

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.

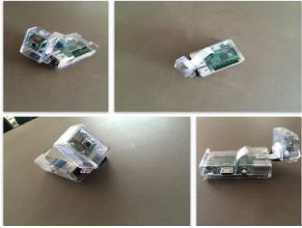


151	121	1	93	165	204	14	214	20	235	29	142	142	75	22	189	111	28	6	9
82	67	17	234	27	1	221	27	189	141	137	148	42	286	189	78	228	227	114	191
28	166	155	113	178	228	25	139	139	221	285	154	226	14	89	85	242	67	283	15
236	136	136	238	18	1	165	17	98	155	247	47	128	123	253	229	181	251	232	28
114	146	93	78	95	146	151	18	168	214	68	75	24	99	93	63	215	222	182	188
383	126	58	16	138	138	98	242	42	233	286	246	85	183	215	3	52	64	77	126
215	183	52	37	94	184	173	86	223	113	126	88	165	149	196	75	186	68	179	183
212	15	179	119	48	212	184	46	174	37	44	253	184	253	14	216	175	18	46	164
119	81	241	172	95	178	20	218	22	194	137	13	33	283	241	21	144	63	244	188
139	19	33	253	229	1	152	213	52	44	37	214	142	121	149	189	99	232	183	71
88	288	194	185	148	288	223	158	164	182	45	36	152	27	198	137	51	1	237	247
213	18	228	213	143	184	147	39	97	288	1	14	242	78	2	38	215	87	149	185
9	218	182	246	75	9	158	184	184	129	32	88	182	32	99	169	91	166	73	114
124	52	76	148	249	187	65	218	187	181	186	219	9	283	289	248	48	249	119	232
6	251	52	288	46	85	185	38	77	248	177	232	38	283	119	8	217	139	139	157
158	194	26	288	148	187	288	28	74	88	154	145	45	253	158	185	255	23	236	156
13	183	248	153	168	285	146	188	254	218	157	148	221	68	247	113	5	138	26	186
138	53	128	212	61	216	281	118	148	183	182	288	195	246	148	138	54	182	139	79
185	246	22	182	151	213	48	138	8	93	17	233	85	169	166	24	49	188	97	18
157	251	181	238	23	162	78	238	75	24	84	242	247	144	283	3	19	24	188	88
187	186	152	63	167	98	125	188	136	121	67	47	185	88	123	188	185	127	153	12
139	197	55	289	28	124	288	288	184	48	37	113	214	252	283	88	148	211	7	16
113	19	144	223	62	253	282	188	47	242	142	241	66	86	214	133	148	233	189	288
218	144	31	16	136	113	227	82	183	183	67	215	174	111	188	54	144	56	39	183



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.



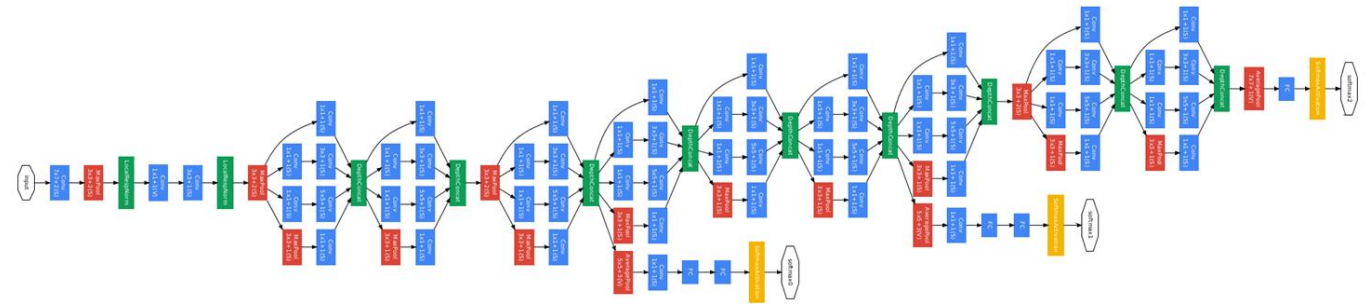
The same venti coffee will look dramatically different when it is **photographed up close** and when it is **captured farther way**.



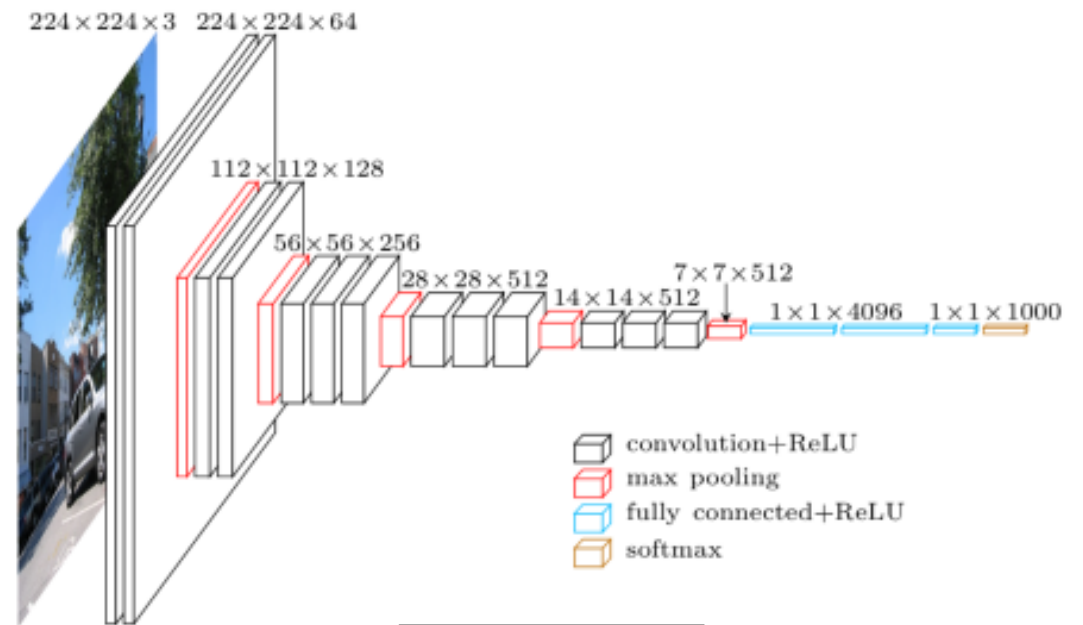
On the left we have a picture of a dog. The right we have a picture of the same dog, but notice how the dog is resting underneath the covers, **occluded from our view**.

STATE-OF-THE-ART CNN FOR IMAGE CLASSIFICATION

VGG16	VGG19	OverFeat	GoogLeNet	ResNet50
image	image	image	image	image
conv-64	conv-64	conv-96	conv-64	conv-64
conv-64	conv-64	maxpool	maxpool	maxpool
maxpool	maxpool			
		conv-256	conv-192	conv2_x
conv-128	conv-128	maxpool	maxpool	conv-64
conv-128	conv-128	conv-512	inception-256	conv-64 x 3
maxpool	maxpool	conv-1024	inception-480	conv-256
		conv-1024	maxpool	
conv-256	conv-256	conv-1024	inception-512	conv3_x
conv-256	conv-256	maxpool	inception-512	conv-128
conv-256	conv-256	FC-3072	inception-512	conv-128 x 4
maxpool	conv-256	FC-4096	inception-528	conv-512
	maxpool	FC-1000	inception-832	
conv-512	conv-512	softmax	maxpool	conv4_x
conv-512	conv-512		inception-832	conv-256
conv-512	conv-512		inception-1024	conv-256 x 6
maxpool	conv-512		avgpool	conv-1024
	conv-512		dropout-1024	
conv-512	conv-512		FC-1000	conv5_x
conv-512	conv-512		softmax	conv-512
maxpool	conv-512			conv-512 x 3
	conv-512			conv-2048
FC-4096	maxpool			
FC-4096	FC-4096			avgpool
FC-1000	FC-4096			FC-1000
softmax	FC-1000			softmax
	softmax			



