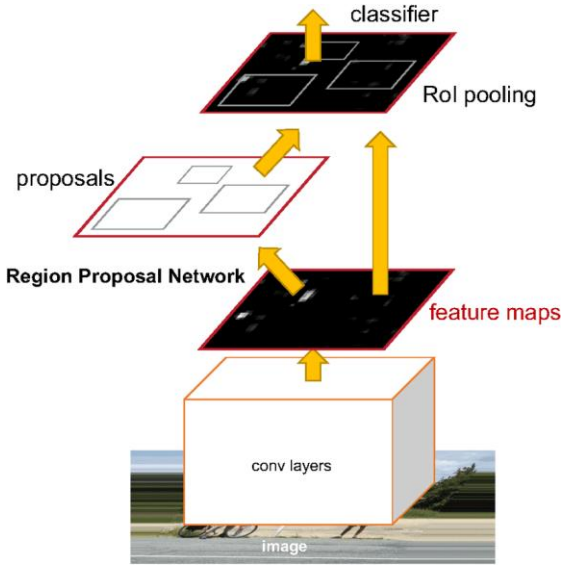
- In R-CNN → each bounding box was independently classified by the image classifier.
  - We have 2000 region proposals and the image classifier calculated a feature map for each region proposal.
  - Expensive process.
- Fast R-CNN → version to speed-up the detection.
- Main idea
  - calculate a single feature map for the entire image instead of 2000 feature maps for the 2000 region proposals.
  - For each region proposal, a region of interest (RoI) pooling layer extracted a fixed-length feature vector from the feature map.



FASTER R-CNN

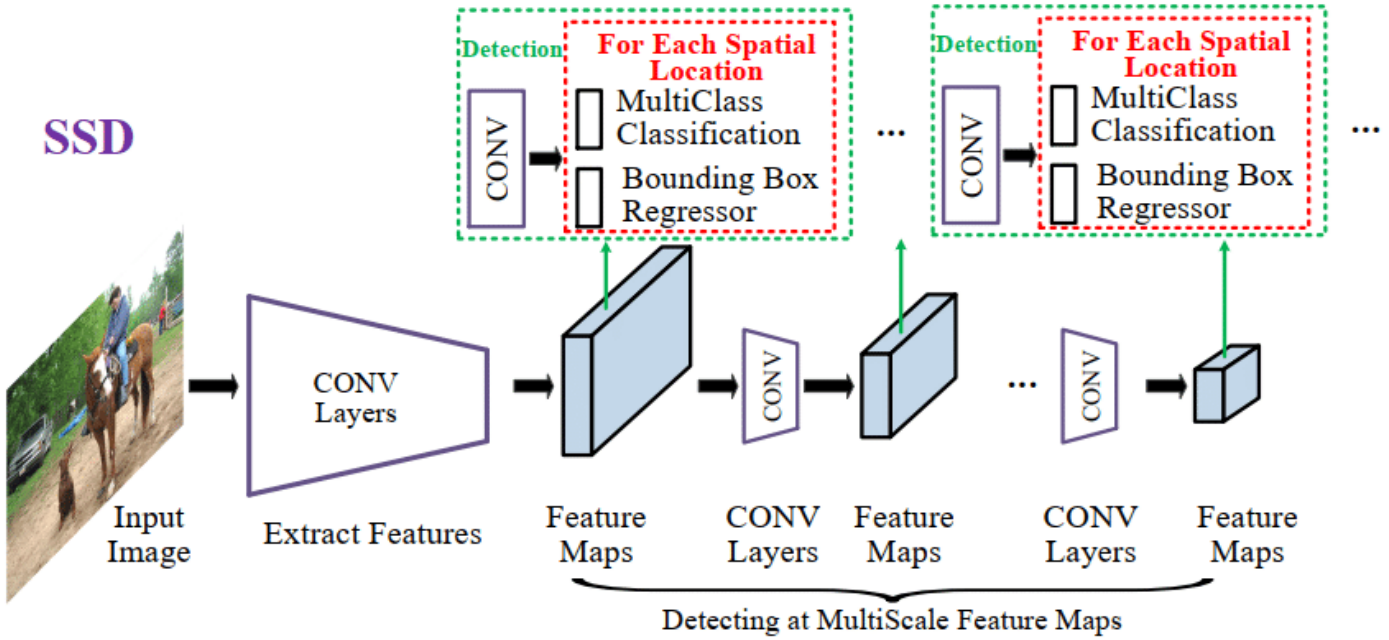
- In Fast R-CNN → computation for classifying 2000 region proposals was shared while the part of the algorithm generating the region proposals did not share any computation with the part that performed image classification.
- Faster R-CNN → version to speed-up the detection.
- Main idea
  - calculating region proposals and image classification could use the same feature map and therefore share the computational load.

