Deep Learning for Object Detection, Classification and Segmentation

Dr.Mohan Raj,
Data Scientist,
HCL Technologies,
Chennai.
Twitter - @mohanrajphd
mohanraj4072@gmail.com

Agenda

- Need for Deep Learning
- Image Classification
- Various CNN architecture for Image Classification
- Object Detection
- Various CNN architectures for Object Detection
- Image Segmentation
- Demo

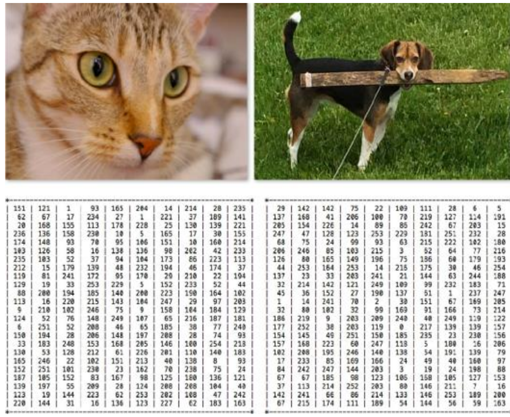
IMAGE CLASSIFICATION

Image classification is the task of assigning a label to an image from a predefined set of categories.

- Let's assume the set of possible categories are:
  categories = {cat, dog, panda}
- Classification algorithm assign multiple labels to the image via probabilities, such as
  - dog: 95%
  - cat: 4%
  - panda: 1%.

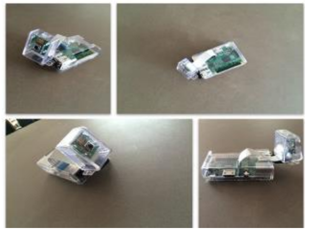
SEMANTIC GAP

- It should be fairly trivial for us to tell the difference between the two photos – there is clearly **a cat** on the left and **a dog** on the right. But all a computer sees is **two big matrices of pixels** (bottom).
- The **semantic gap** is the difference between how a human perceives the contents of an image versus how an image can be represented in a way a computer can understand the process.
- Visual examination of the two photos above can reveal the difference between the two species of an animal. But in reality, the computer has no idea there are animals in the image.
- We might describe the image as follows:
  - **Spatial**: The sky is at the top of the image and the sand/ocean are at the bottom.
  - **Color**: The sky is dark blue, the ocean water is a lighter blue than the sky, while the sand is tan.
  - **Texture**: The sky has a relatively uniform pattern, while the sand is very coarse.



Feature extraction is the process of taking an input image, applying an algorithm, and obtaining a feature vector (i.e., a list of numbers) that quantifies our image.

REAL-TIME CHALLENGES



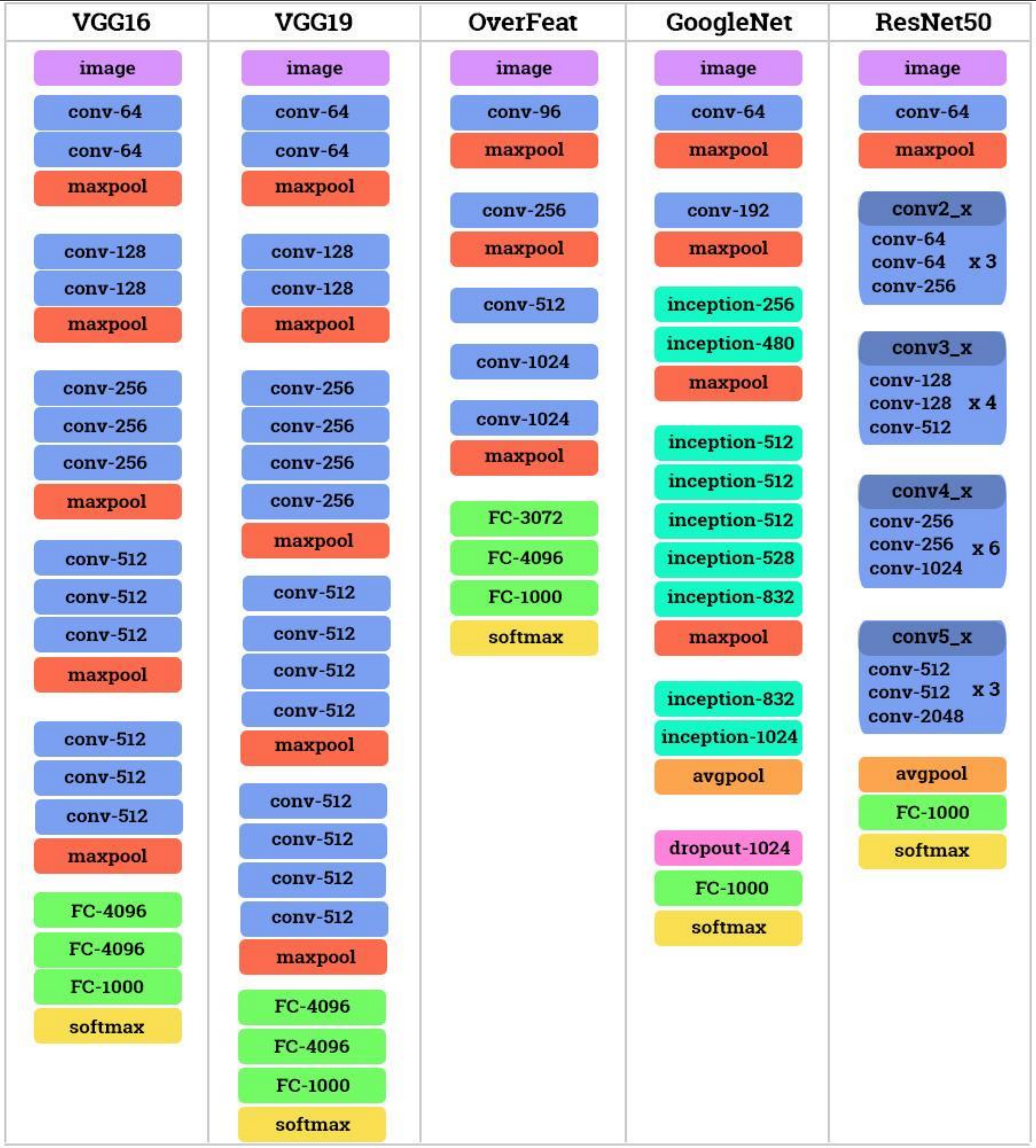
The object can be **oriented/rotated** in multiple dimensions with respect to how the object is photographed and captured.

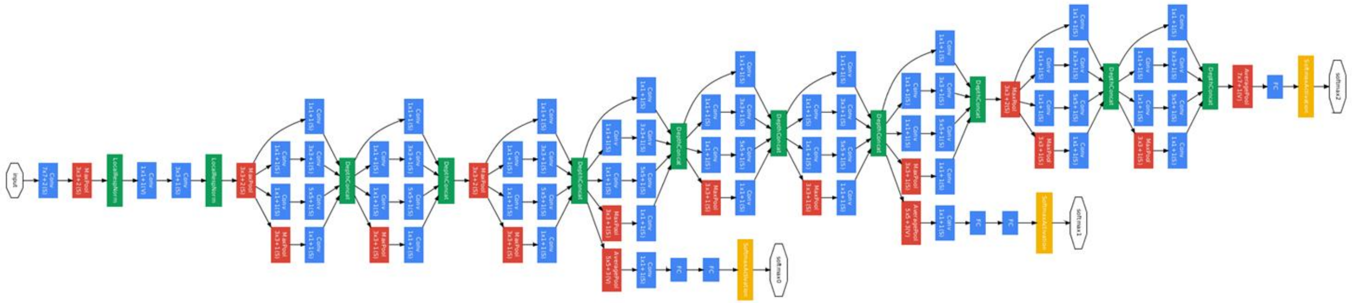
The image on the left was photographed with standard **overhead lighting**. The image on the right was captured with very **little lighting**. We are still examining the same coffee cup — but based on the lighting conditions the cup looks dramatically different.

The same venti coffee will look dramatically different when it is **photographed up close** and when it is **captured farther way**.

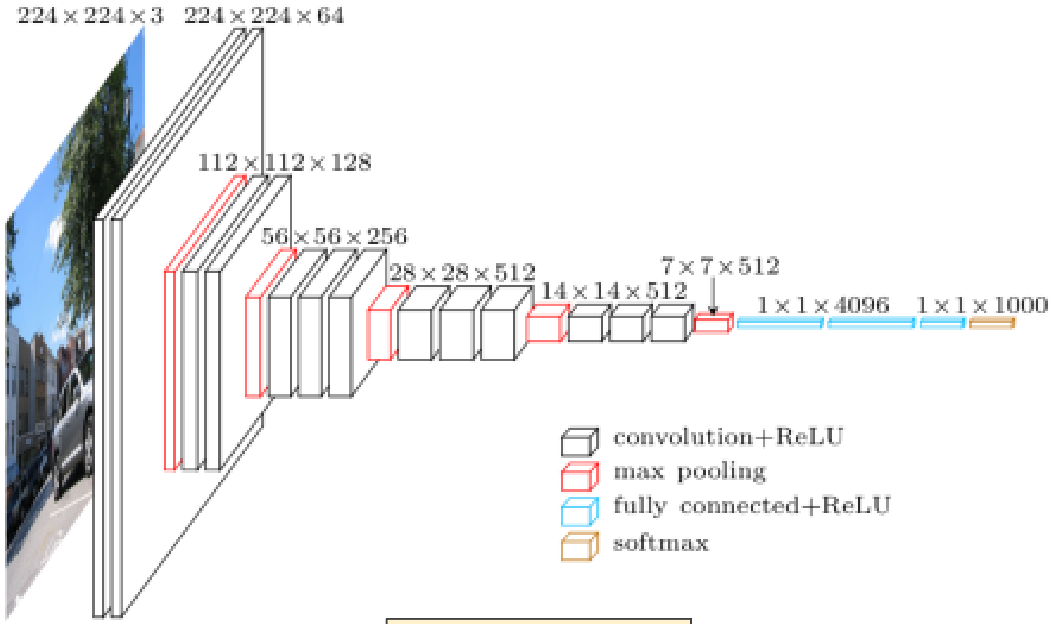
On the left we have a picture of a dog. The right we have a picture of the same dog, but notice how the dog is resting underneath the covers, **occluded from our view**.