Google Inception-V3

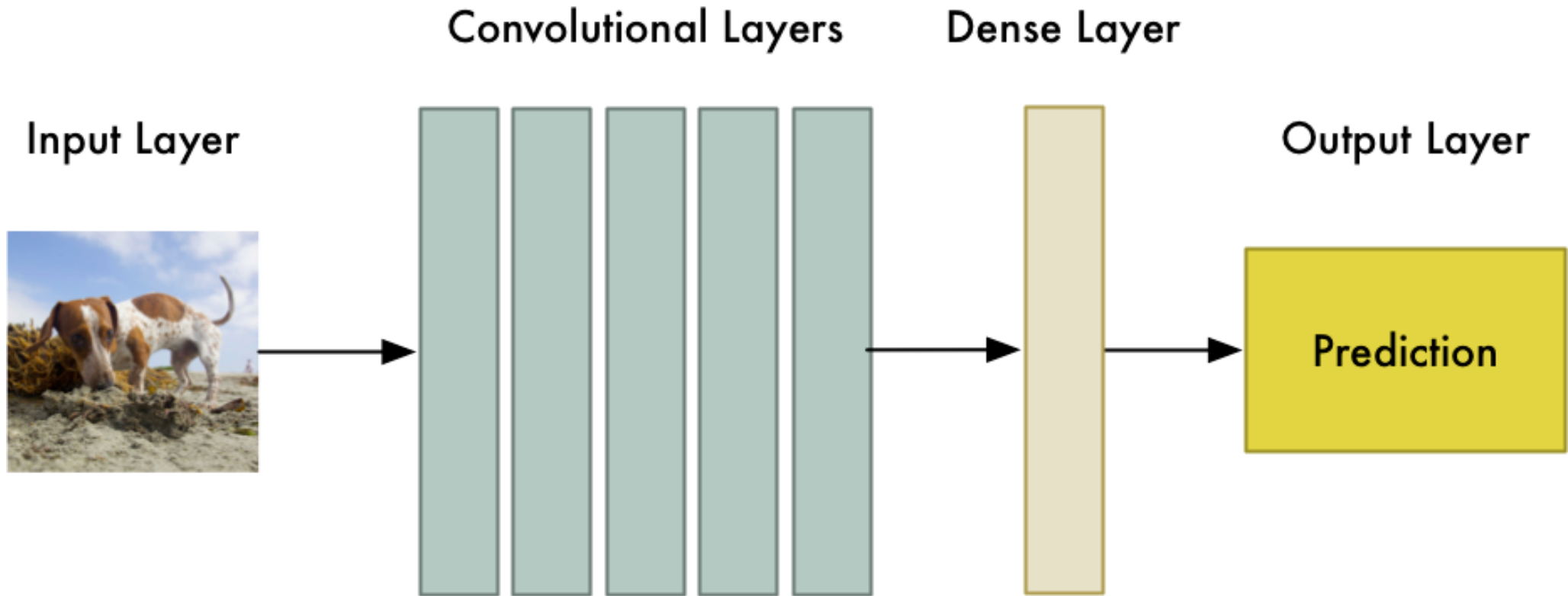


VGG-16

Transfer Learning

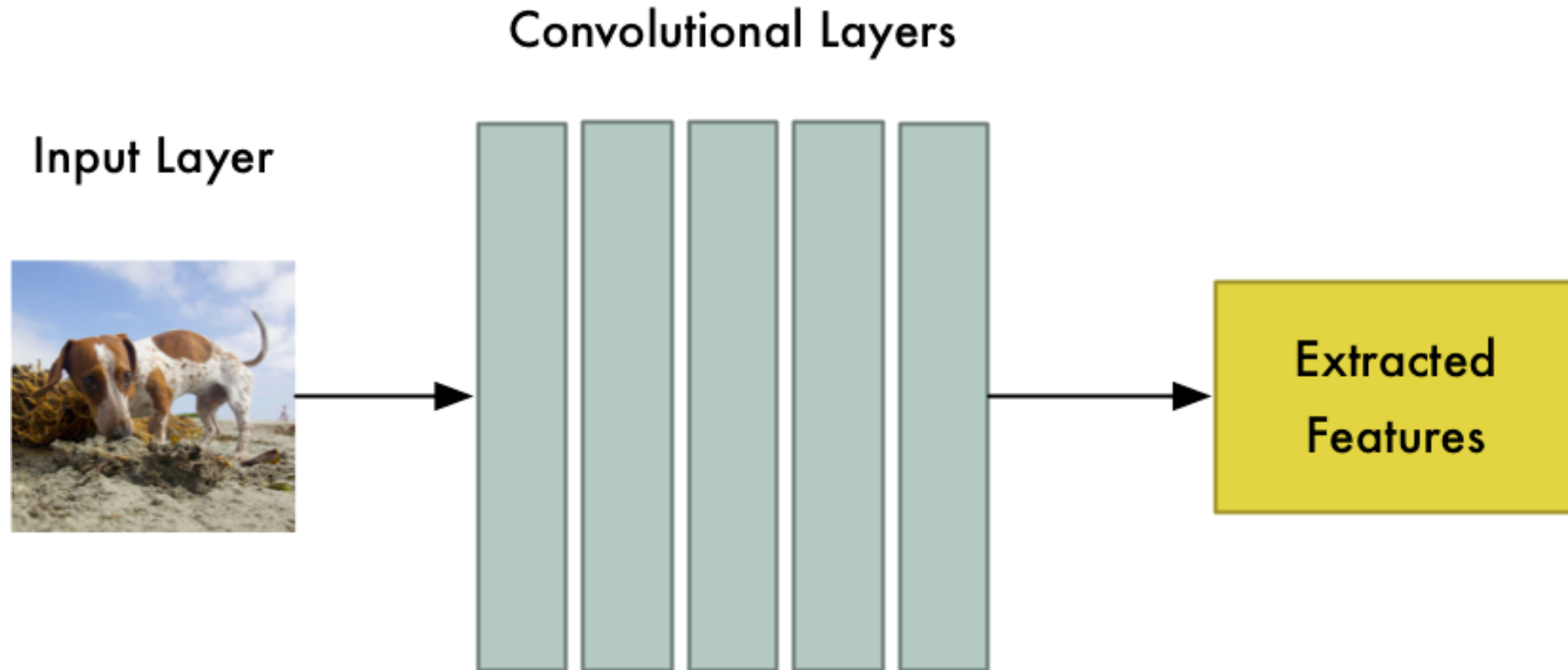
- Convolutional neural networks (CNNs) are great at image classification.
- But whenever we train a new convolutional neural network, it has to relearn how to classify images from scratch—which means that we need a massive amount of training data to make CNNs work well.
- In transfer learning, you take 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



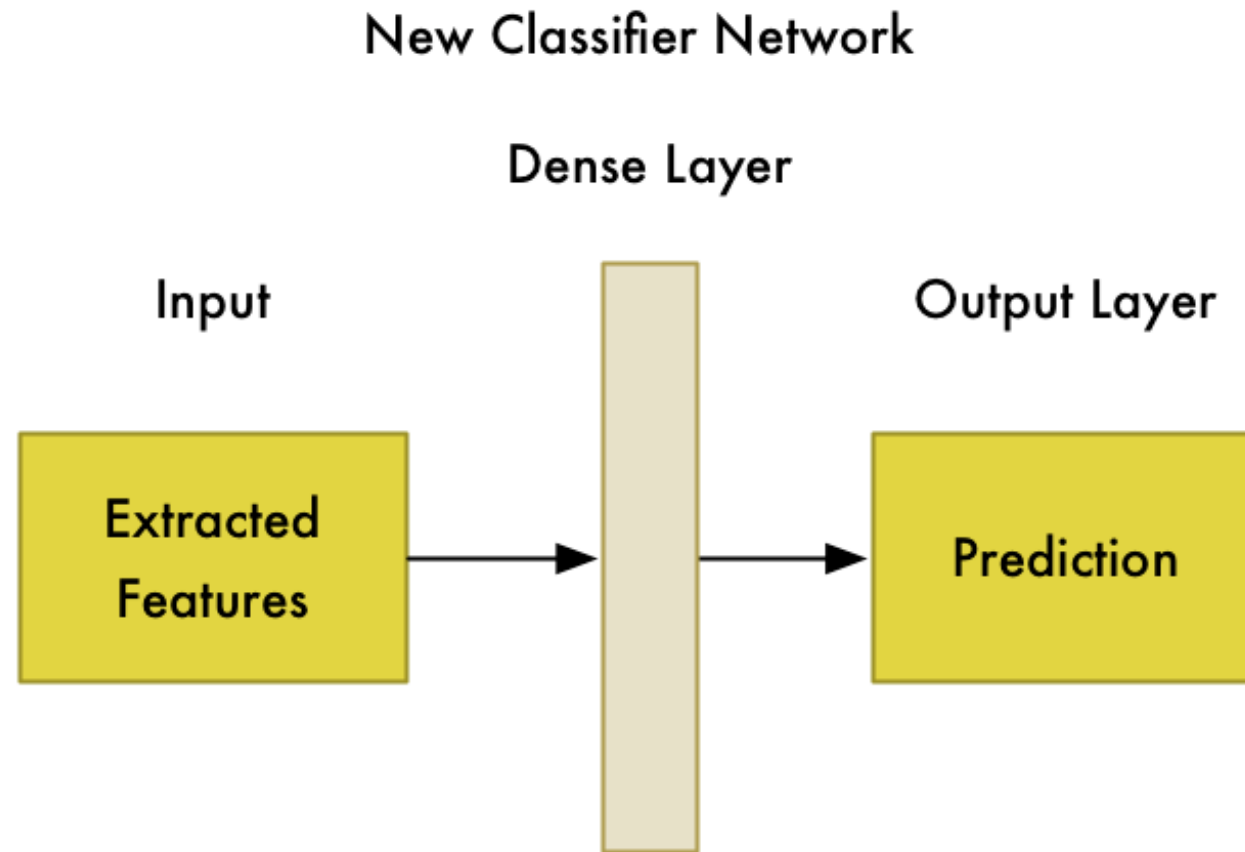
Transfer Learning

Pre-Trained Feature Extractor Network



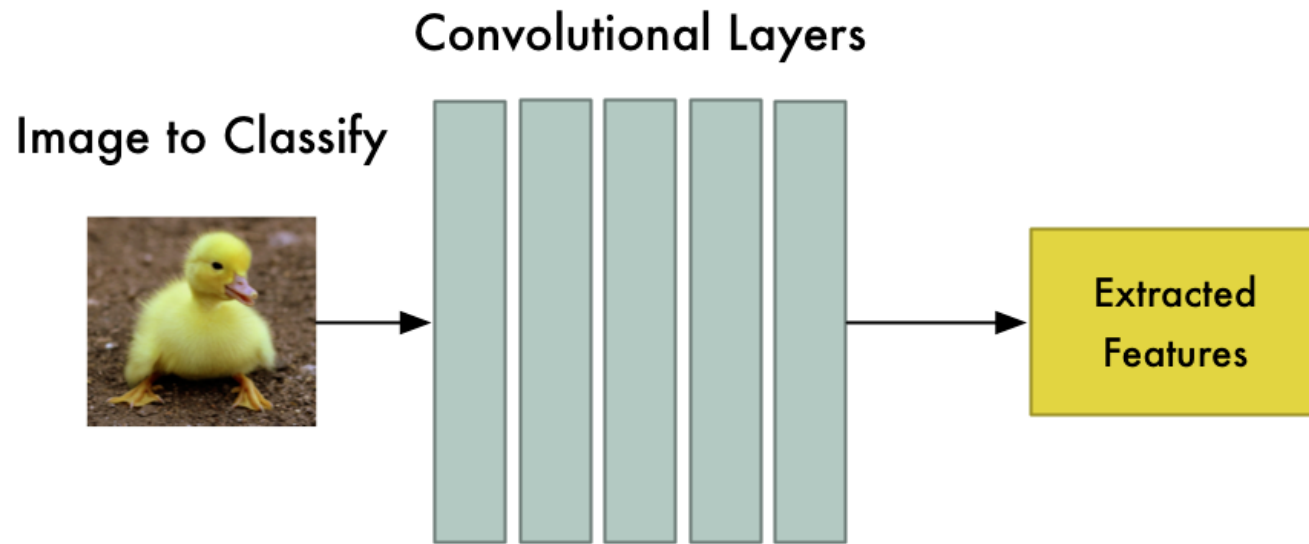
The neural network had to learn to detect all kinds of animal shapes that are probably also useful for detecting birds. We'll keep all the knowledge it has for detecting shapes.