A CNN is used to produce a feature map of the image which was simultaneously used for training a region proposal network and an image classifier.



SSD

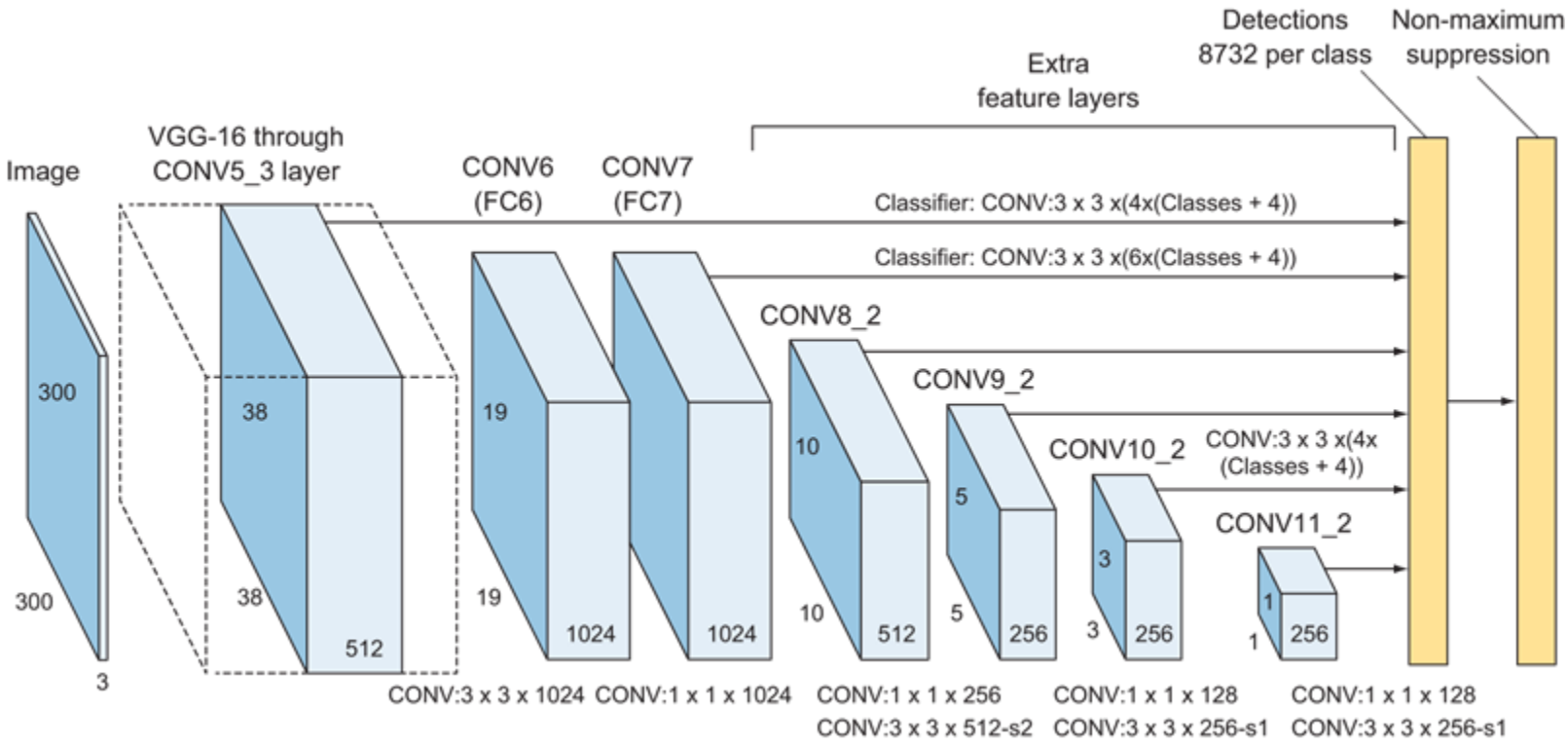
- **Single Shot Multibox Detector (SSD):**
  - performs detection directly from feature maps at multiple scales.
  - predicts bounding boxes and class probabilities using convolutional filters applied to feature maps at different layers of the network.
  - SSD achieves high accuracy and real-time performance.