STATE-OF-THE-ART CNN FOR IMAGE CLASSIFICATION





Google Inception-V3

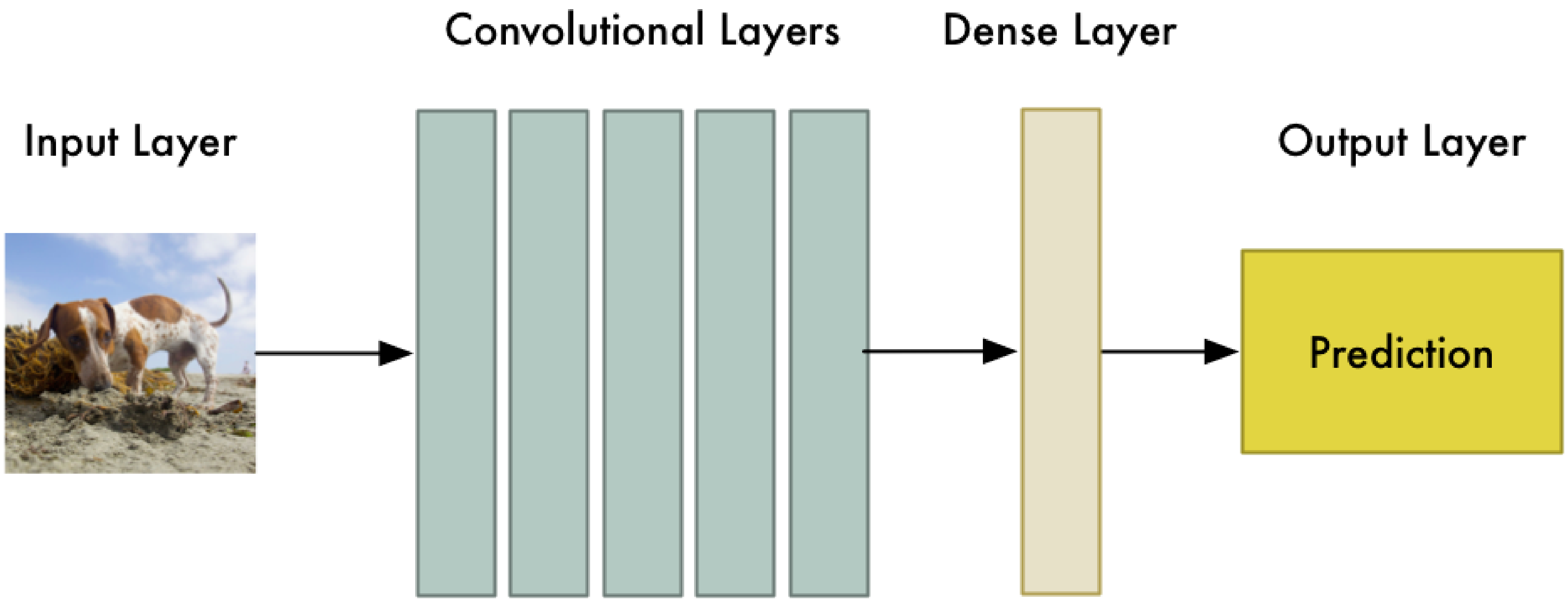


VGG-16

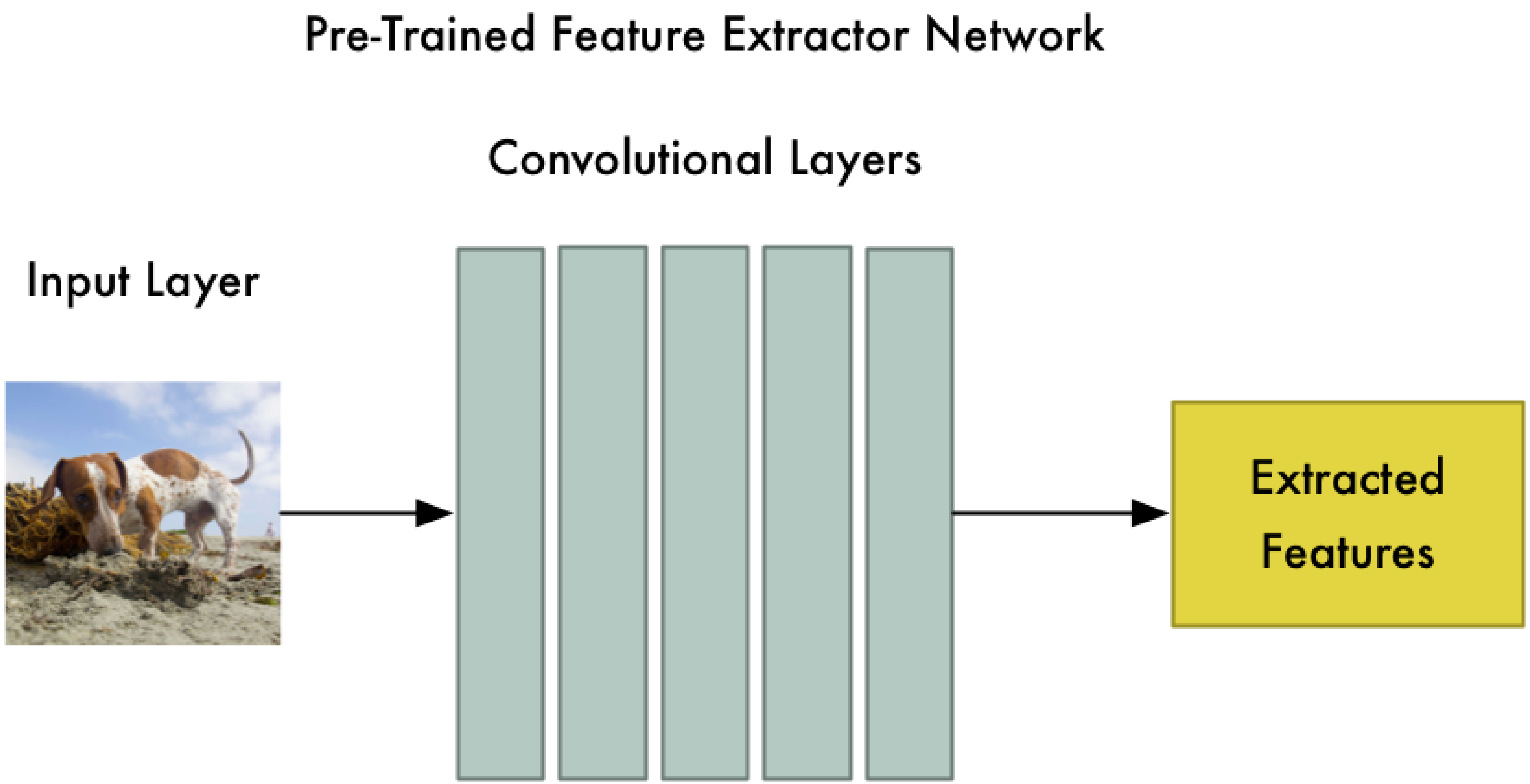
Transfer Learning

- Convolutional neural networks (CNNs) are great at image classification.
- But whenever we train a new convolutional neural network, it has to relearn how to classify images from scratch—which means that we need a massive amount of training data to make CNNs work well.
- In transfer learning, you take a neural network trained on one set of data and use what it has learned to give it a head start at solving a new problem.

Transfer Learning

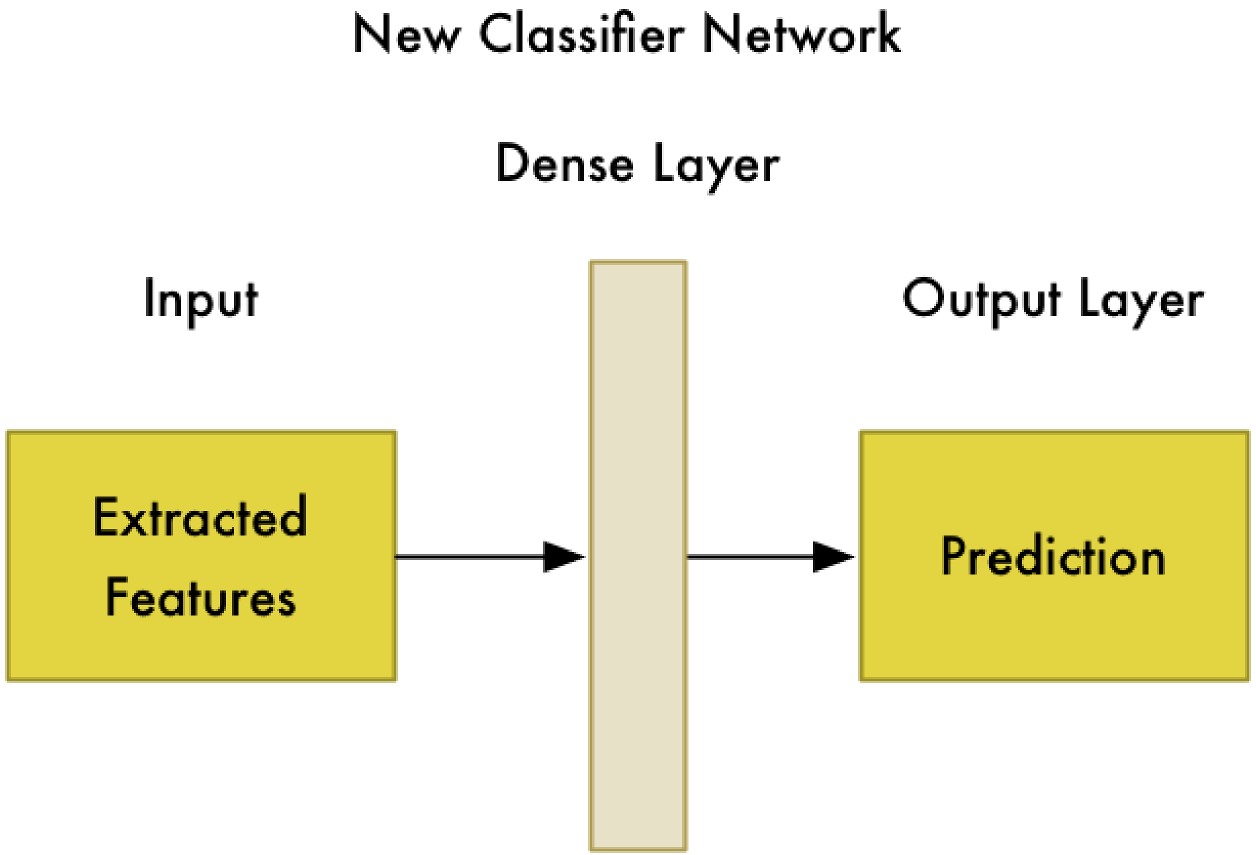


Transfer Learning

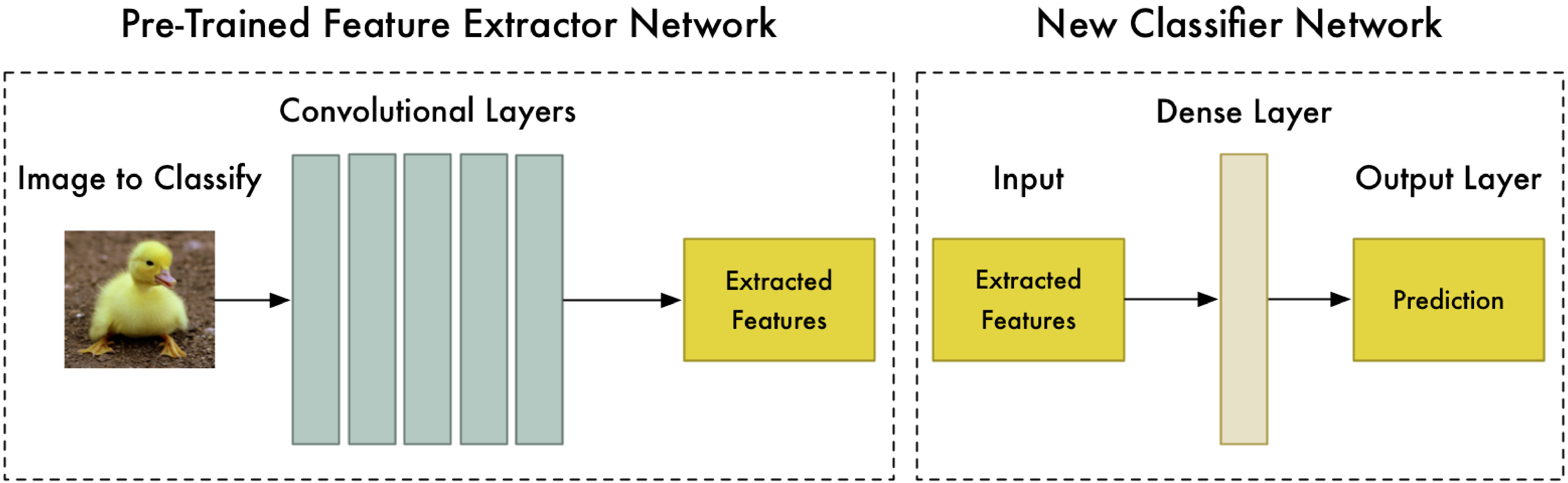


The neural network had to learn to detect all kinds of animal shapes that are probably also useful for detecting birds. We'll keep all the knowledge it has for detecting shapes.

Transfer Learning



Transfer Learning



WHY OBJECT DETECTION?

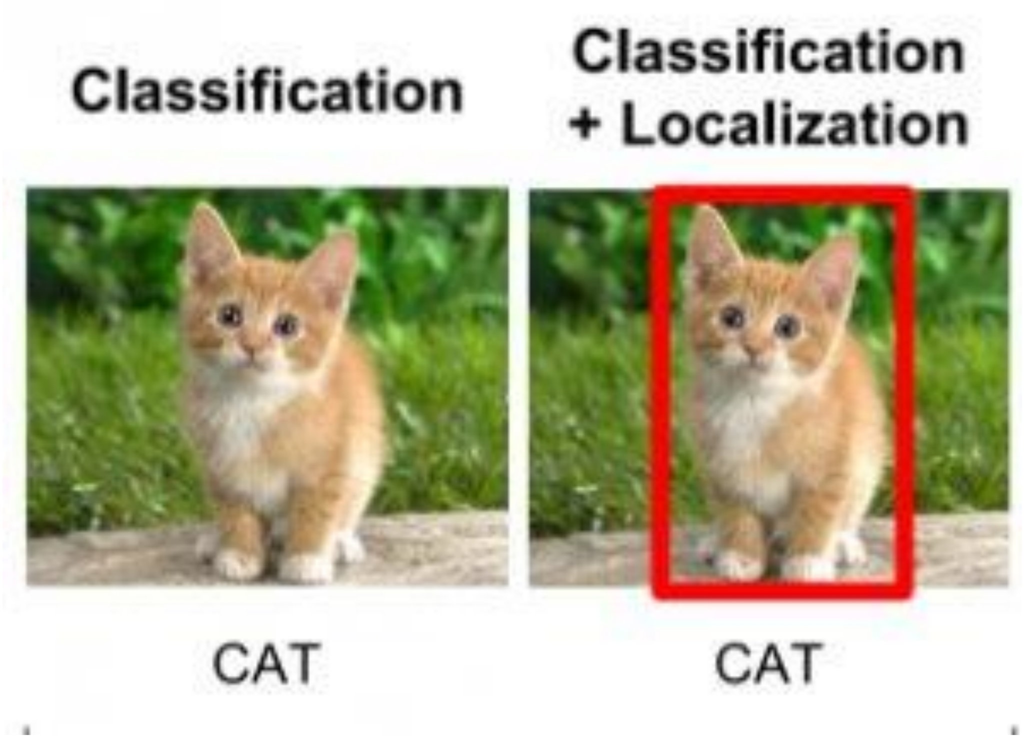
Image classification is the task of **assigning a label** to an image from a predefined set of categories.