Transfer Learning

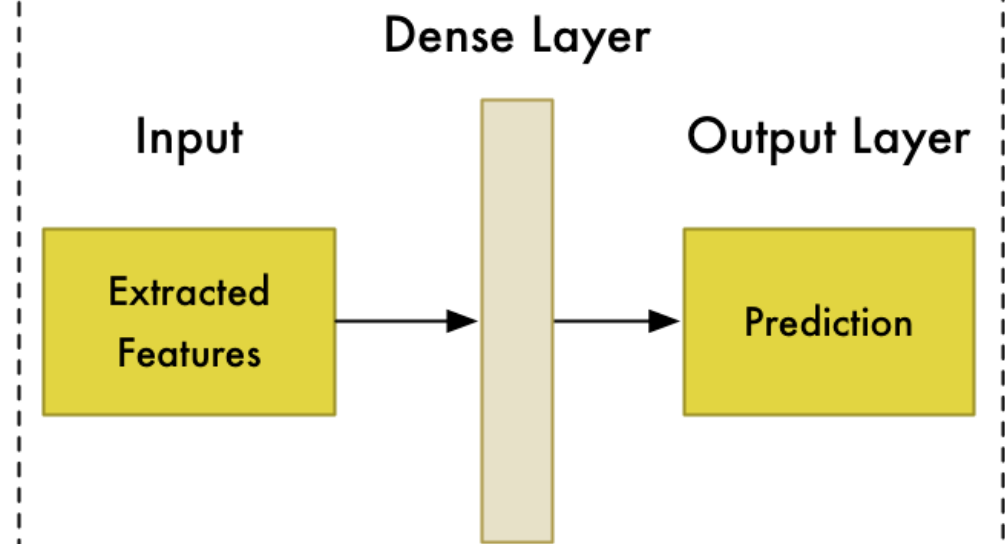


Transfer Learning

Pre-Trained Feature Extractor Network



New Classifier Network



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**.

Classification



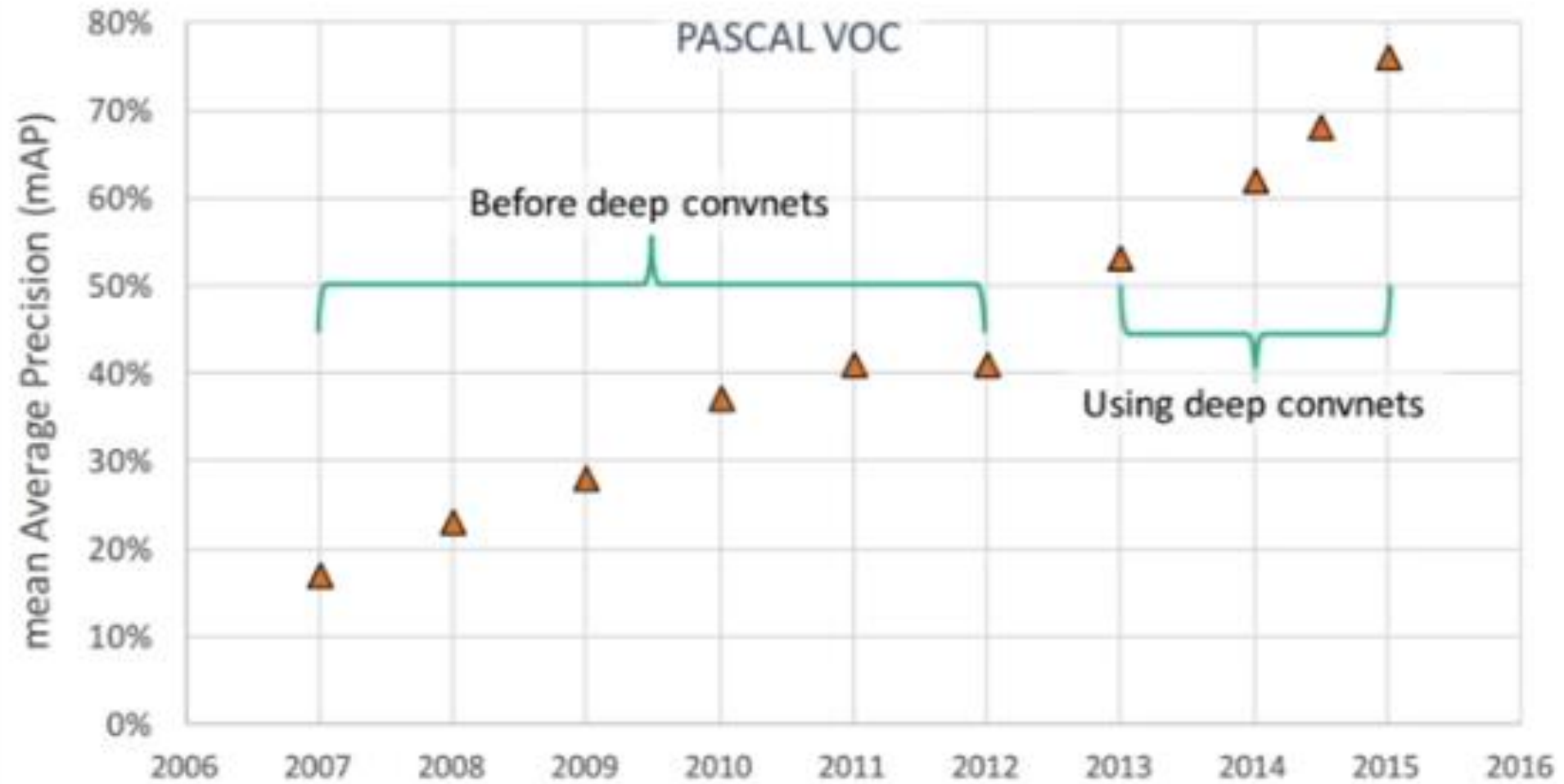
CAT

Classification
+ Localization



CAT

DEEP LEARNING FOR OBJECT DETECTION



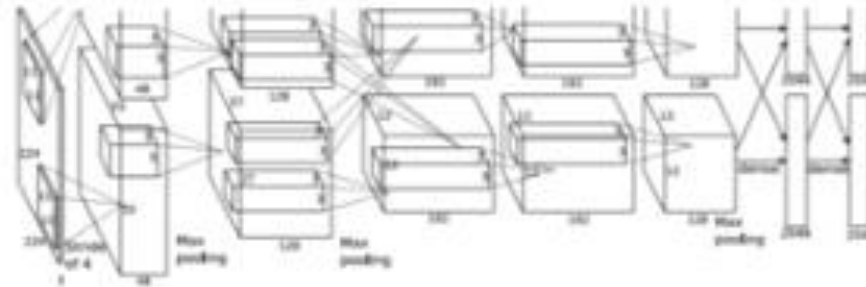
SLIDING WINDOWS - HOW IT WORKS?

- **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

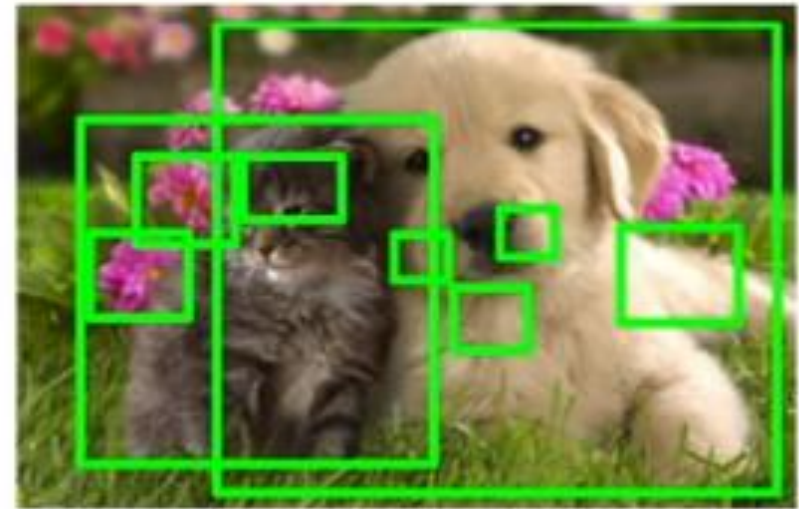
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