SSD

- **Single Shot Multibox Detector (SSD):**
  - ✓**Single Shot:** this means that the tasks of object localization and classification are done in a *single forward pass* of the network
  - ✓**MultiBox:** this is the name of a technique for bounding box regression developed by Szegedy et al.
  - ✓**Detector:** The network is an object detector that also classifies those detected objects

SSD


Detections
8732 per class
Non-maximum
suppression
Extra
feature layers
Image
VGG-16 through
CONV5_3 layer
CONV6
(FC6)
CONV7
(FC7)
Classifier: CONV:3 x 3 x(4x(Classes + 4))
Classifier: CONV:3 x 3 x(6x(Classes + 4))
CONV8_2
CONV9_2
CONV10_2
CONV:3 x 3 x(4x
(Classes + 4))
CONV11_2
300
300
3
38
38
512
19
19
1024
1024
10
10
512
5
5
256
3
3
256
1
1
256
CONV:3 x 3 x 1024
CONV:1 x 1 x 1024
CONV:1 x 1 x 256
CONV:3 x 3 x 512-s2
CONV:1 x 1 x 128
CONV:3 x 3 x 256-s1
CONV:1 x 1 x 128
CONV:3 x 3 x 256-s1

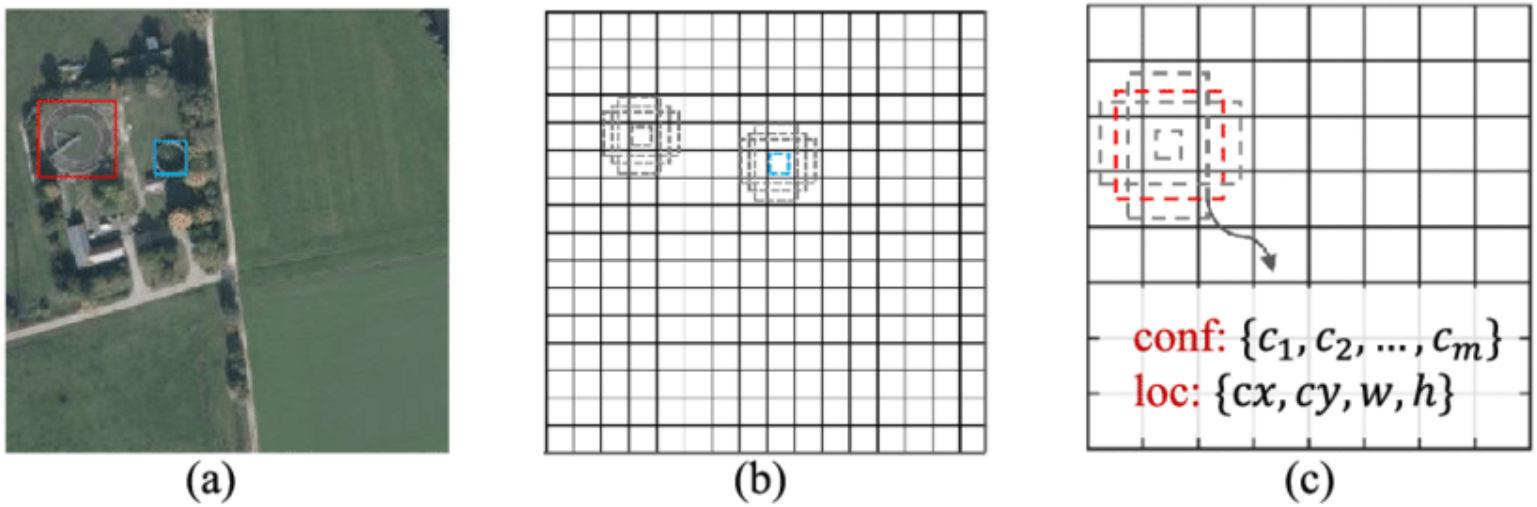
HOW IT WORKS

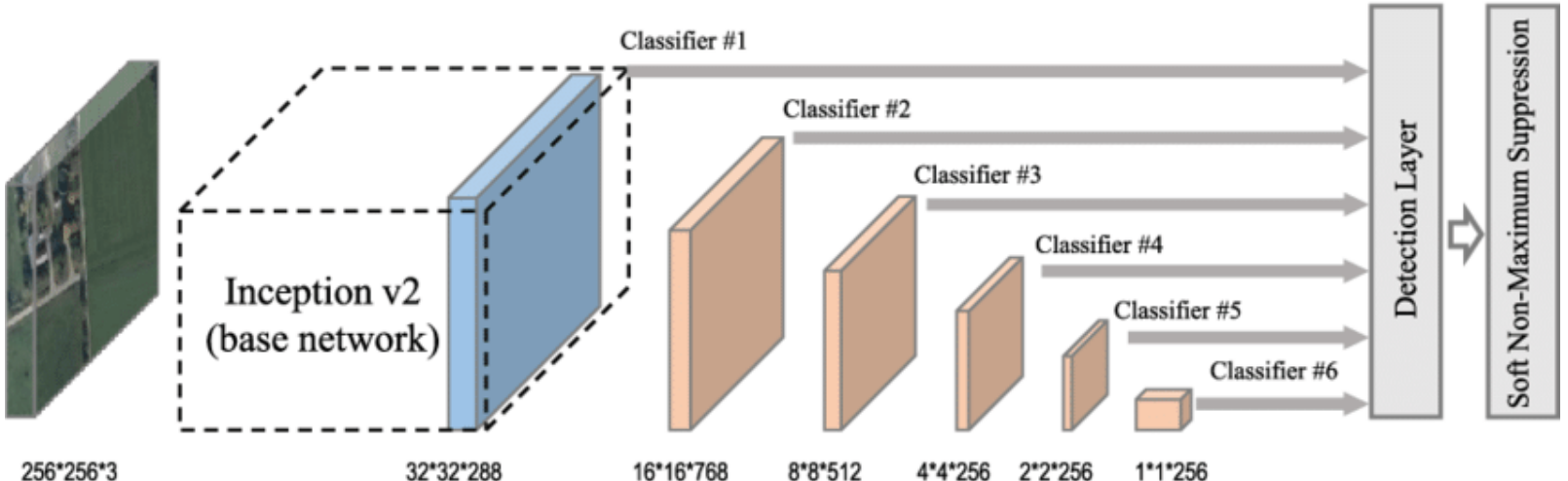
- **Base Convolutional Network**: SSD starts with a base convolutional network, typically a pre-trained convolutional neural network (CNN) like VGG or ResNet. This network is responsible for extracting feature maps from the input image.
- **Feature Pyramid Network (FPN)**: SSD then incorporates a feature pyramid network to extract features at multiple scales. This helps in detecting objects of different sizes in the image.
- **Multiscale Feature Maps**: SSD applies a set of convolutional layers of different sizes (typically 3x3 and 1x1 convolutions) to the feature maps obtained from the base network. These layers are used to predict the presence of objects at various locations and scales within the image.
- **Default Boxes**: SSD uses a set of default bounding boxes (or default boxes) with different aspect ratios at each spatial location of the feature maps. These default boxes are used to predict the bounding boxes for objects.

HOW IT WORKS

- **Predictions**: For each default box, SSD predicts two things:
  - The class scores: These scores represent the probability of each class being present in the box.
  - The offset vectors: These vectors encode the offsets between the default box and the ground truth bounding box.
- **Non-maximum Suppression (NMS)**: After obtaining the predicted bounding boxes and their associated confidence scores, SSD applies non-maximum suppression to remove redundant bounding boxes and keep only the most confident ones.