To solve this problem, we can train a **multi-label classifier** which will predict the probabilities of both the classes (dog as well as cat).

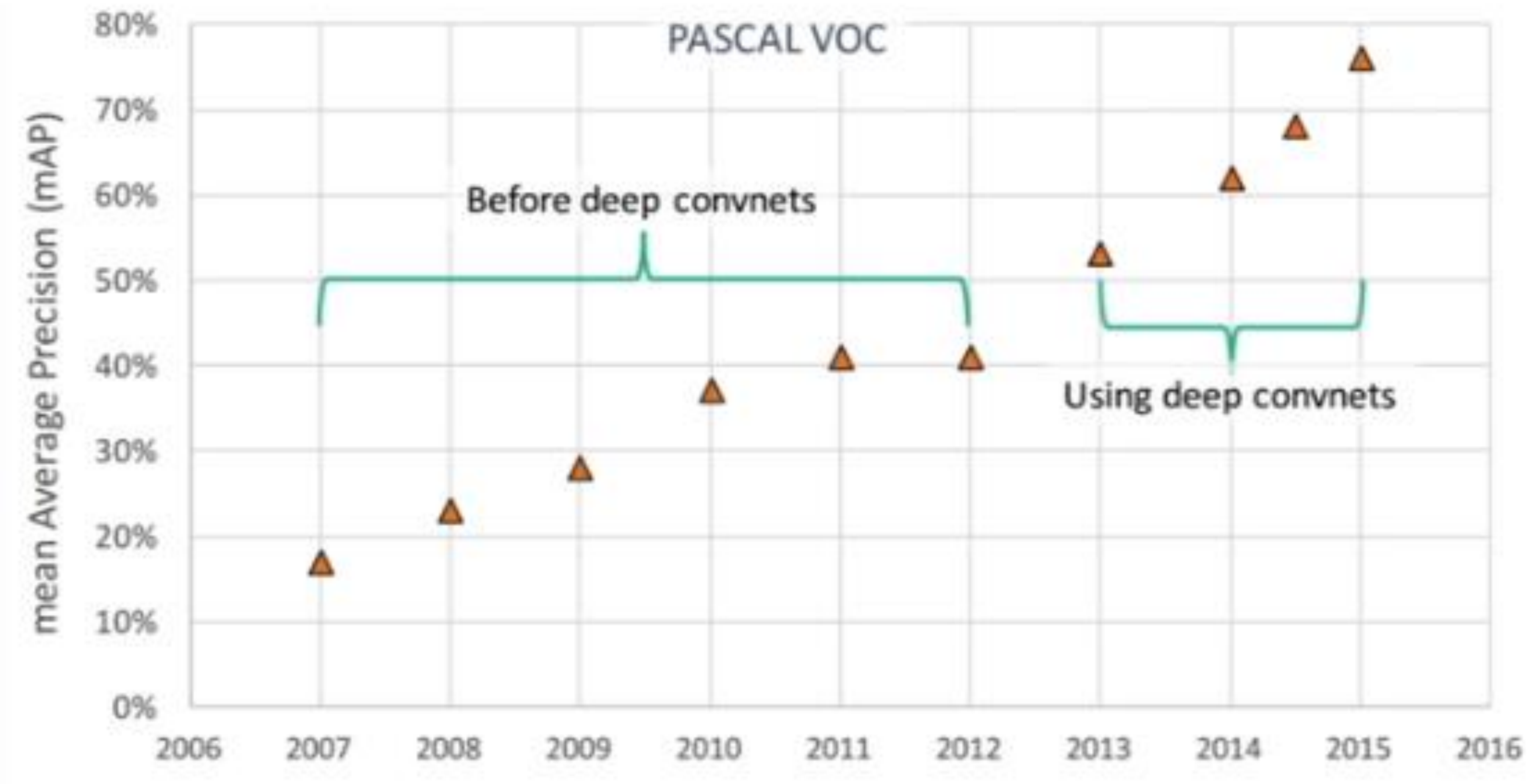
However, we still don't know the location of cat or dog in the image.

OBJECT DETECTION

Predicting the location of the object in an image or video is called **object detection** or **localization**.



DEEP LEARNING FOR OBJECT DETECTION

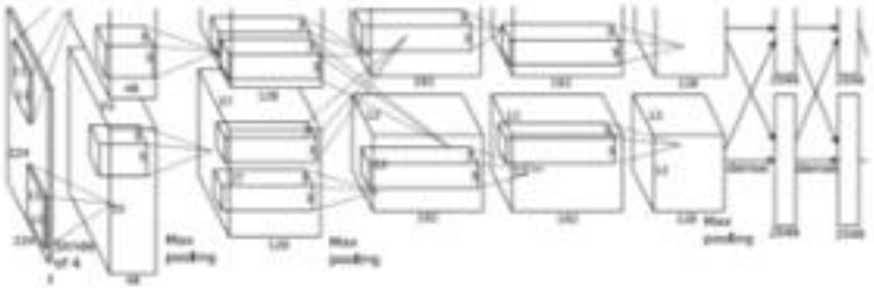


SLIDING WINDOWS - HOW IT WORKS?

- **Sliding window** is rectangular region of fixed width and height that "**slides**" across the image, from left-to-right and top-to-bottom.
- A sliding window slides from left-to-right and top-to-bottom across an input image taking **N pixel steps at a time**.
- The ROI at each step of the sliding window is **extracted and passed** into the feature extraction/object detection pipeline.

OBJECT DETECTION USING CNN

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? NO
Background? YES

REGION PROPOSALS

- Find “**blobby**” image regions that are **likely to contain objects**.
- Relatively fast to run; e.g. **Selective search** gives 1000 region proposals in a few seconds on CPU.

HISTORY OF OBJECT DETECTION

While there are many object detection methods in the computer vision literature, **two stand out amongst the others**:

- HOG + Linear SVM (Histogram of Oriented Gradients + Linear Support Vector Machine)
- Haar cascades

HISTOGRAM OF ORIENTED GRADIENTS (HOG)

- Normalizing the image prior to description.
- Computing gradients in both the x & y directions.
- Obtaining weighted votes in spatial & orientation cells.
- Contrast normalizing in the overlapping spatial cells.
- Collect all HOGs to form the final feature vector.