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



$$G_x = I \star D_x$$



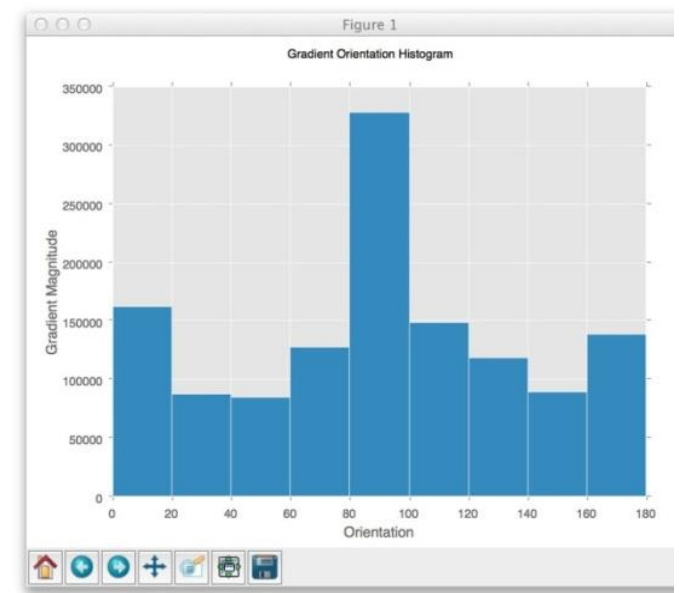
$$G_y = I \star D_y$$



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

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

WEIGHTED VOTES IN EACH CELL

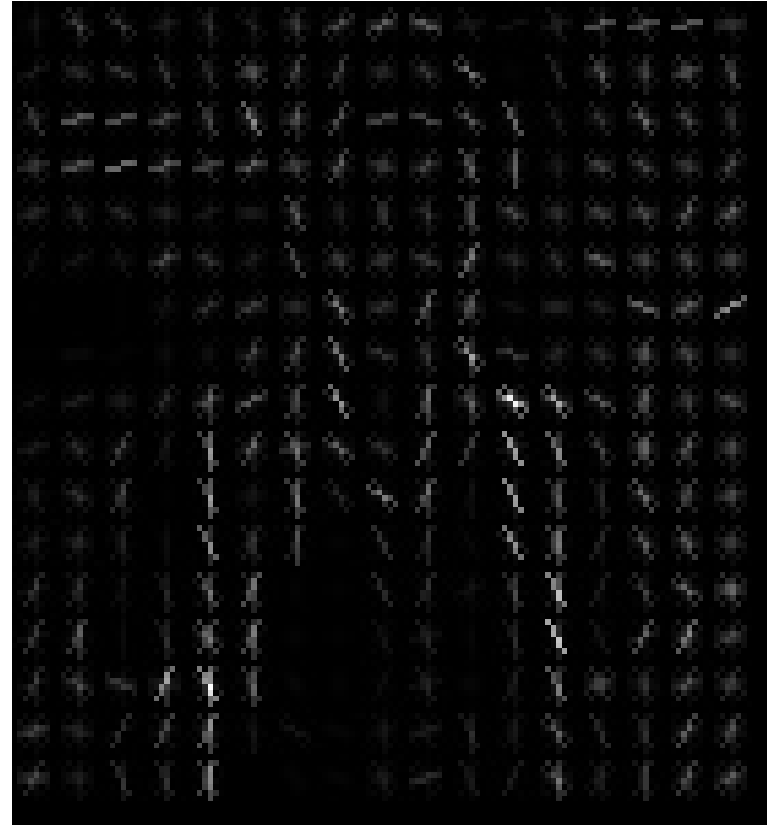


CONTRAST NORMALIZATION OVER BLOCK

Block 1

Cell #1	Cell #2	Cell #3
Cell #4	Cell #5	Cell #6
Cell #7	Cell #8	Cell #8

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

R-CNN: *Regions with CNN features*

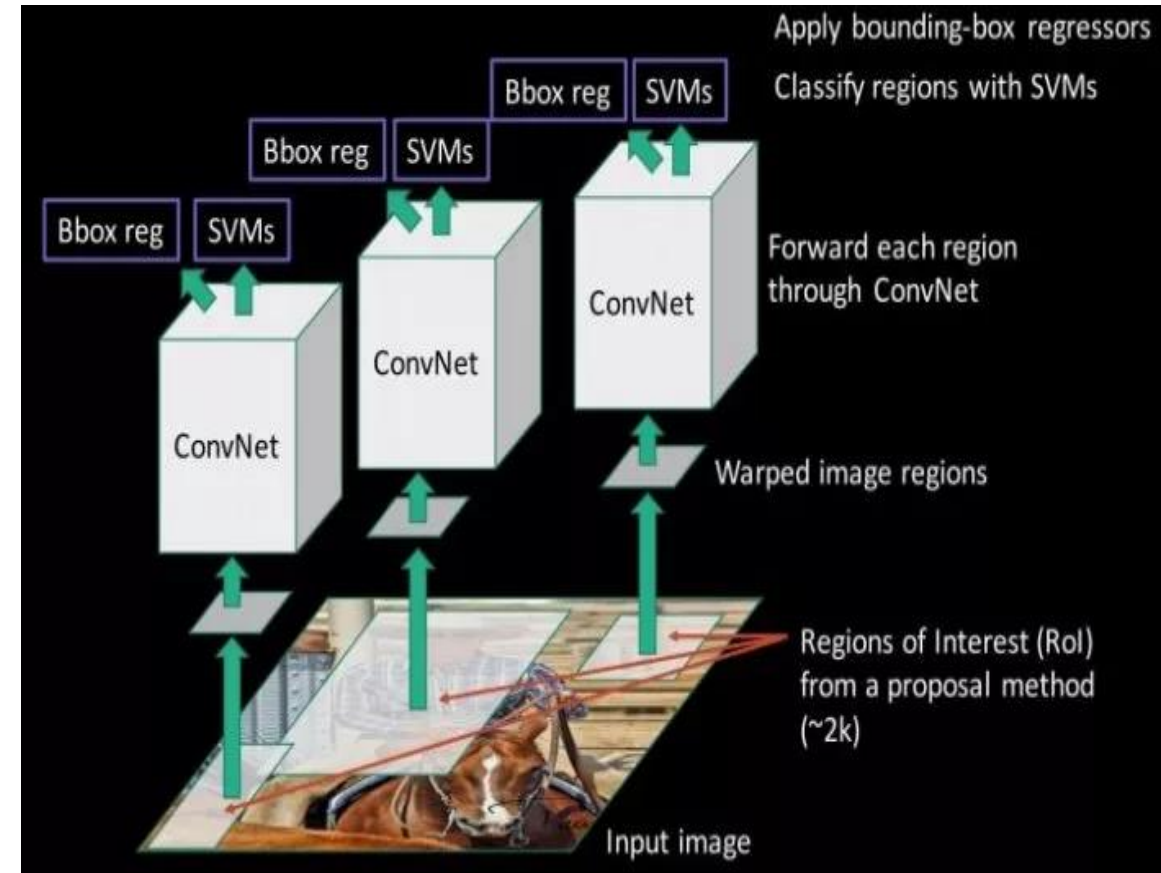
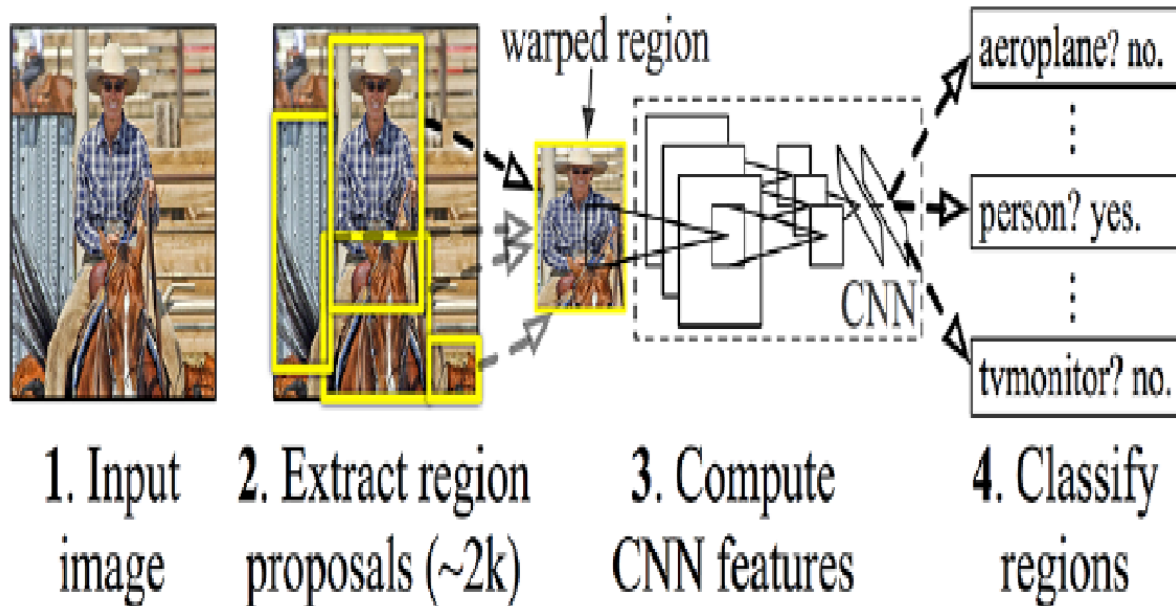
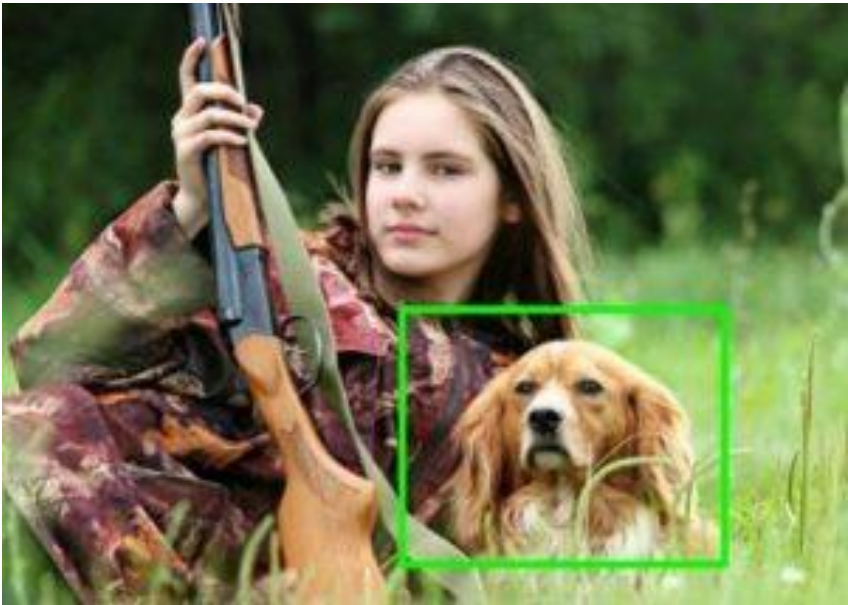
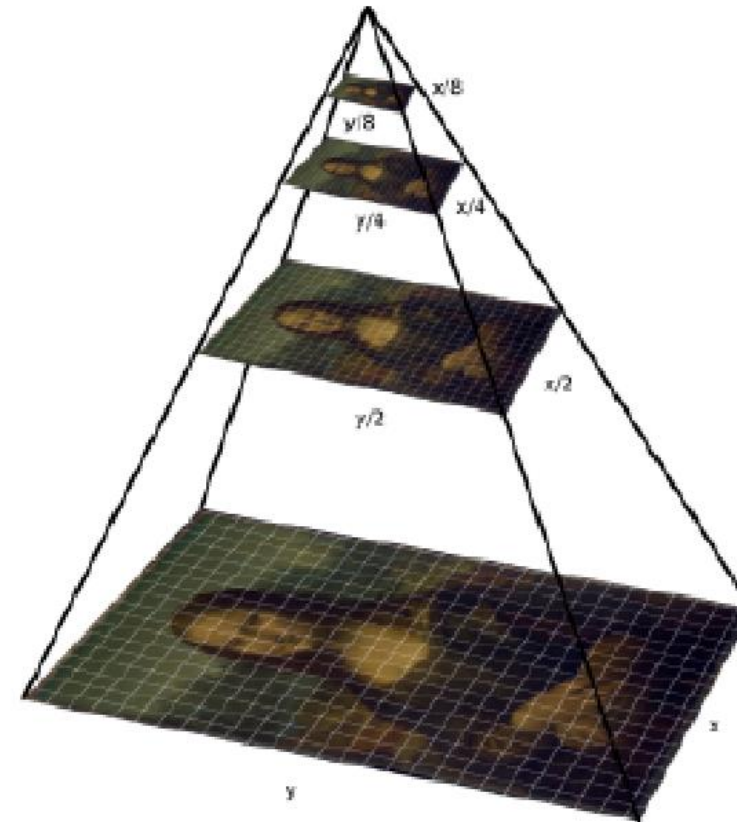
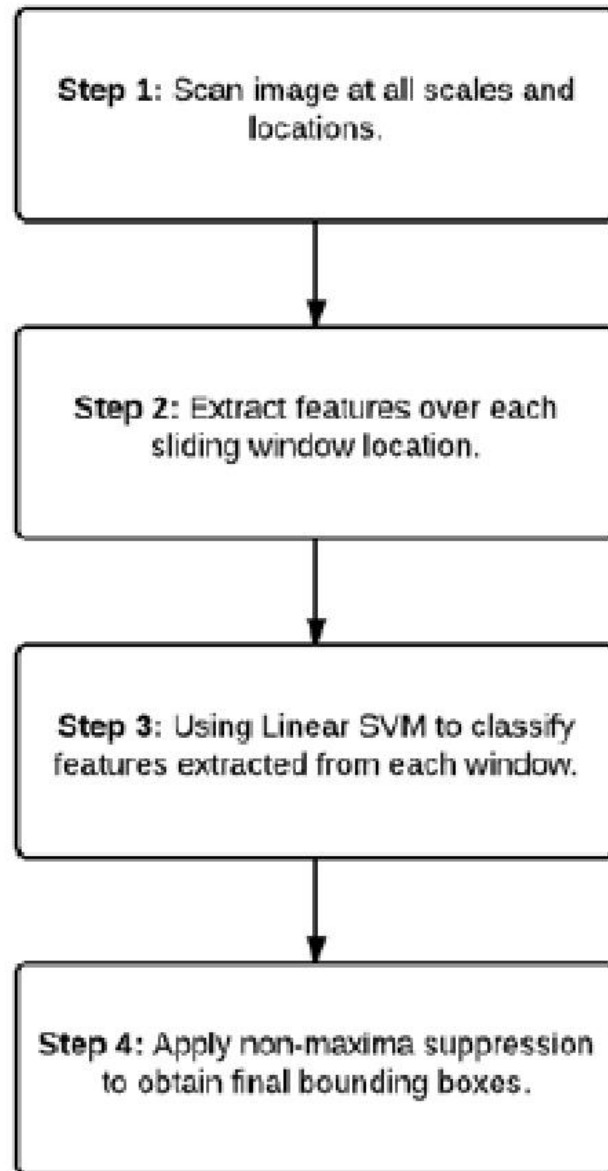