HOW IT WORKS


conf: $\{c_1, c_2, \ldots, c_m\}$
loc: $\{cx, cy, w, h\}$
(a)
(b)
(c)


Classifier #1
Classifier #2
Classifier #3
Classifier #4
Classifier #5
Classifier #6
Inception v2
(base network)
Detection Layer
Soft Non-Maximum Suppression
256*256*3
32*32*288
16*16*768
8*8*512
4*4*256
2*2*256
1*1*256

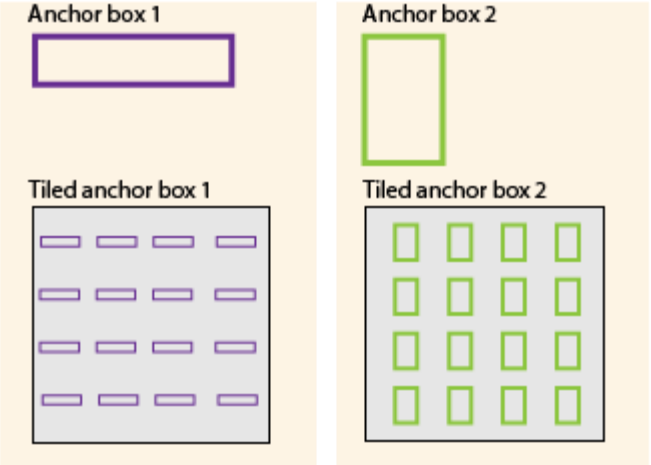
ANCHOR BOXES

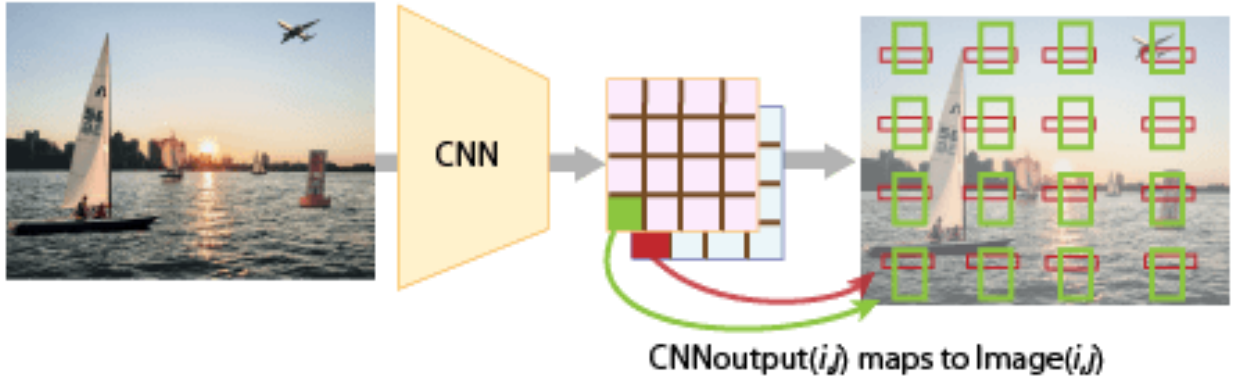
- Anchor boxes → **predefined boxes of various sizes and aspect ratios** placed across different feature maps of the SSD network.
- Number of anchor boxes generated by SSD depends on:

1. **Multiple Aspect Ratios**: SSD typically generates default boxes with multiple aspect ratios, such as 1:1, 1:2, 2:1, etc., to handle objects of different shapes.
2. **Multiple Scales**: SSD also generates default boxes at multiple scales to detect objects at different sizes. The scales are often chosen based on the size of the input image and the network architecture.
3. **Feature Map Sizes**: SSD generates default boxes at different feature map layers. Typically, the feature maps at higher layers capture finer details and smaller objects, while those at lower layers capture larger objects.