GRADIENT COMPUTATION

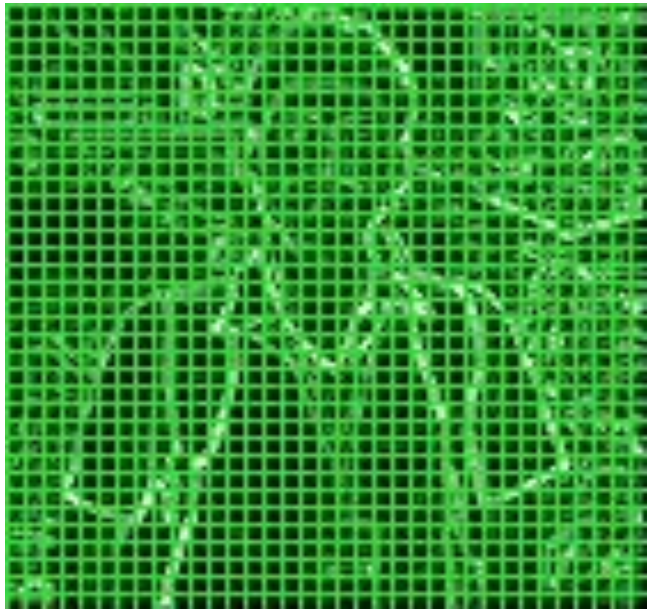
$$G_x = I \star D_x$$

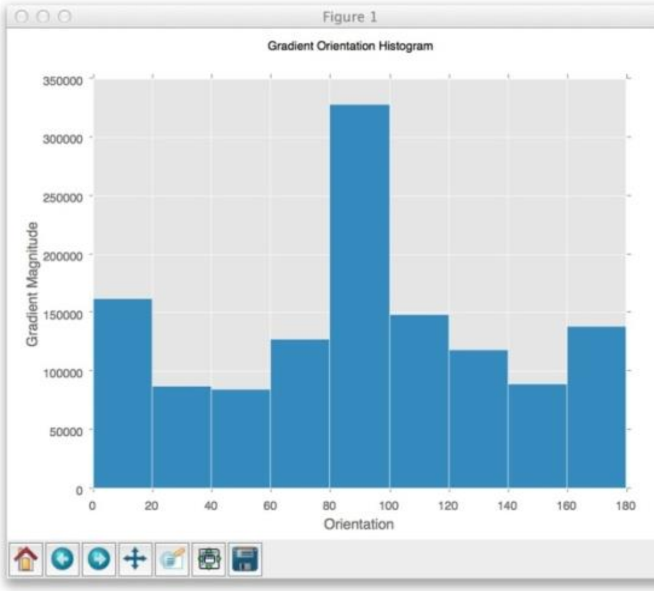
$$G_y = I \star D_y$$

$$|G| = \sqrt{G_x^2 + G_y^2}$$

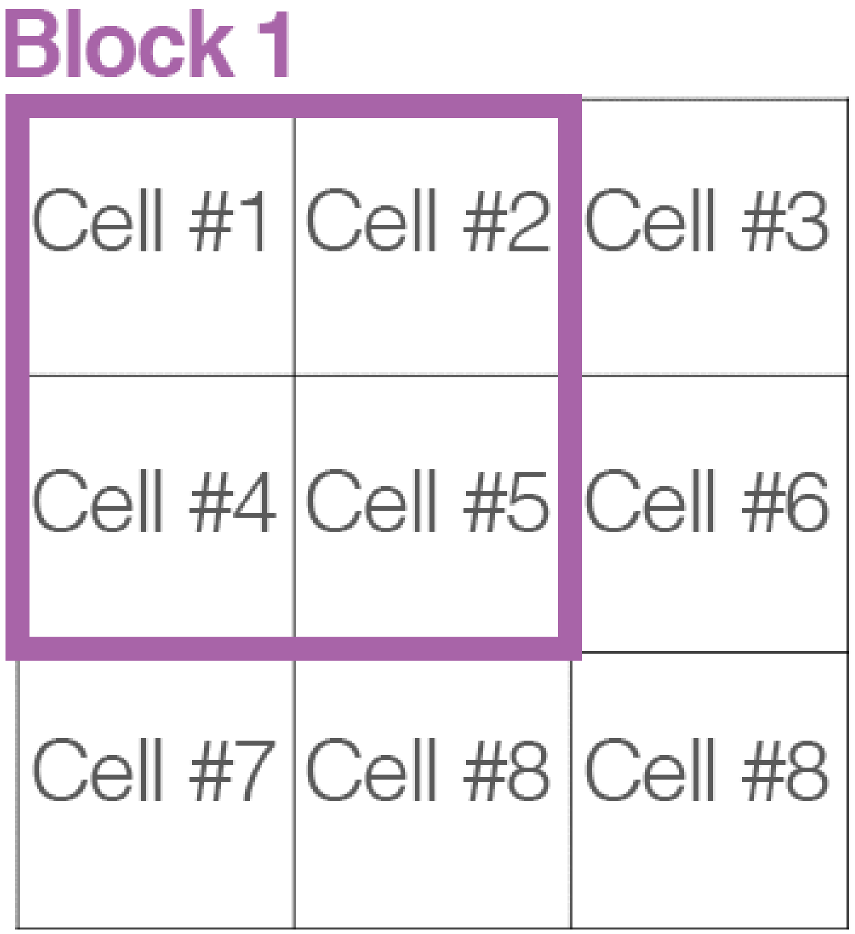
$$\theta = tan^{-1}(\frac{G_y}{G_x})$$

WEIGHTED VOTES IN EACH CELL

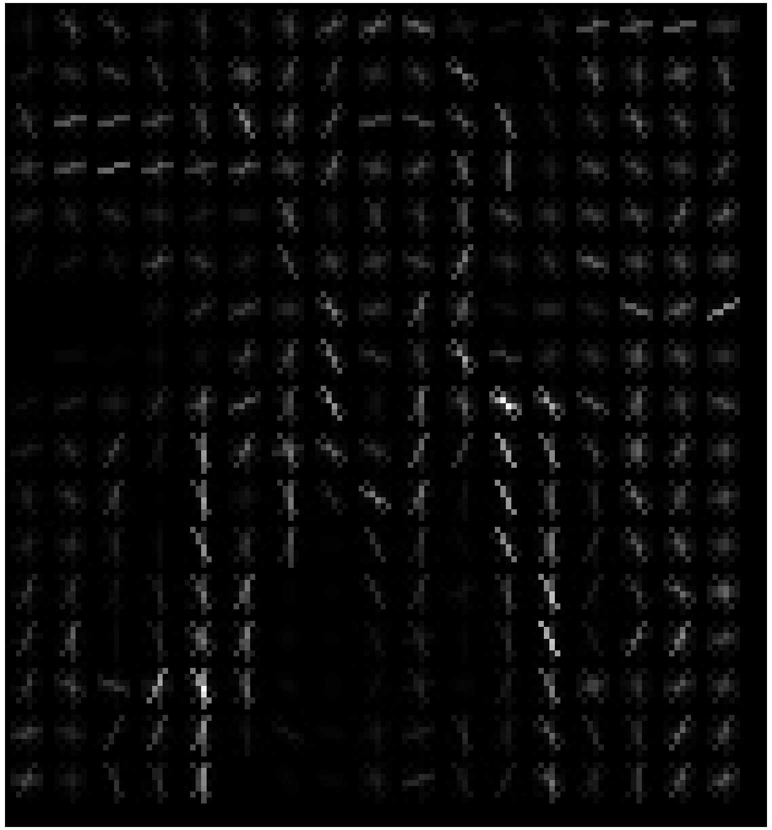




CONTRAST NORMALIZATION OVER BLOCK



HOG FEATURE VECTOR



HAAR CASCADES

- One of the most famous object detectors, **Rapid Object Detection using a Boosted Cascade of Simple Features**, by Viola and Jones (2004).
- **Pre-trained Haar cascades** are distributed with the OpenCV library, and are arguably the most used models for **face detection**.
- While Haar cascades are fast, they -

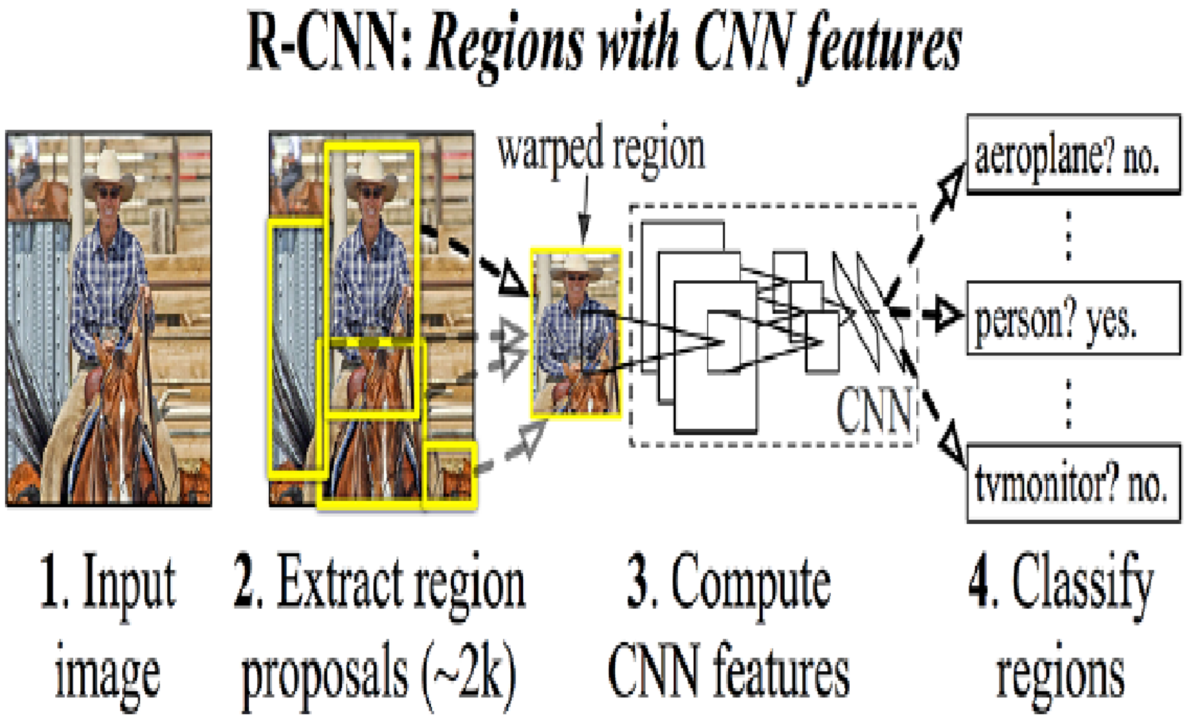
  a. Tend to have a high false-positive detection rate.

  b. Can miss objects entirely based on the parameters supplied to the cascade.

REGION-BASED CONVOLUTIONAL NEURAL NETWORK (R-CNN)

- R-CNN solves this problem by using an object proposal algorithm called **Selective Search** which reduces the number of bounding boxes that are fed to the classifier close to 2000 region proposals.
- Selective search uses **local cues** like texture, intensity, color to generate all the possible locations of the object.
- There are **3 important parts** in R-CNN :
    a. Run Selective Search to generate probable objects.
    b. Feed these patches to CNN, followed by SVM to predict the class of each patch.
    c. Optimize patches by training bounding box regression separately.

R-CNN ARCHITECTURE



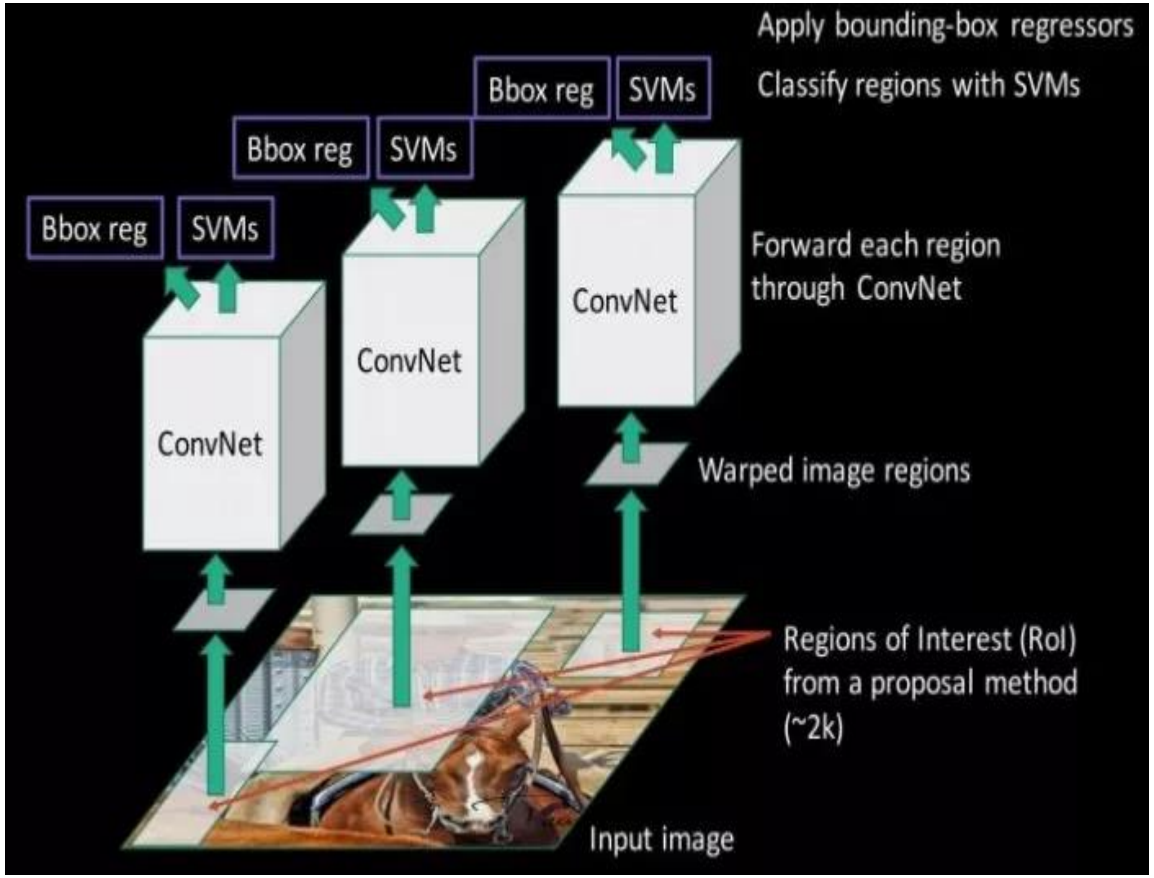
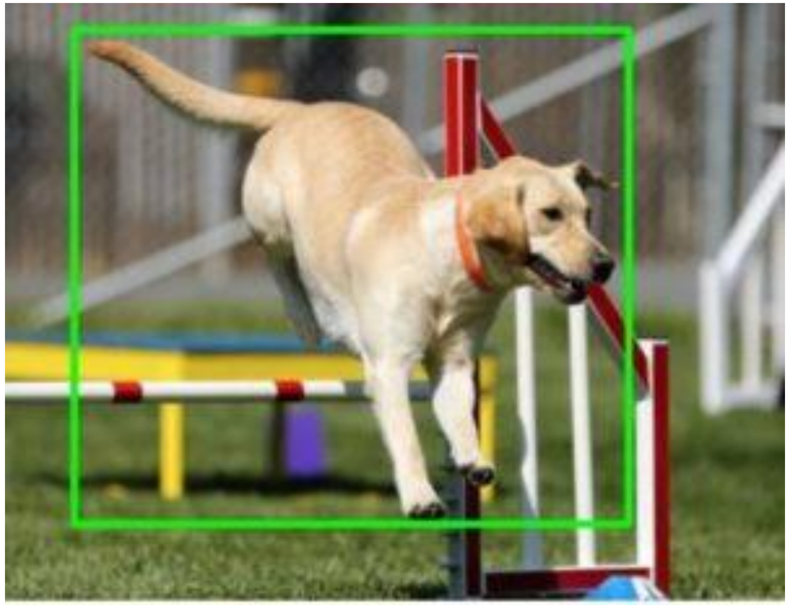


IMAGE PYRAMIDS

- An **image pyramid** is simply a **multi-scale representation** of an image. Using an image pyramid allows us to find objects in images at **different scales** of an image.
- A **sliding window** requires fixed spatial dimensions. If the object in the window is too large or small for the sliding window size, we can miss the detection.



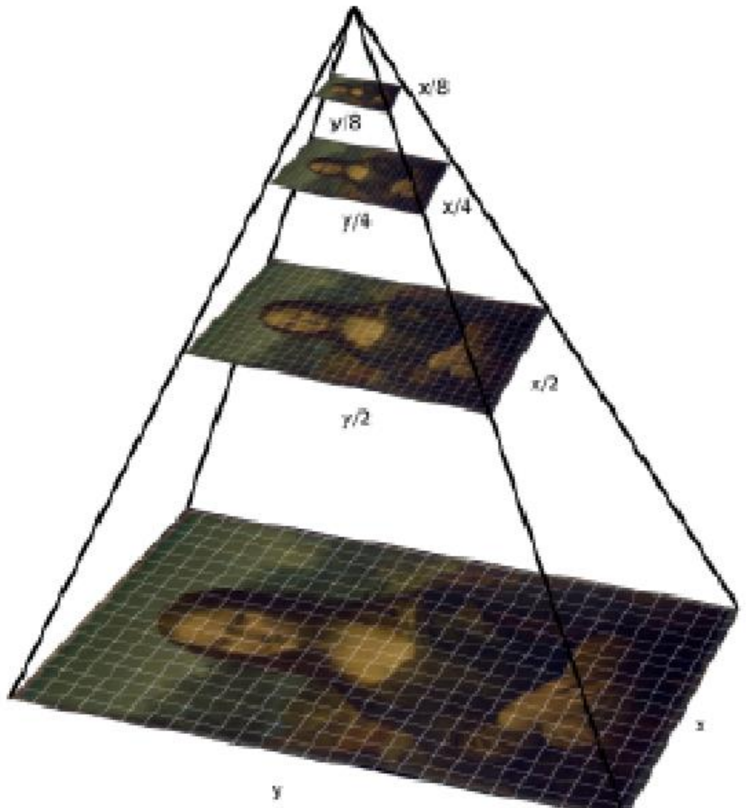
OBJECT DETECTION PIPELINE

Step 1: Scan image at all scales and locations.