IMAGE PYRAMIDS

- 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.

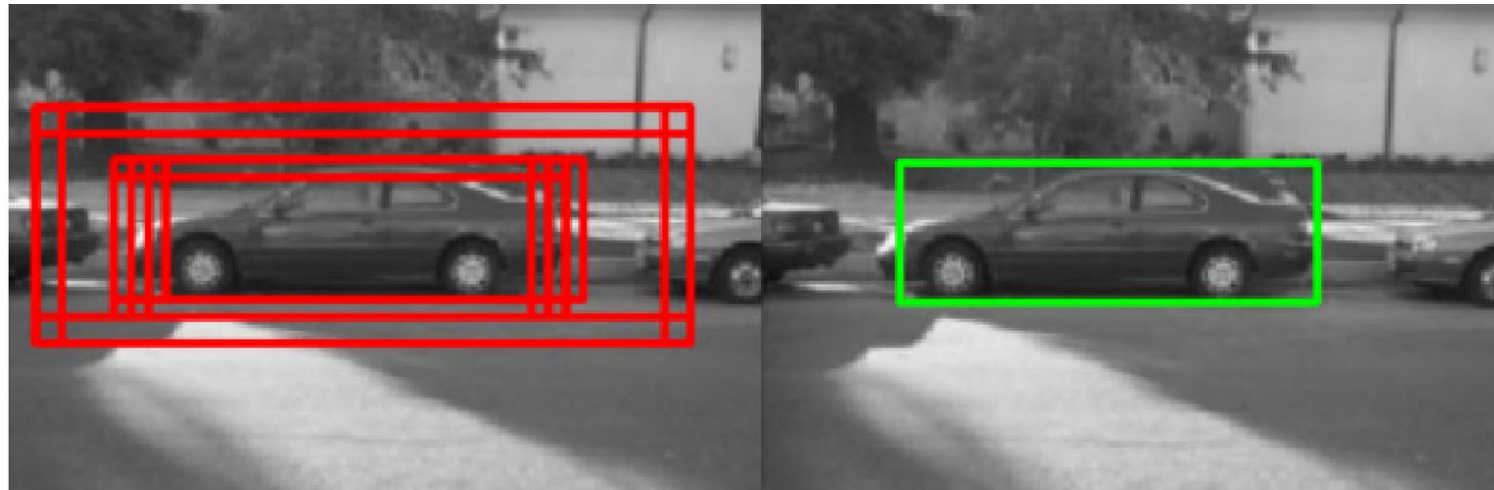


OBJECT DETECTION PIPELINE

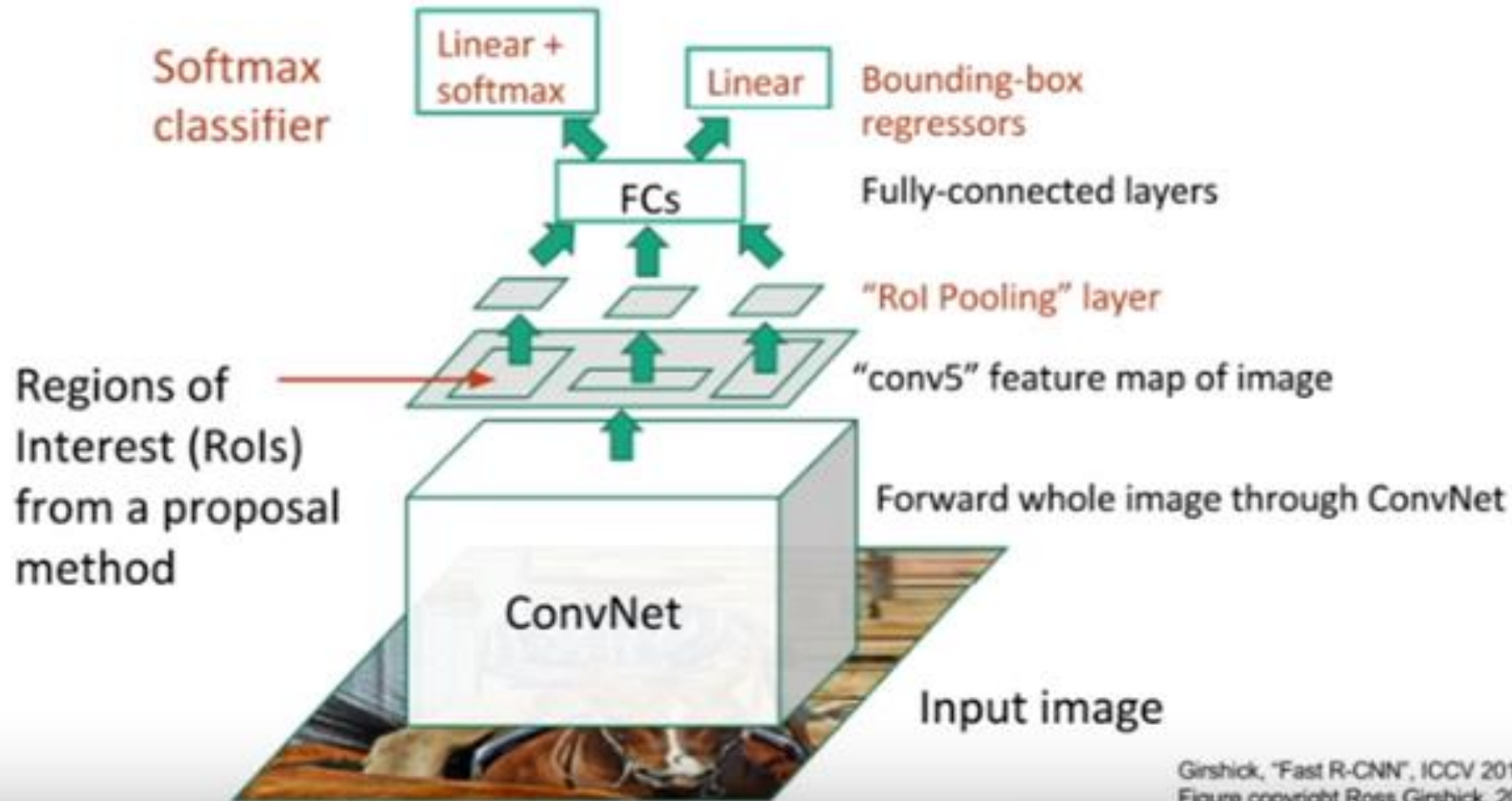


NON-MAXIMA SUPPRESSION

- 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



Girshick, "Fast R-CNN", ICCV 2015.