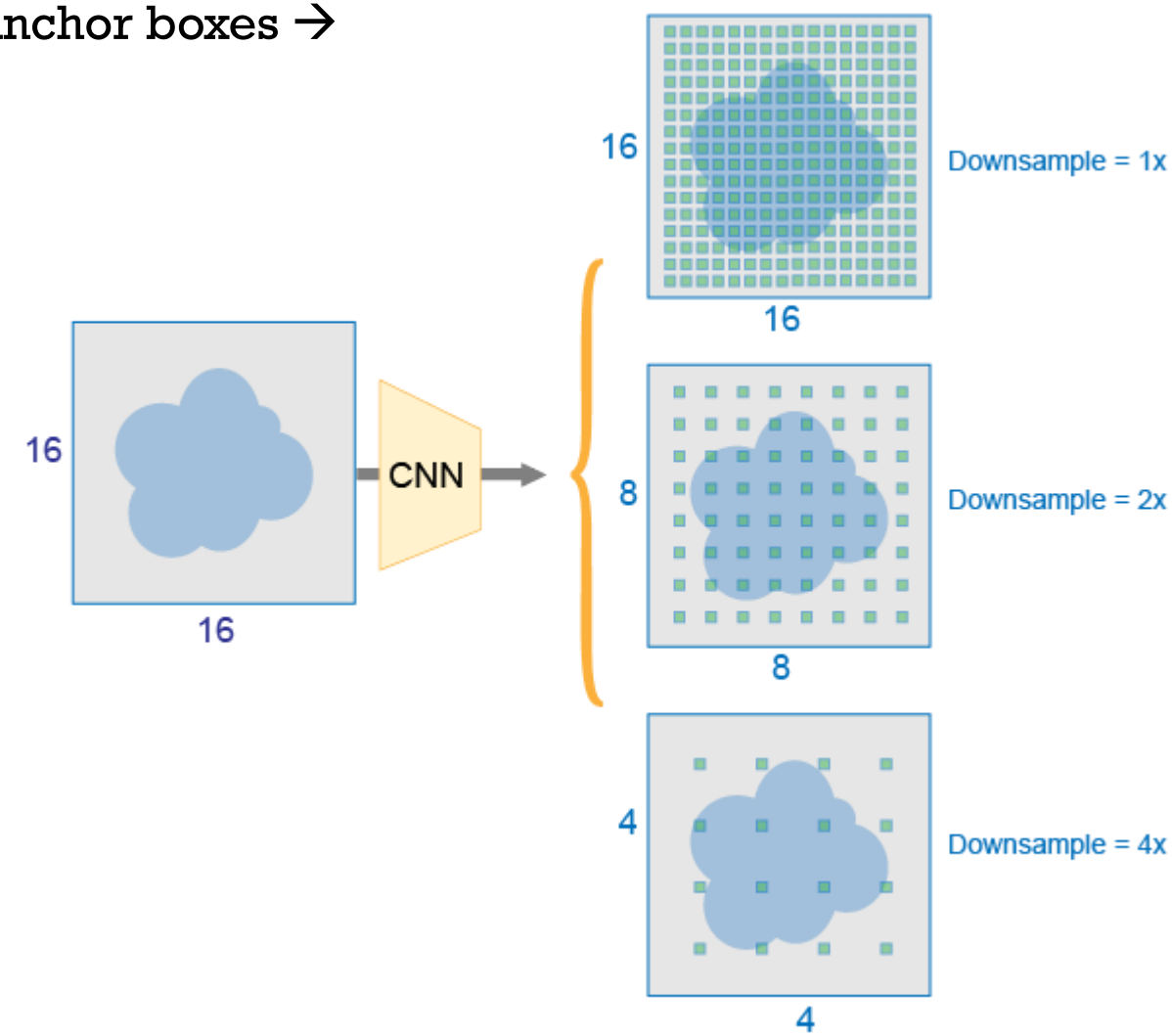
ANCHOR BOXES

- Anchor boxes → **predefined boxes of various sizes and aspect ratios** placed across different feature maps of the SSD network.