Step 2: Extract features over each sliding window location.

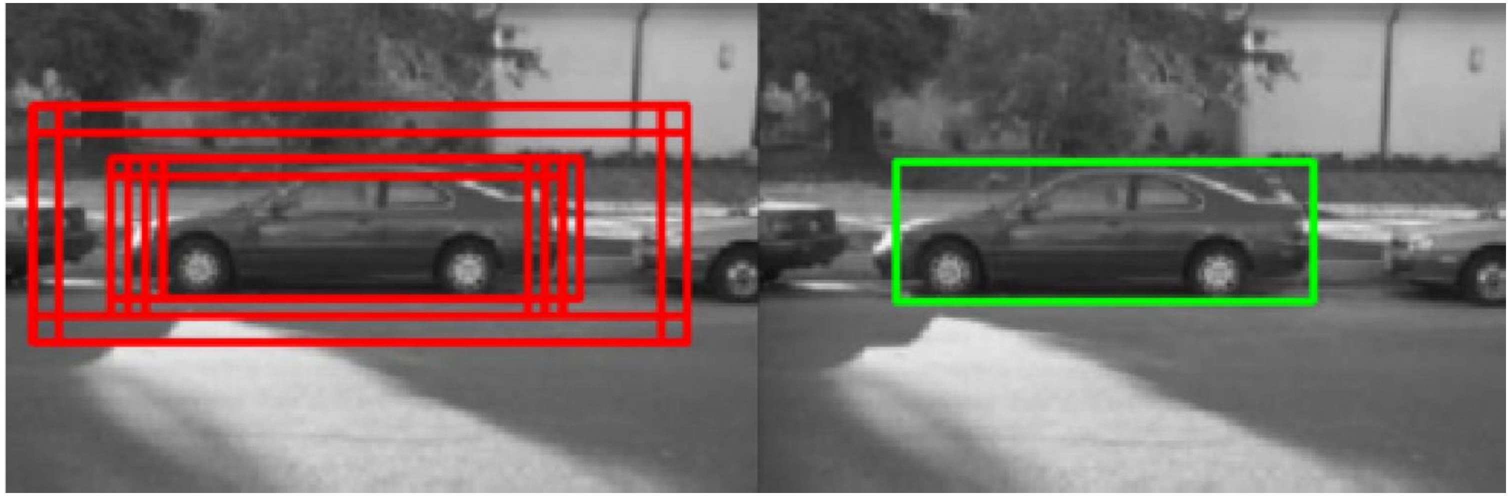
Step 3: Using Linear SVM to classify features extracted from each window.

Step 4: Apply non-maxima suppression to obtain final bounding boxes.

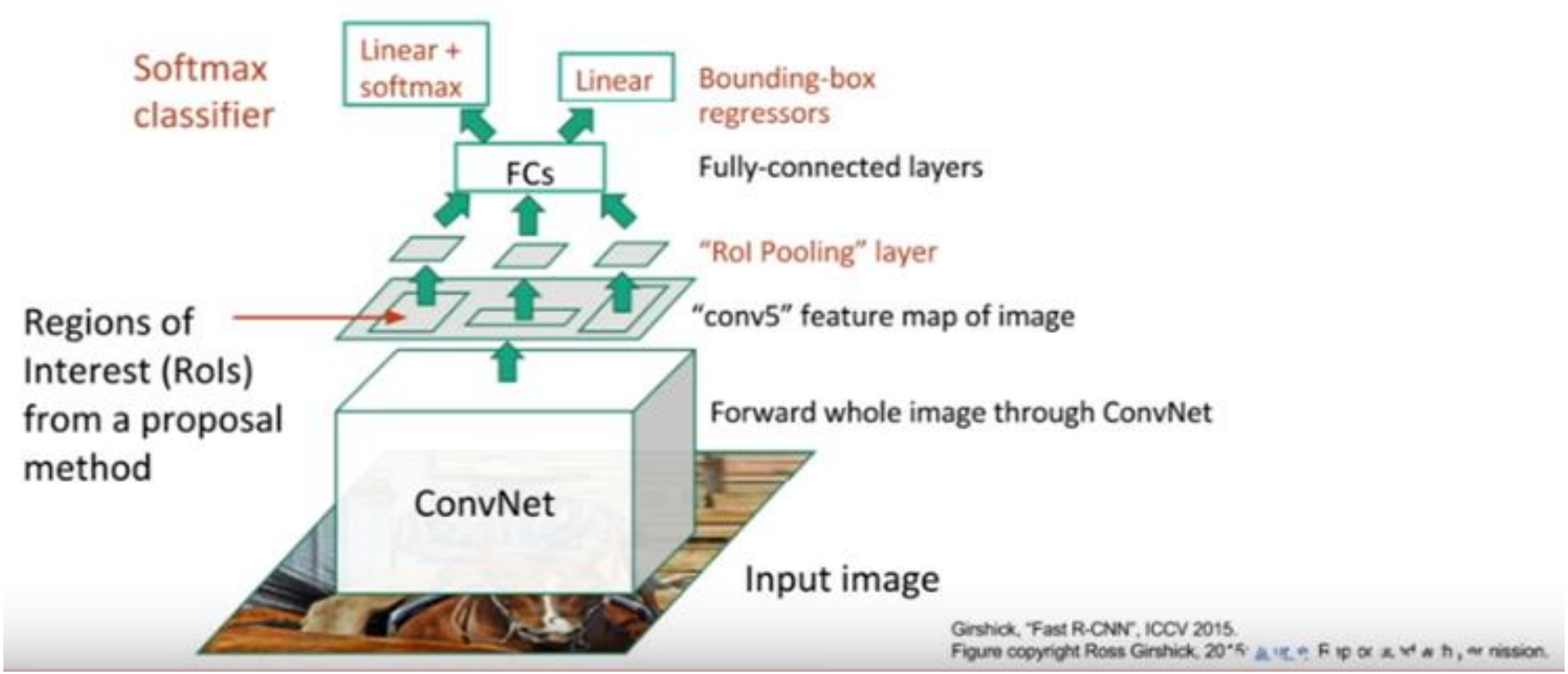


NON-MAXIMA SUPPRESSION

- When combined with image pyramids, this behavior implies that we will have **multiple bounding boxes** surrounding the object at multiple scales, even though there may only be one “true” object in the image.
- To handle the removal of overlapping bounding boxes (that refer to the same object) we can apply **non-maxima suppression (NMS)**.
- **NMS** works by computing the ratio of overlap between bounding boxes, then suppressing (i.e., removing) bounding boxes that have significant overlap.



FAST R-CNN


Softmax classifier
Linear + softmax
Linear
Bounding-box regressors
FCs
Fully-connected layers
"RoI Pooling" layer
"conv5" feature map of image
Regions of Interest (RoIs) from a proposal method
Forward whole image through ConvNet
ConvNet
Input image
Girshick, "Fast R-CNN", ICCV 2015.
Figure copyright Ross Girshick

SINGLE-SHOT DETECTOR (SSD)

- **SSD** architecture builds on the **VGG-16** architecture by removing the fully connected layers.
- A set of *auxiliary* convolutional layers are added for extracting the features at multiple scales and progressively decrease the size of the input to each subsequent layer.

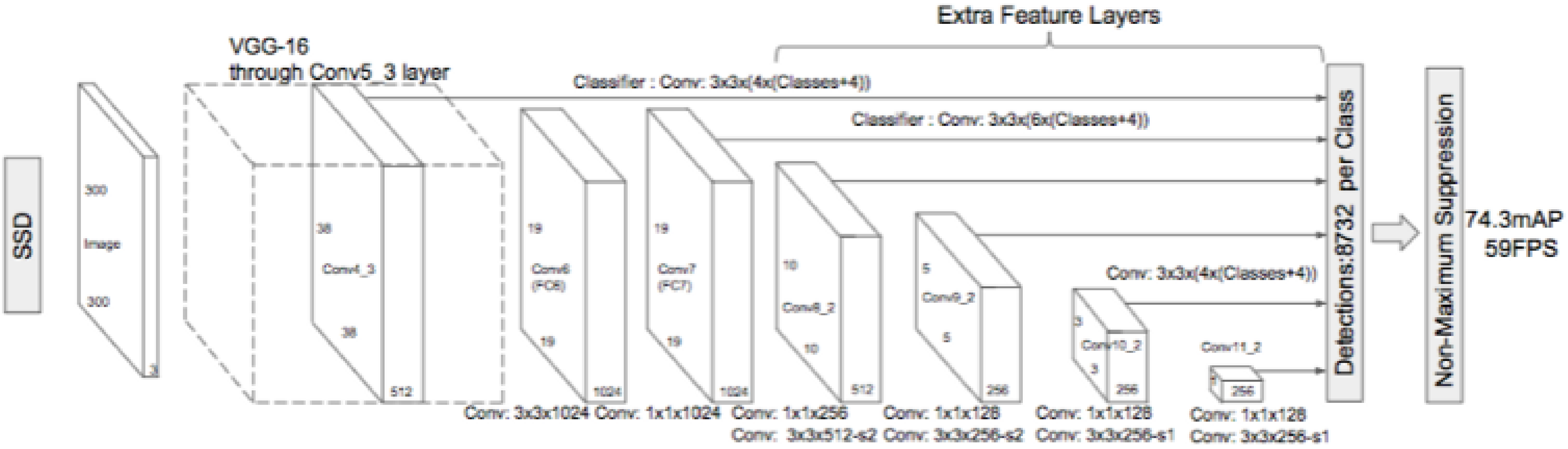


Image Segmentation

We’re not only labeling objects in a picture, but we are locating their position in the picture and finding which exact pixels belong to each object.
Image segmentation systems can trace out each object in a picture, you can use them to count objects. For example, you can use image segmentation to count the number of people waiting in a line.

Image Segmentation - Applications

Image Segmentation - Applications

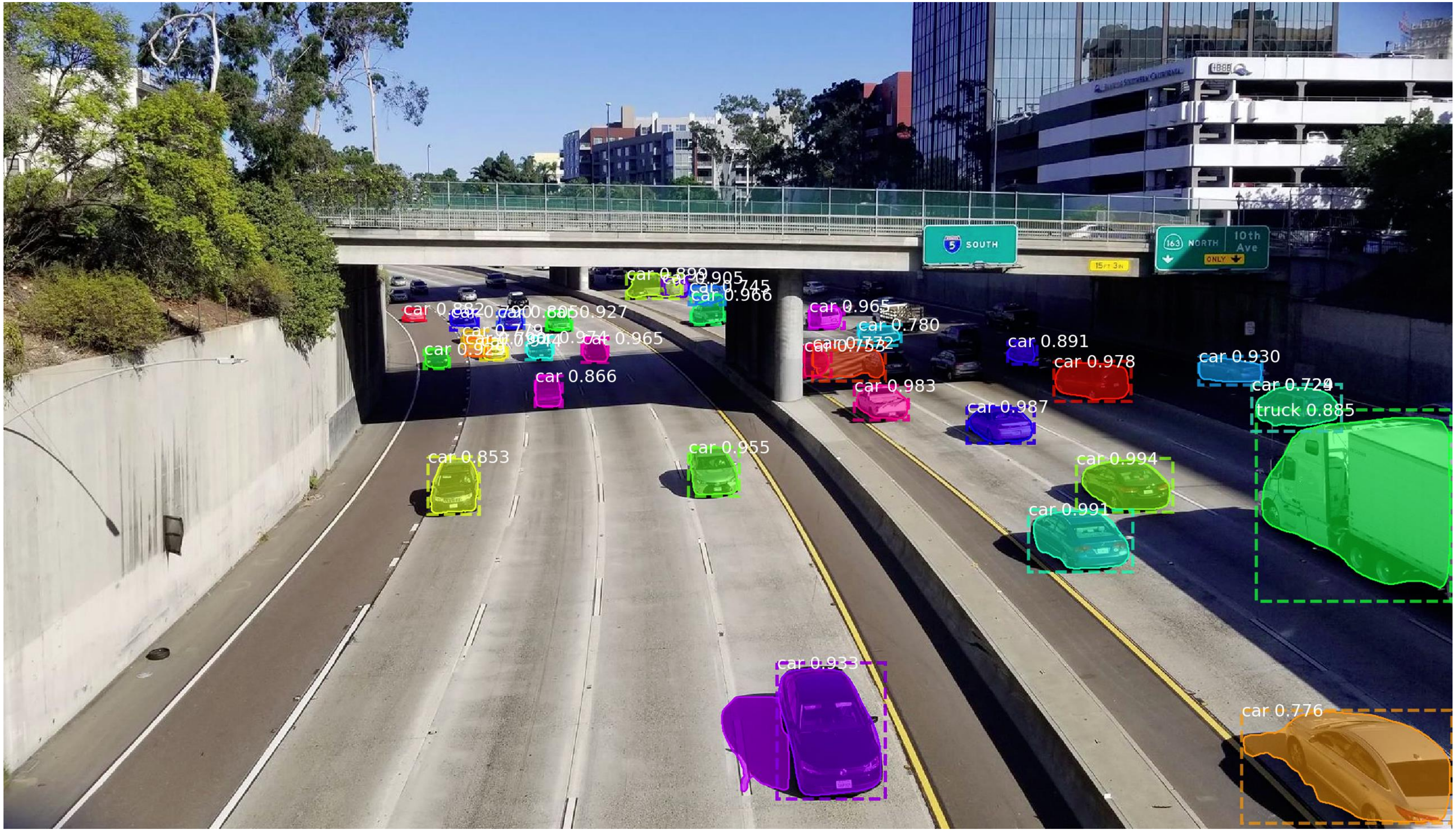


Image Segmentation - Applications

Image segmentation can detect different types of objects and draw their boundaries, it's a natural fit for helping to create maps from satellite imagery.