Figure copyright Ross Girshick, 2015. [License](#) | [Source](#) | [Download](#) | [Feedback](#)

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.

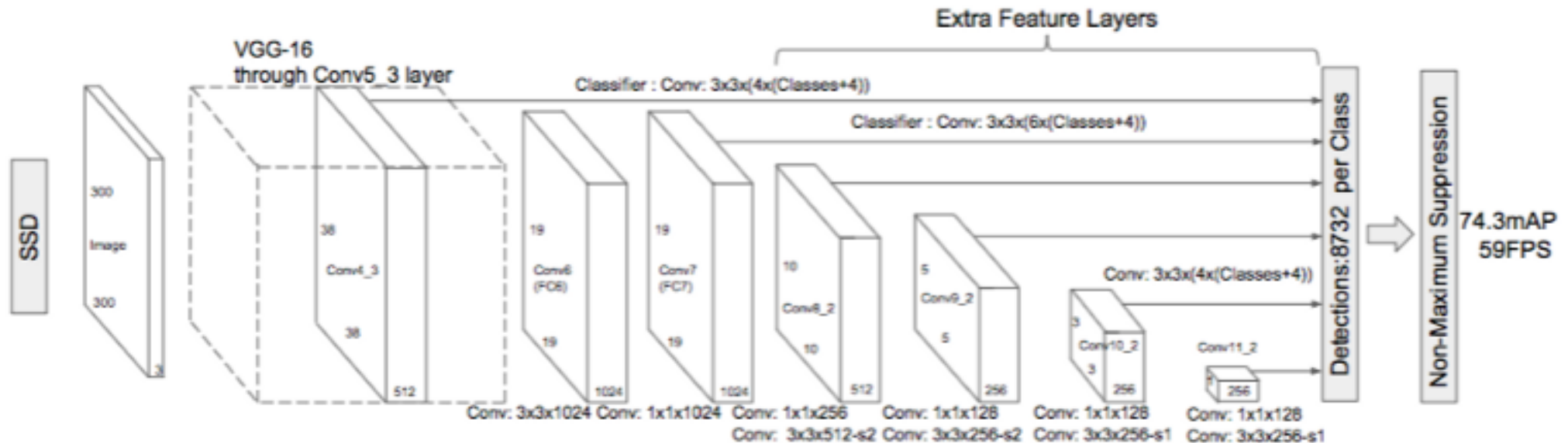


Image Segmentation

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.

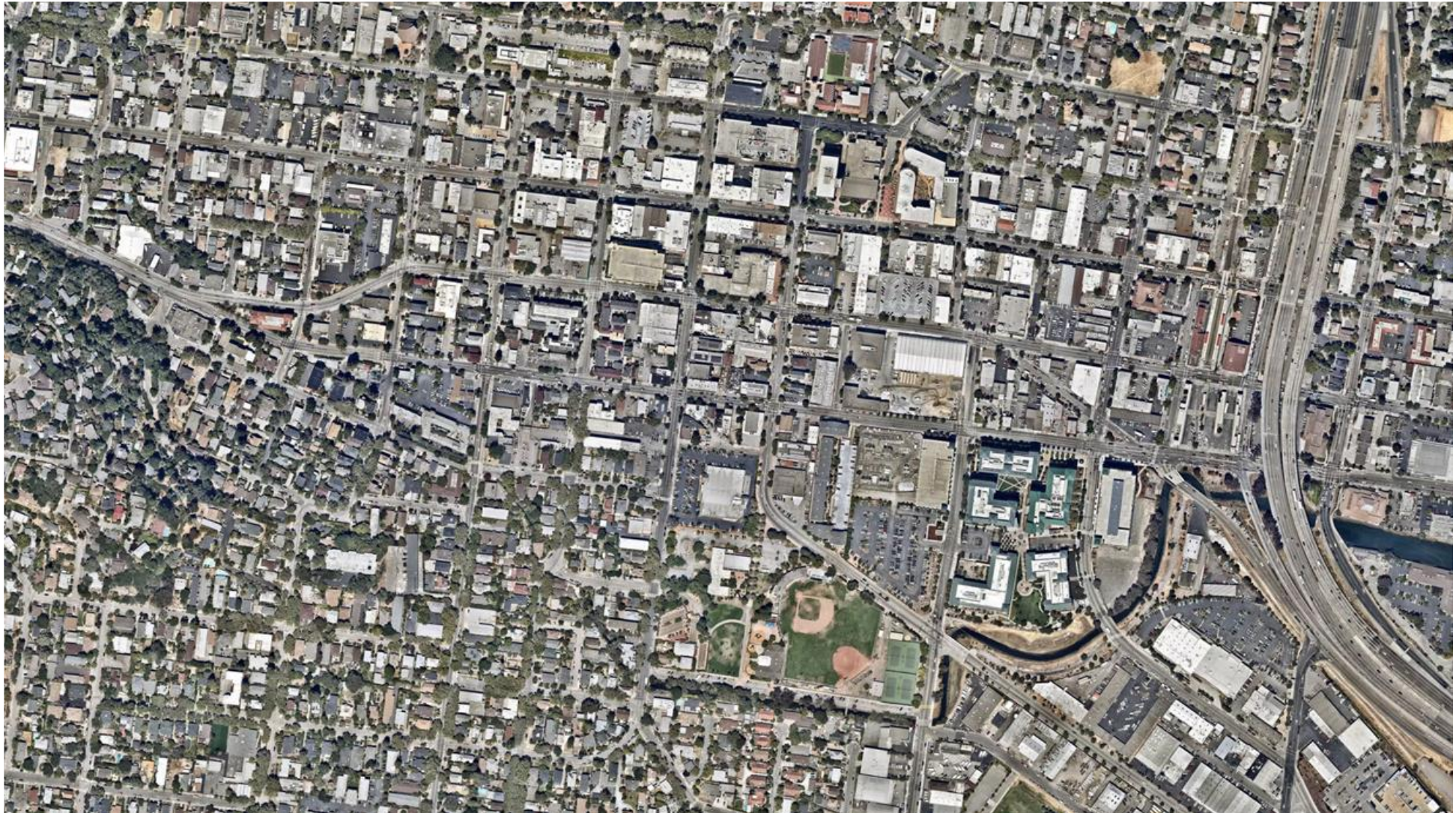
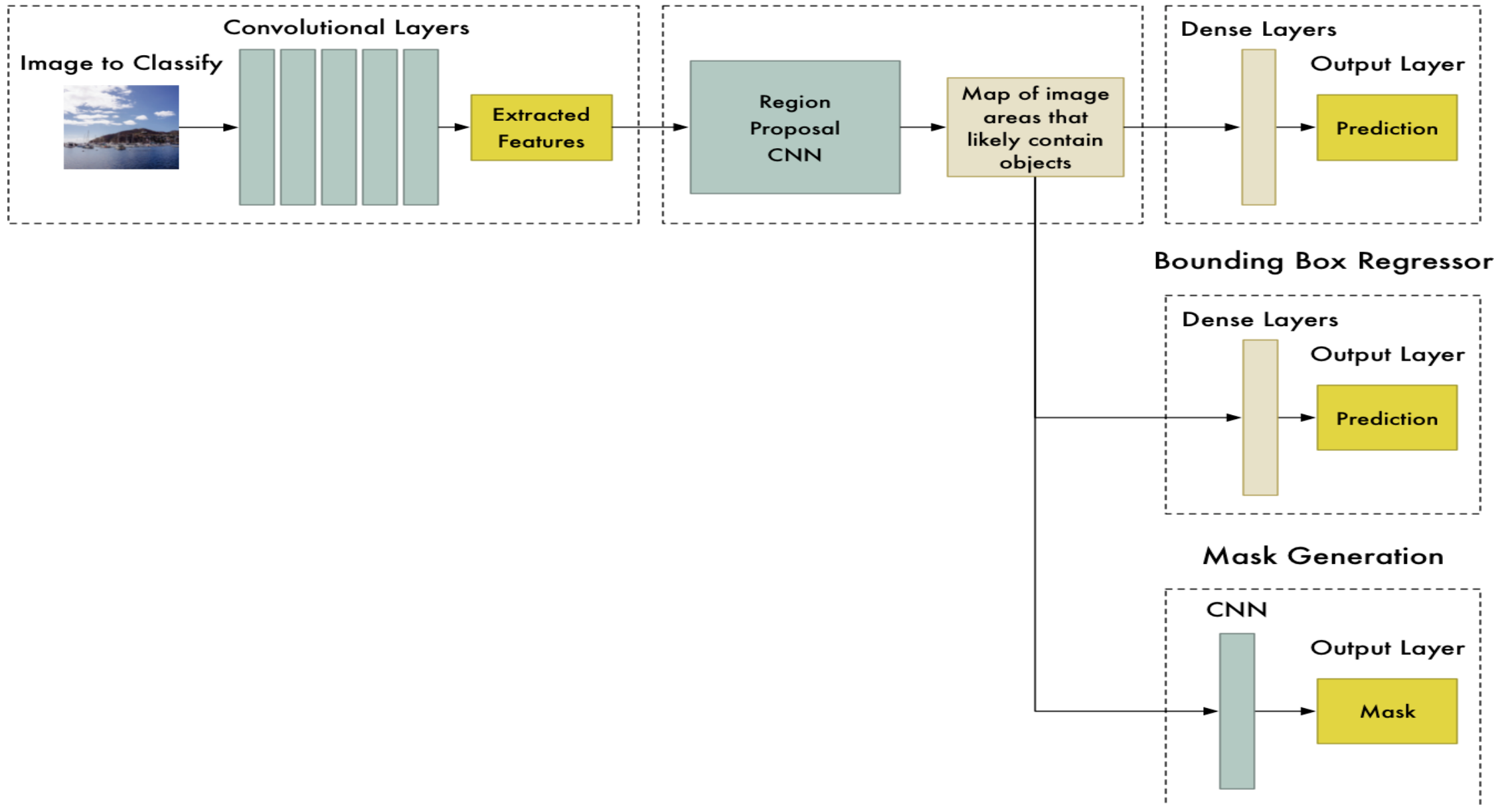