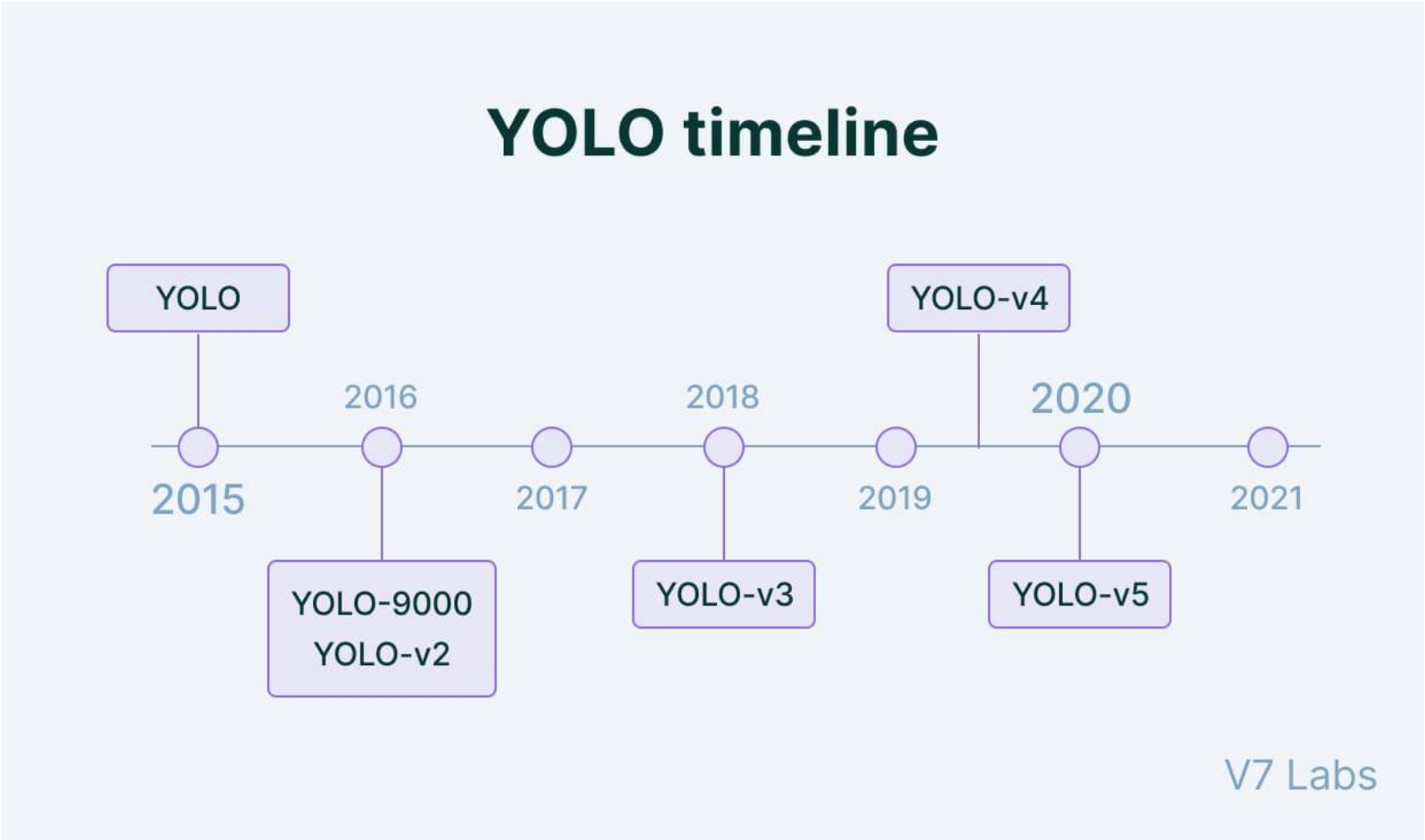
- Anchor boxes →



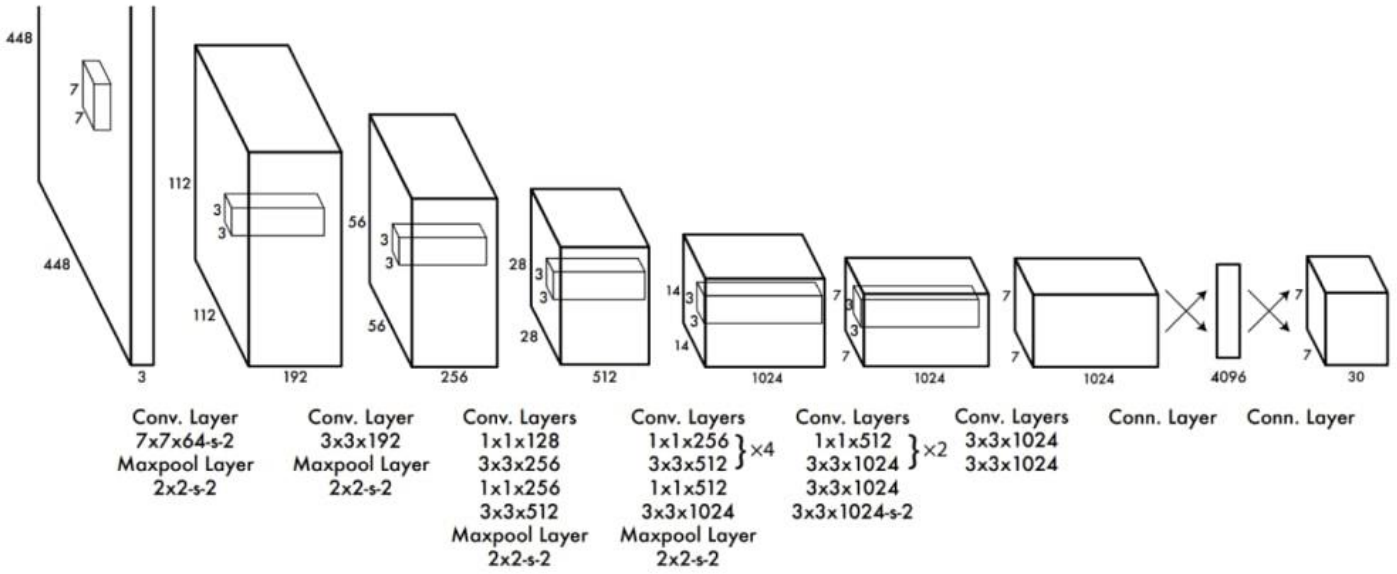
YOLO

- **You Only Look Once (YOLO):**
  - YOLO is another real-time object detection method that frames object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. YOLO divides the input image into a grid and predicts bounding boxes and class probabilities directly from the grid cells.
- **YOLOv2, YOLOv3, YOLOv4, and YOLOv5:**
  - These are successive iterations and improvements over the original YOLO architecture, introducing various enhancements such as better feature extraction, anchor boxes, and improved training strategies.

VERSIONS


YOLO timeline
YOLO
YOLO-v4
2016
2018
2020
2015
2017
2019
2021
YOLO-9000
YOLO-v2
YOLO-v3
YOLO-v5
V7 Labs

NETWORK ARCHITECTURE



The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

YOLO PIPELINE

- **Input Image Division**:
  - YOLO takes the entire input image and divides it into an S×S grid. Each grid cell is responsible for predicting bounding boxes and associated class probabilities.
- **Bounding Box Prediction**:
  - Each grid cell predicts a fixed number of bounding boxes (usually BB) along with confidence scores that indicate the likelihood of an object being present and how accurate the bounding box is.
  - Each bounding box is represented by 5 values: (x,y,w,h,confidence).
    - (x,y) represent the center coordinates of the bounding box relative to the grid cell.
    - (w,h) represent the width and height of the bounding box relative to the entire image.
    - Confidence indicates how confident the model is that the box contains an object and how accurate it thinks the box is.

YOLO PIPELINE

- **Class Prediction**:
  - Each grid cell also predicts the probability of different classes being present in each bounding box.
  - The model performs multi-class classification to predict the probability distribution over all classes for each bounding box.
- **Non-Maximum Suppression (NMS)**:
  - YOLO applies non-maximum suppression to filter out redundant bounding boxes.
  - It removes boxes with low confidence scores and performs NMS to keep only the most confident boxes for each object.
  - This helps eliminate duplicate detections and refine the final set of bounding boxes.
- **Output**:
  - The output of YOLO is a set of bounding boxes along with their associated class probabilities.
  - Each bounding box is associated with a class label and a confidence score.

ZERO-SHOT DETECTION ???