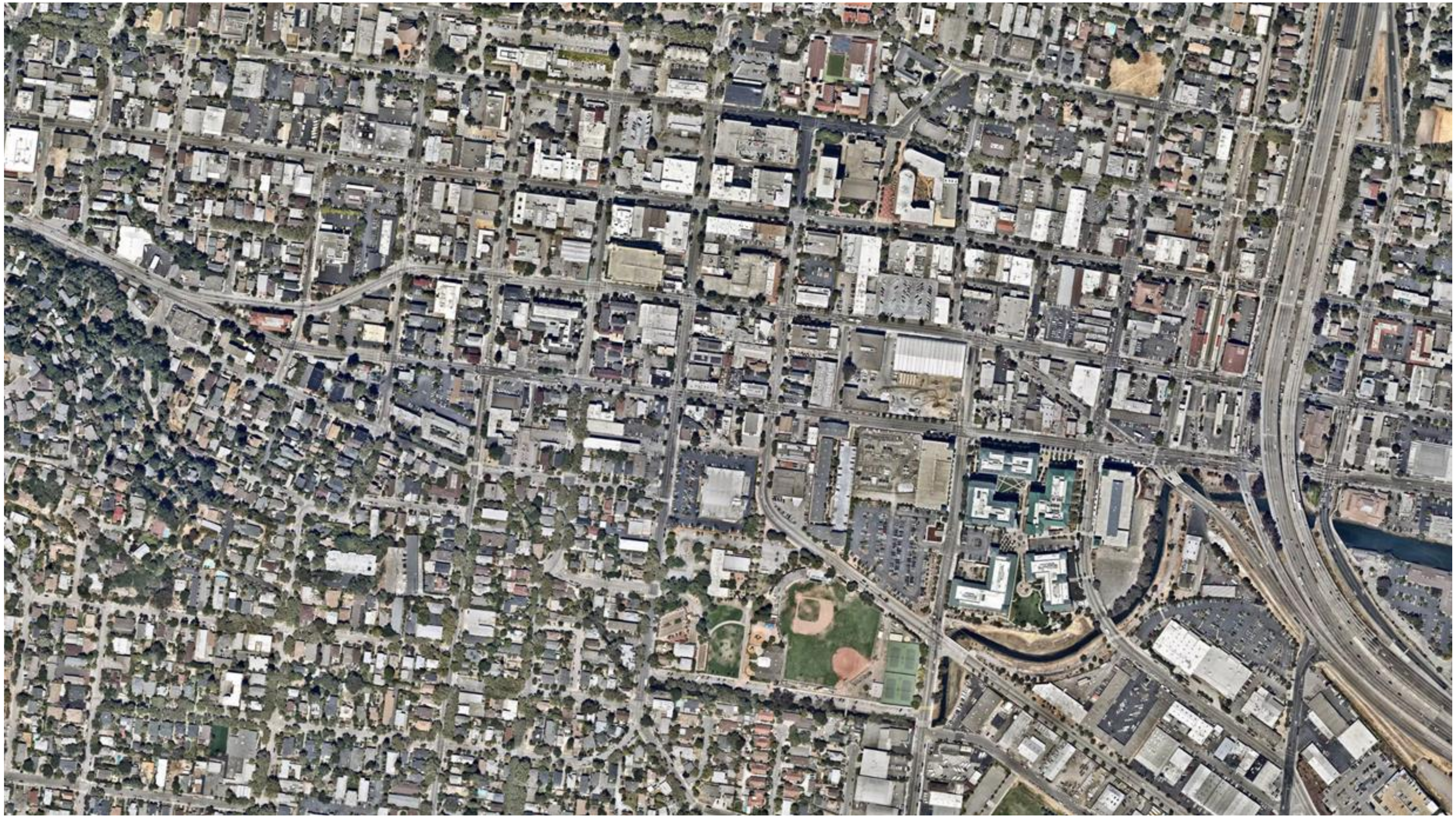
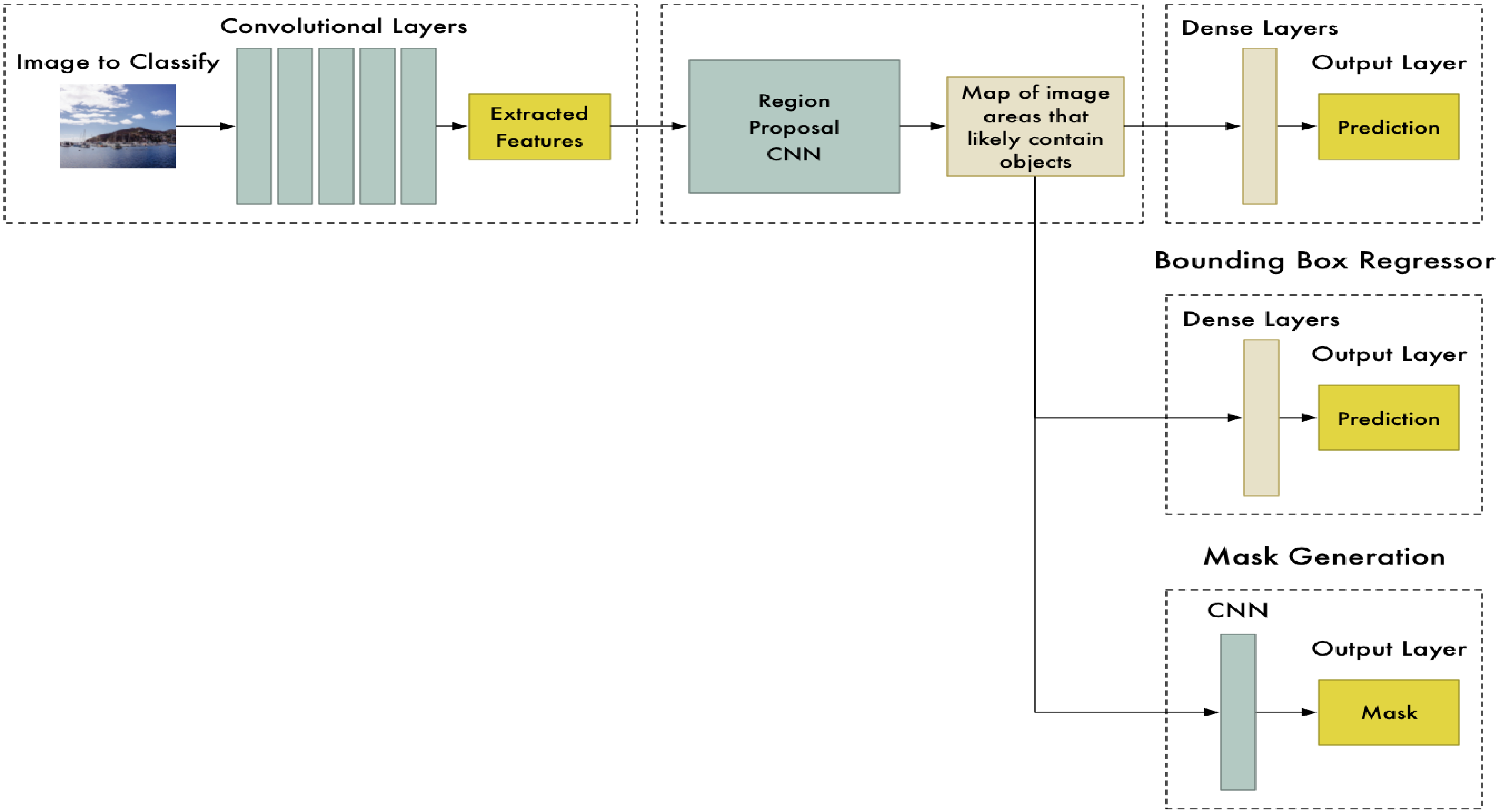