Image Segmentation - Architecture

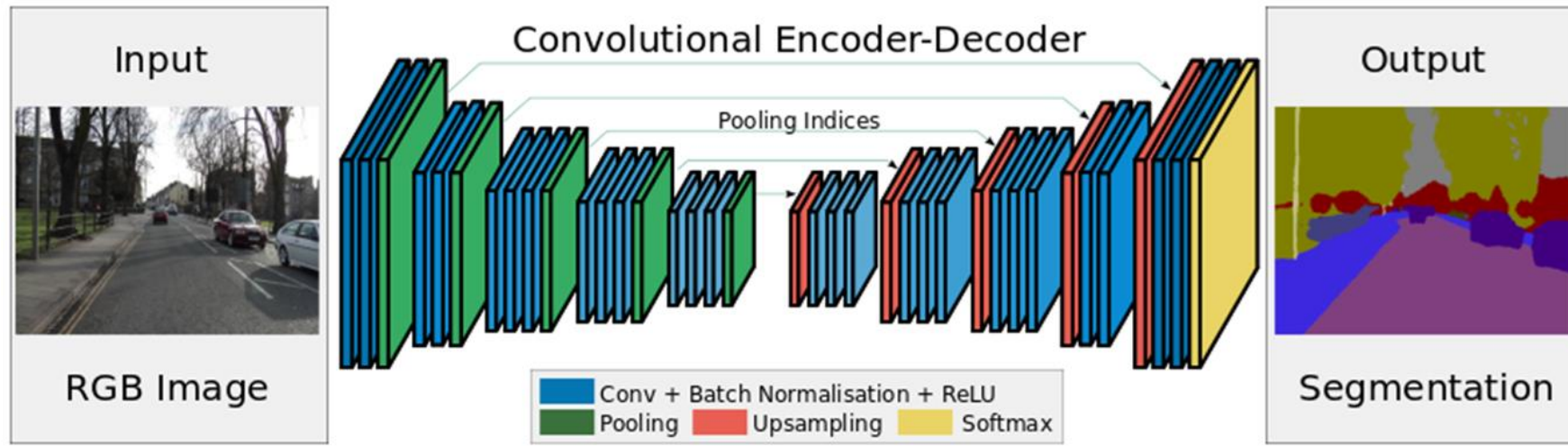


CNN FOR IMAGE SEGMENTATION

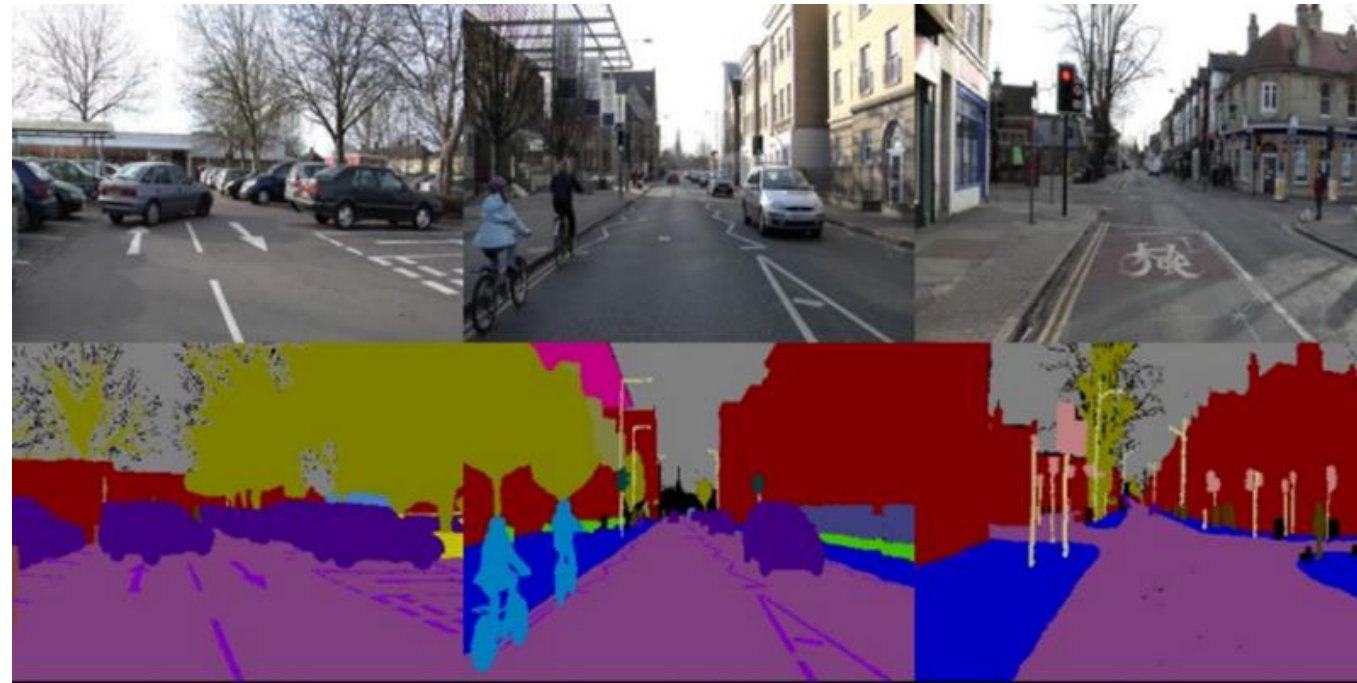
- Pixel level semantic segmentation of a road scene understanding.
- Assign a **label** to **every pixel** in an image.
- Object detection, medical imaging, machine vision, traffic control systems.