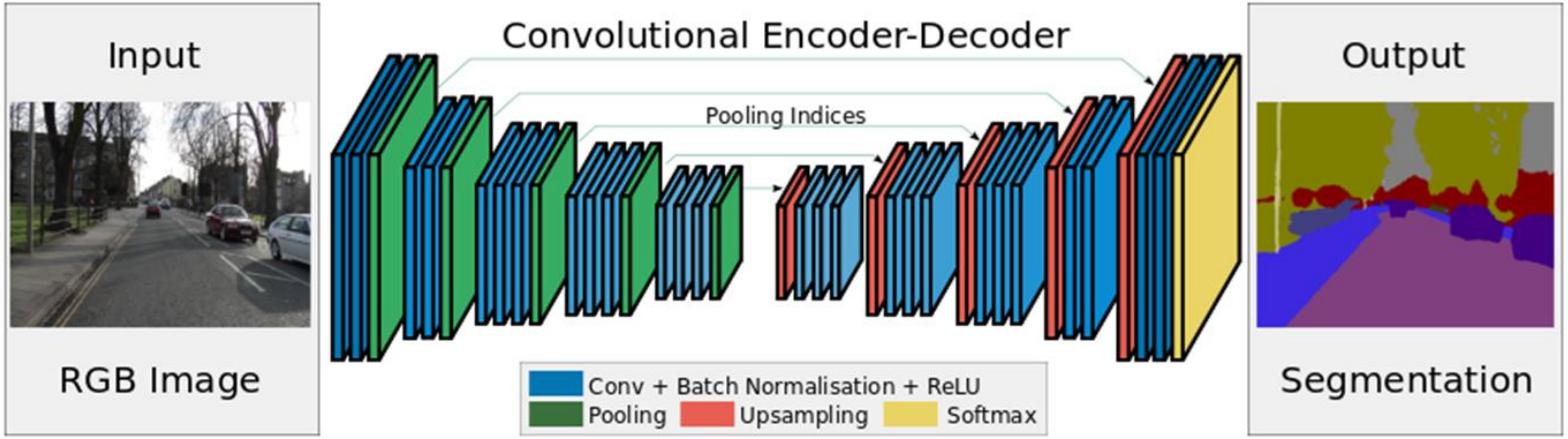
Image Segmentation - Architecture



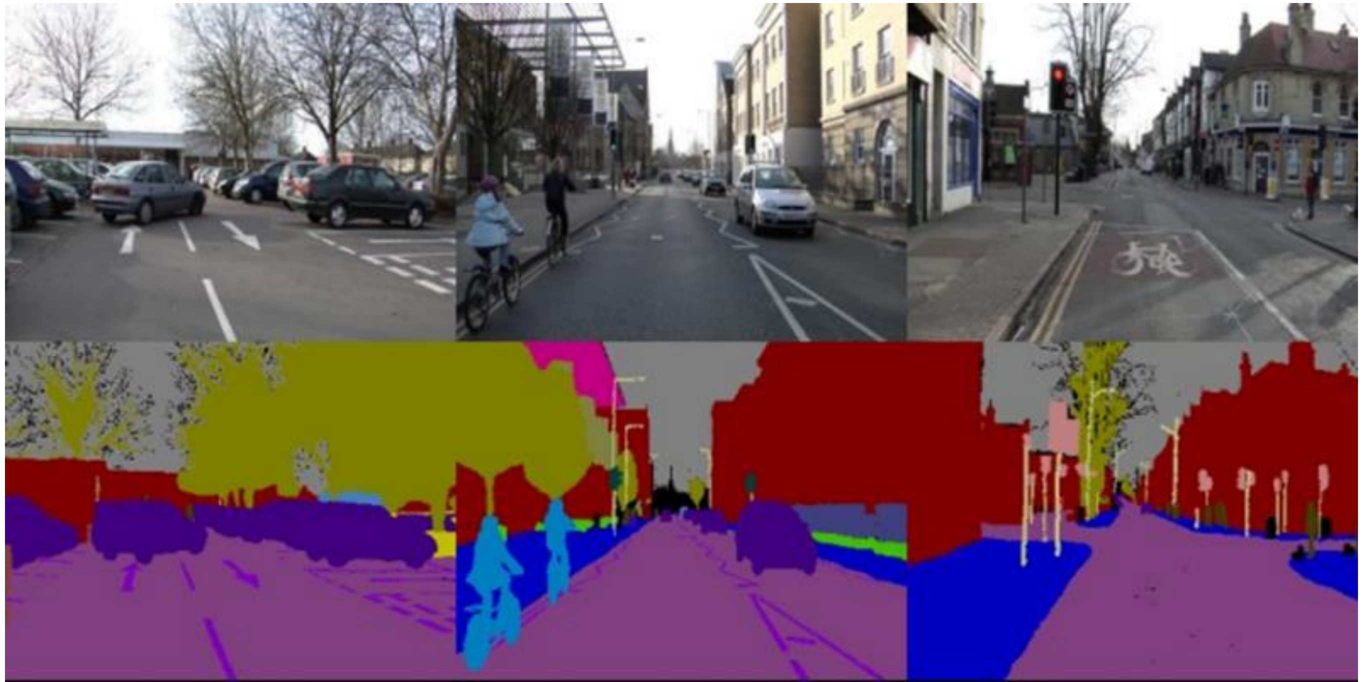
CNN FOR IMAGE SEGMENTATION

- Pixel level semantic segmentation of a road scene understanding.
- **Assign a label to every pixel in an image.**
- Object detection, medical imaging, machine vision, traffic control systems.

SEGNET



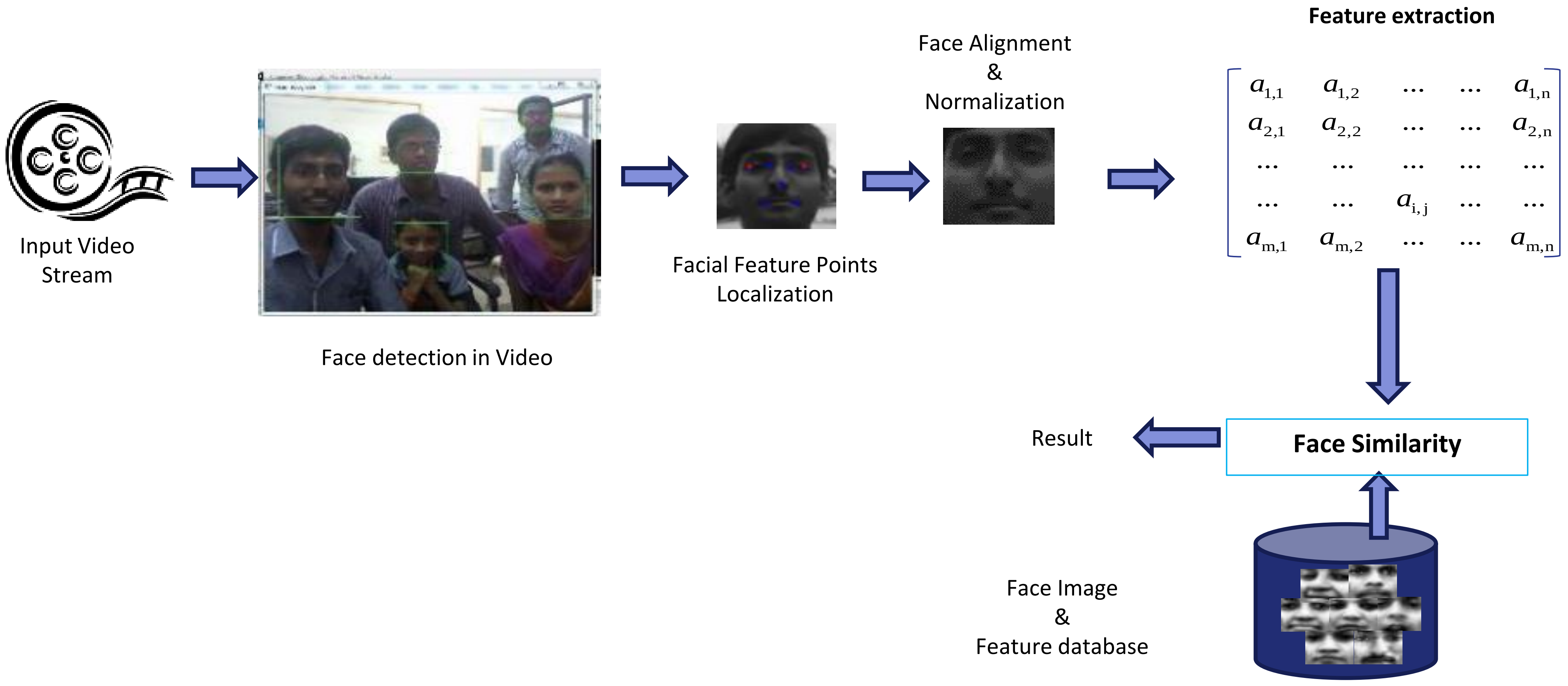
- Deep encoder-decoder architecture for multi-class pixelwise segmentation .
- Keras and Caffe library implementation.
- Indoor and outdoor scene understanding.
- 11 classes .



Demo

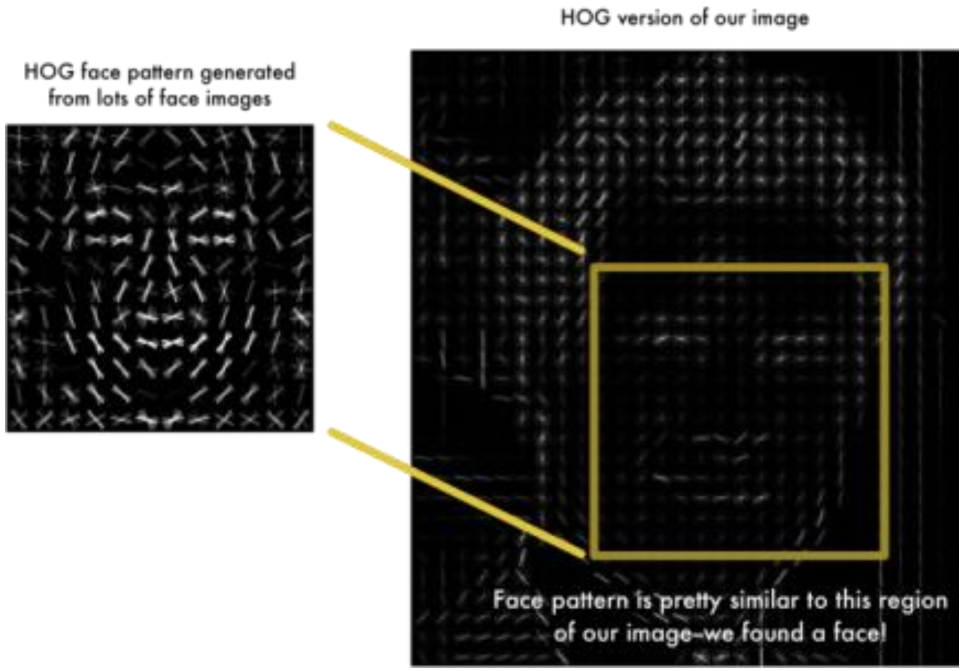
Face Recognition System using Deep learning

Face Recognition System (FRS) automatically identifies a person in a input video stream using source images that are previously stored in the database.

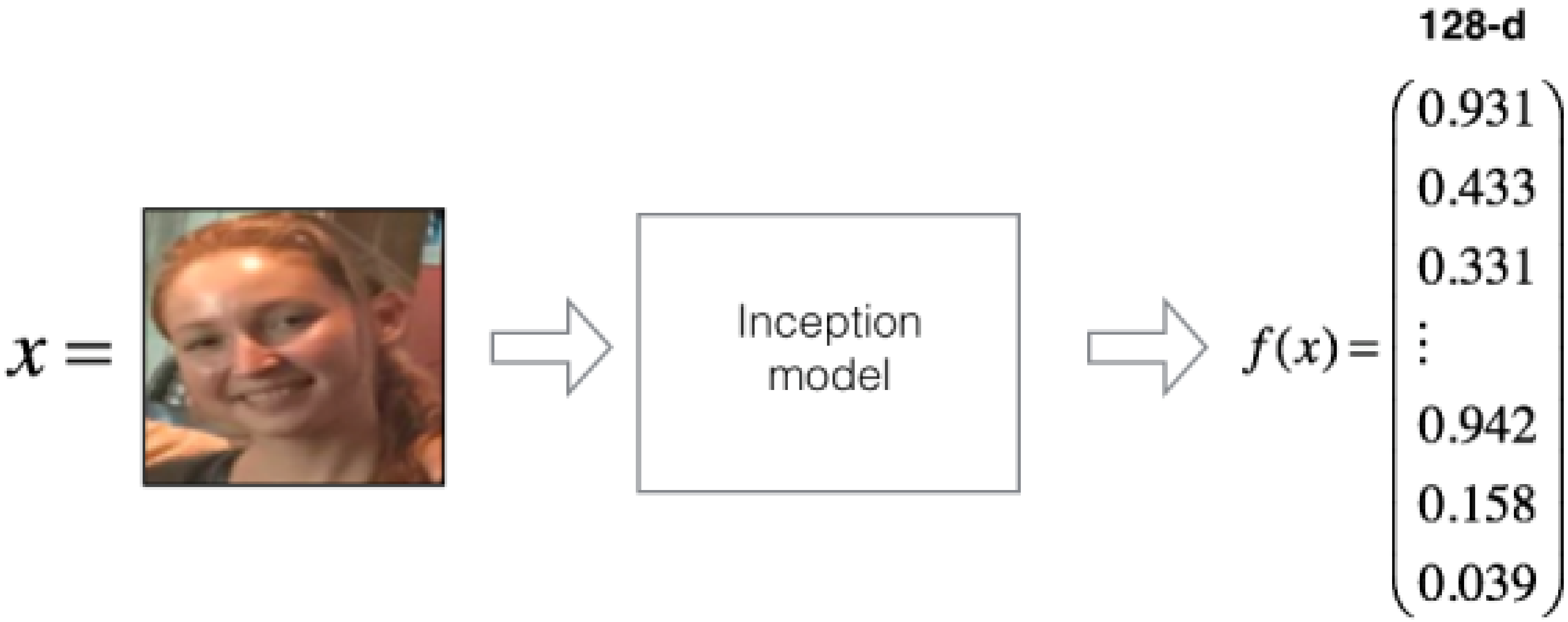


Face Detection and Alignment

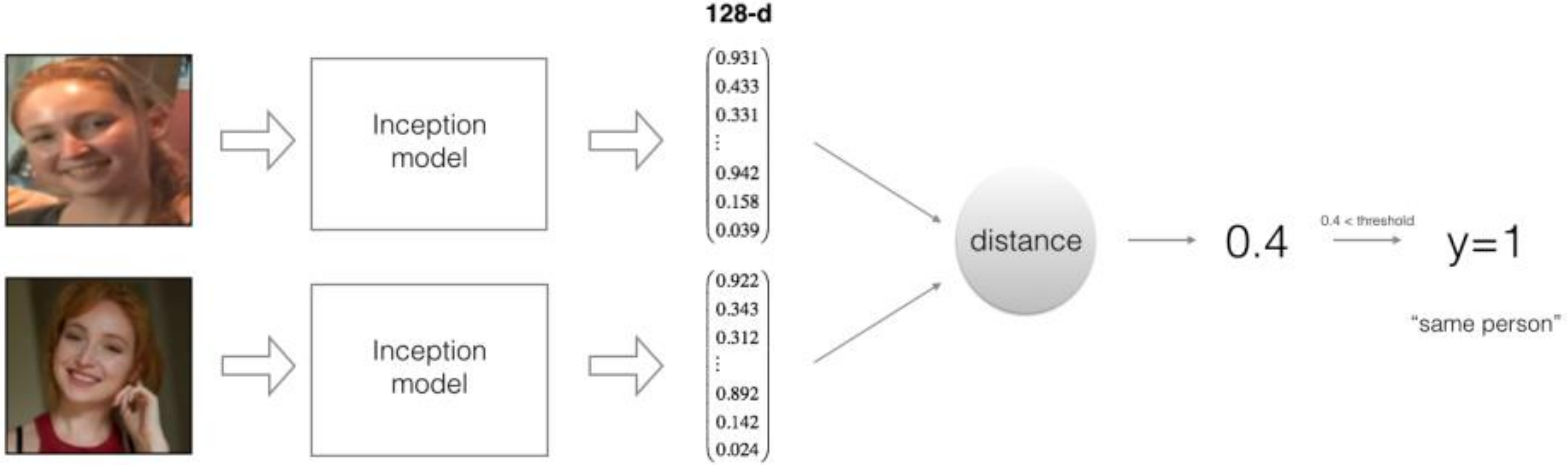
- HOG + Linear SVM based face detection is used to detect the faces in each frame.
- For each detected face image localize 68 (x, y) coordinates that map to facial structures on the face.
- The angle of the face image is computed by taking the difference between left and right eye centers.
- The face image is rotated based on calculated angle value by Affine transformation.

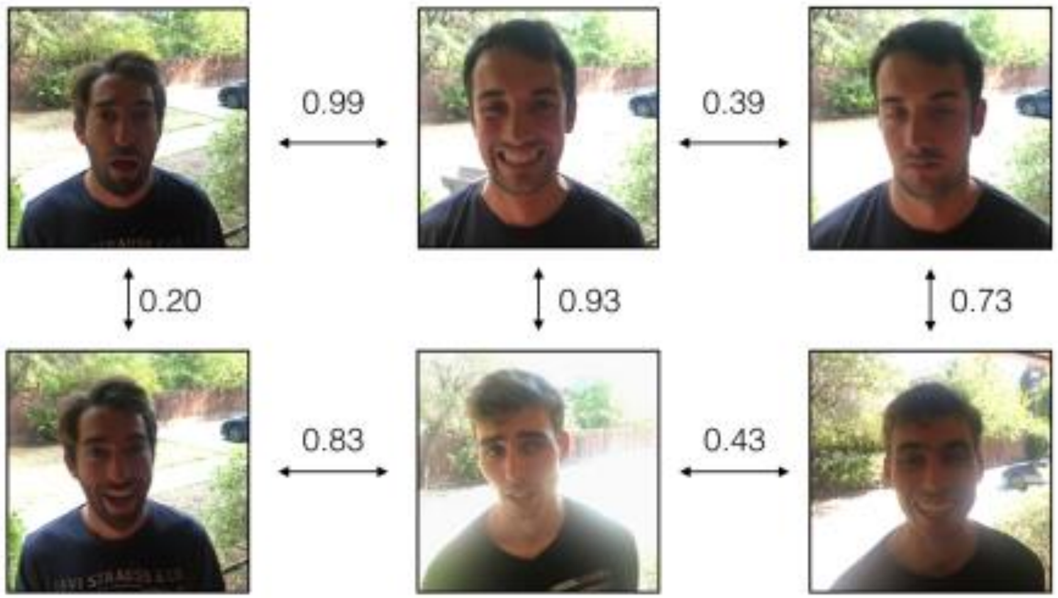


Feature Vector Representation

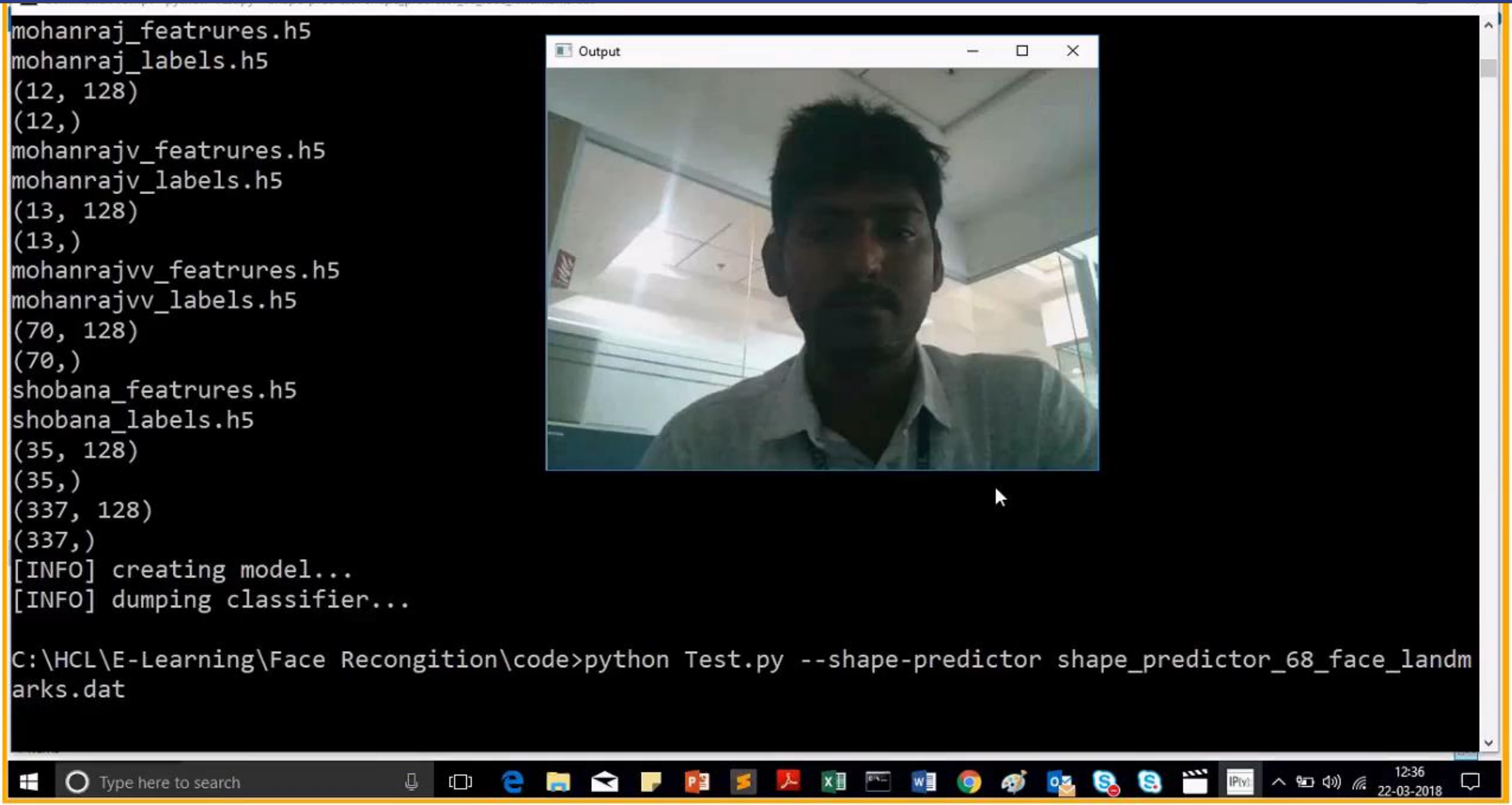


Feature Matching



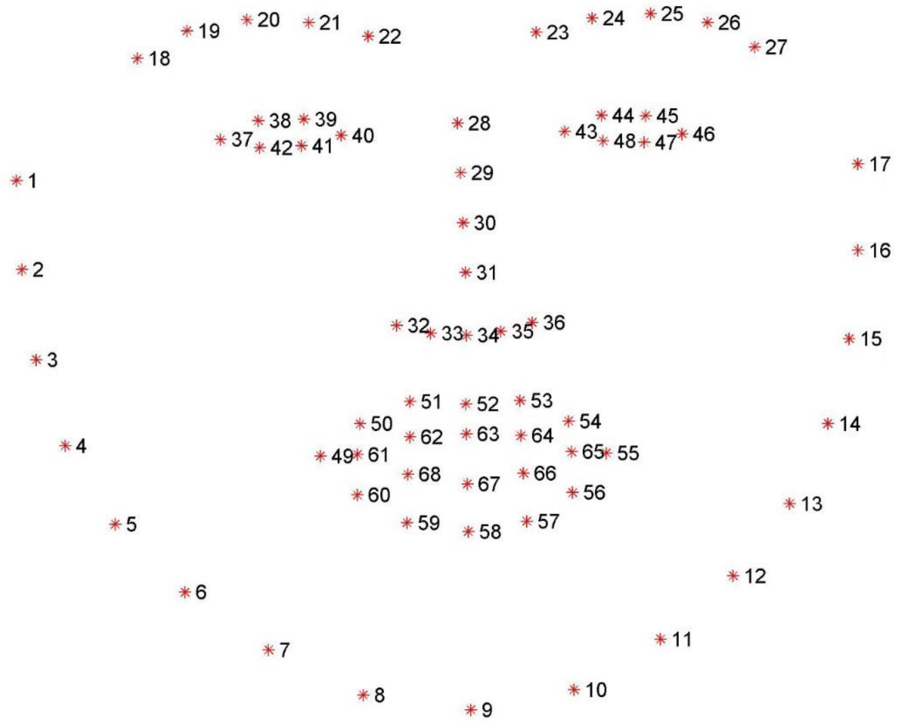


Face Recognition Demo



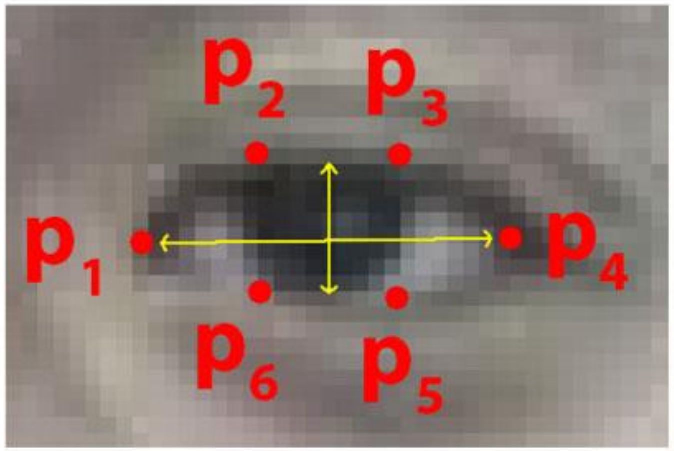
Driver Drowsiness detection

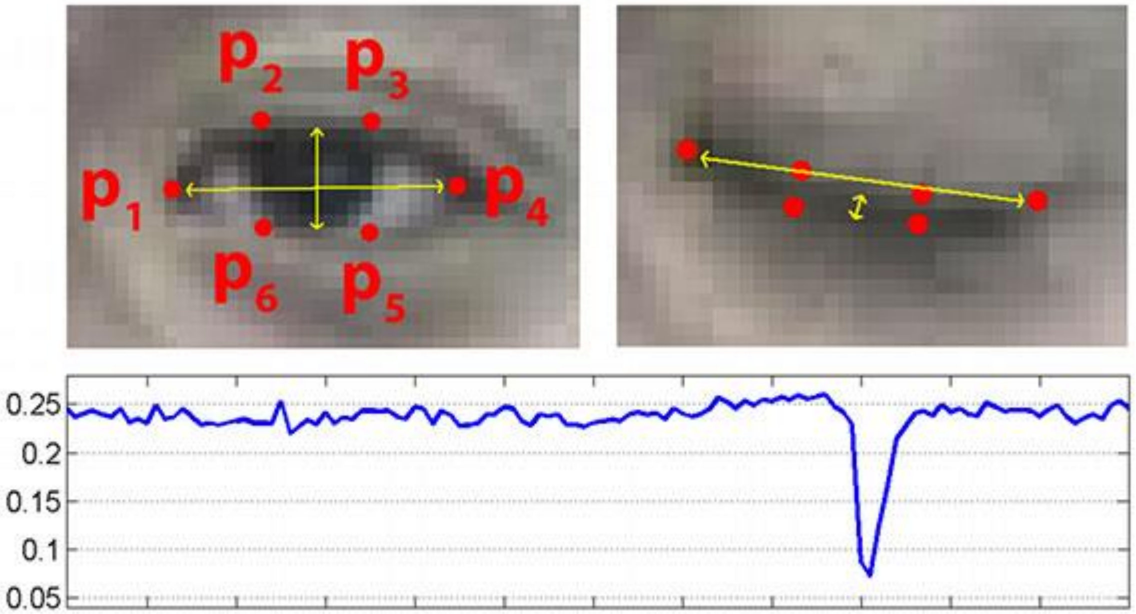
A computer vision system that can automatically detect driver drowsiness in a real-time video stream and then play an alarm if the driver appears to be drowsy.



Driver Drowsiness detection

$$\text{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$





THANK YOU FOR YOUR KIND ATTENTION!

Twitter - @mohanrajphd