SEGNET



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

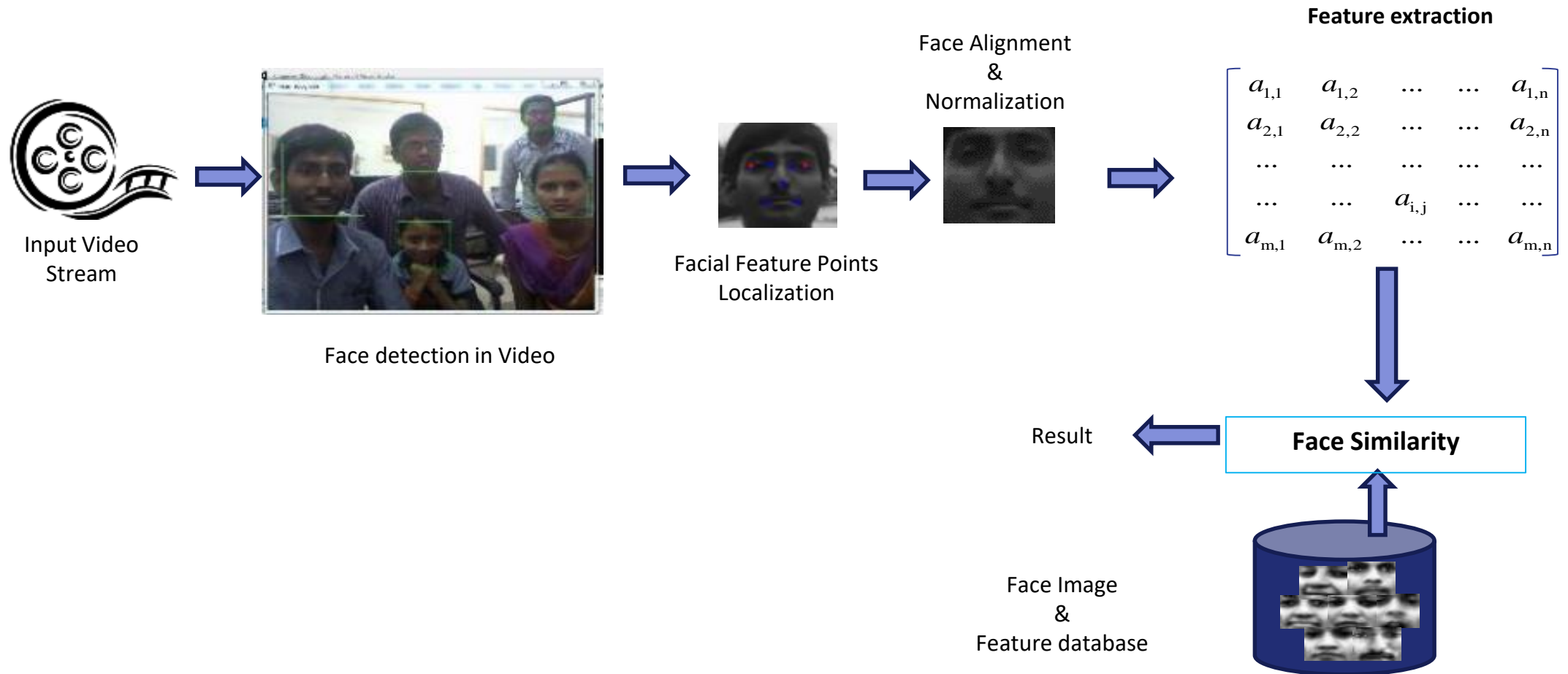




Demo

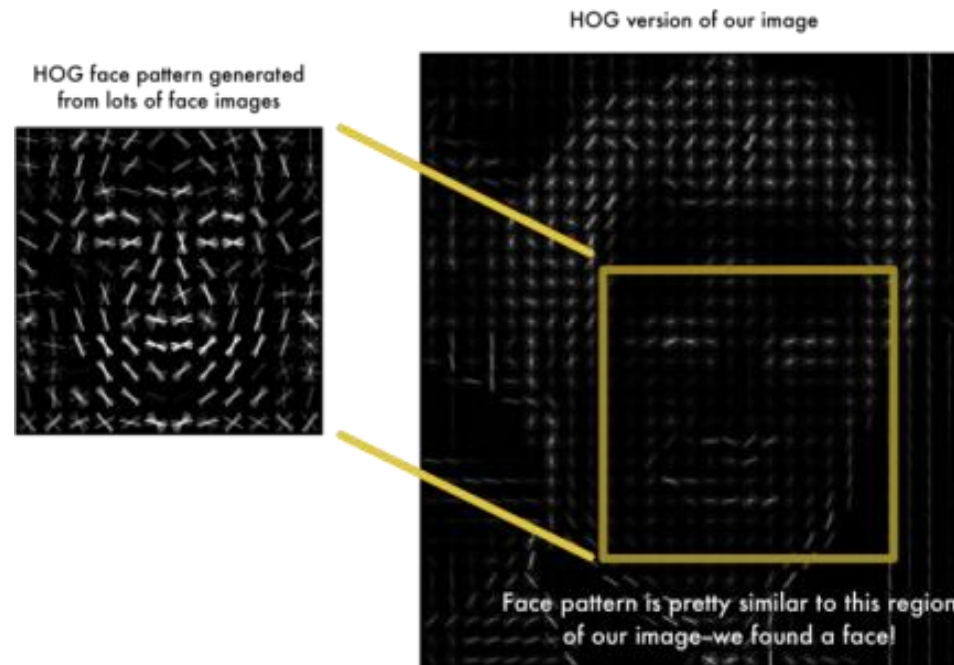
FACE RECOGNITION SYSTEM USING DEEP LEARNING

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

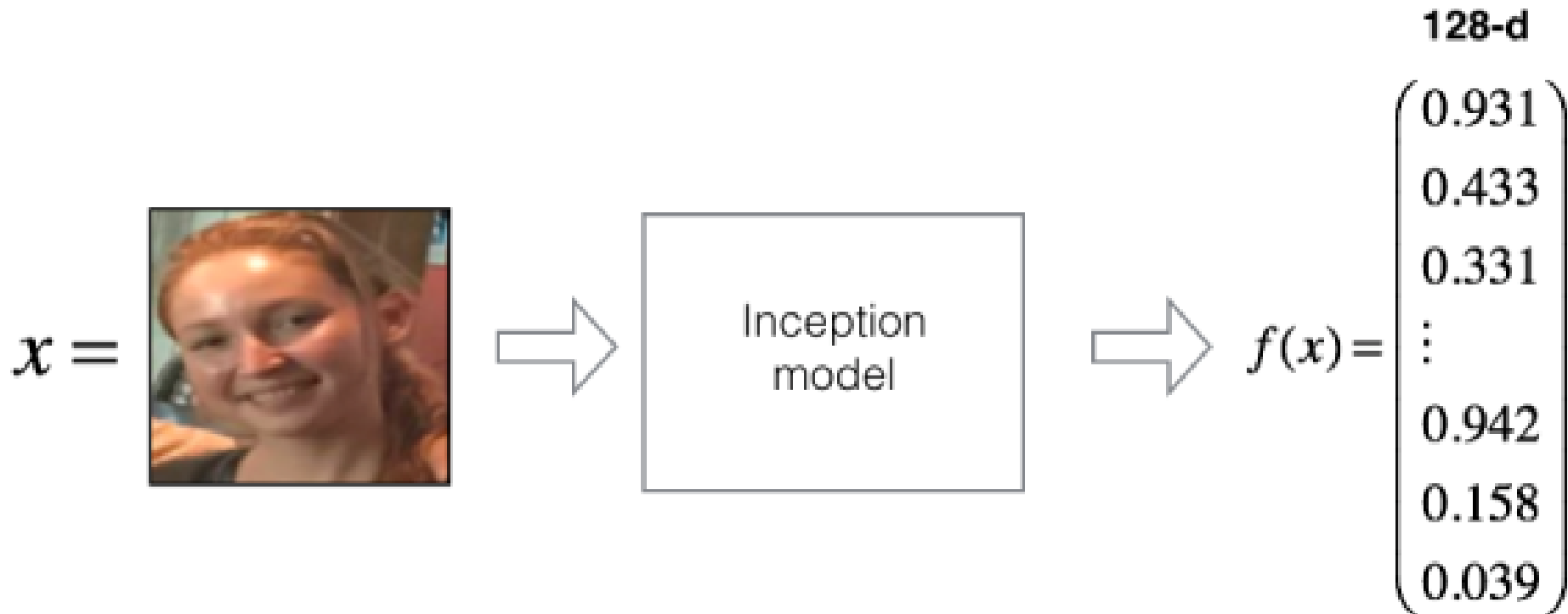


FACE DETECTION AND ALIGNMENT

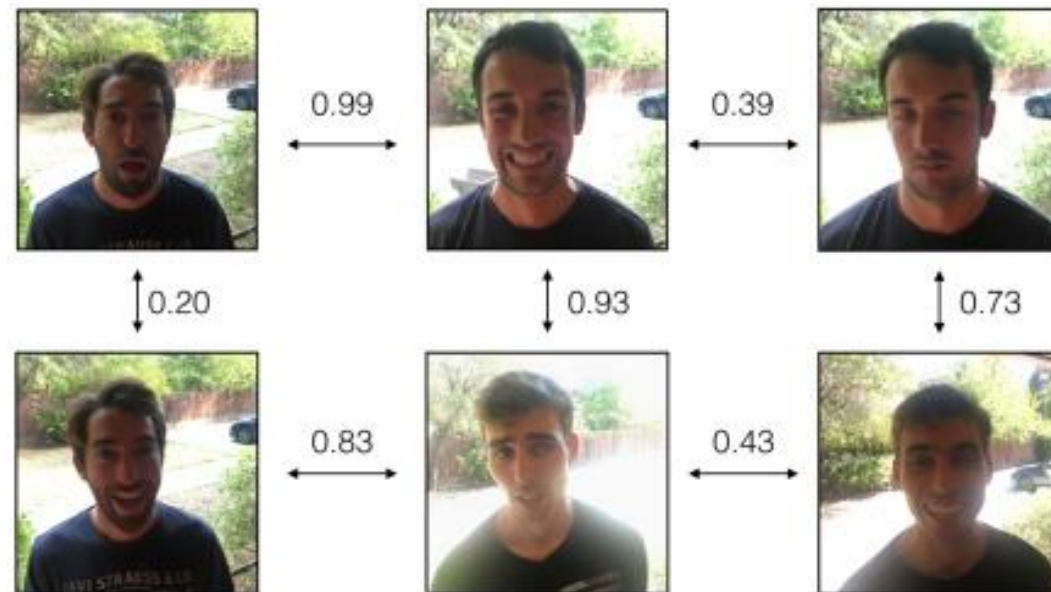
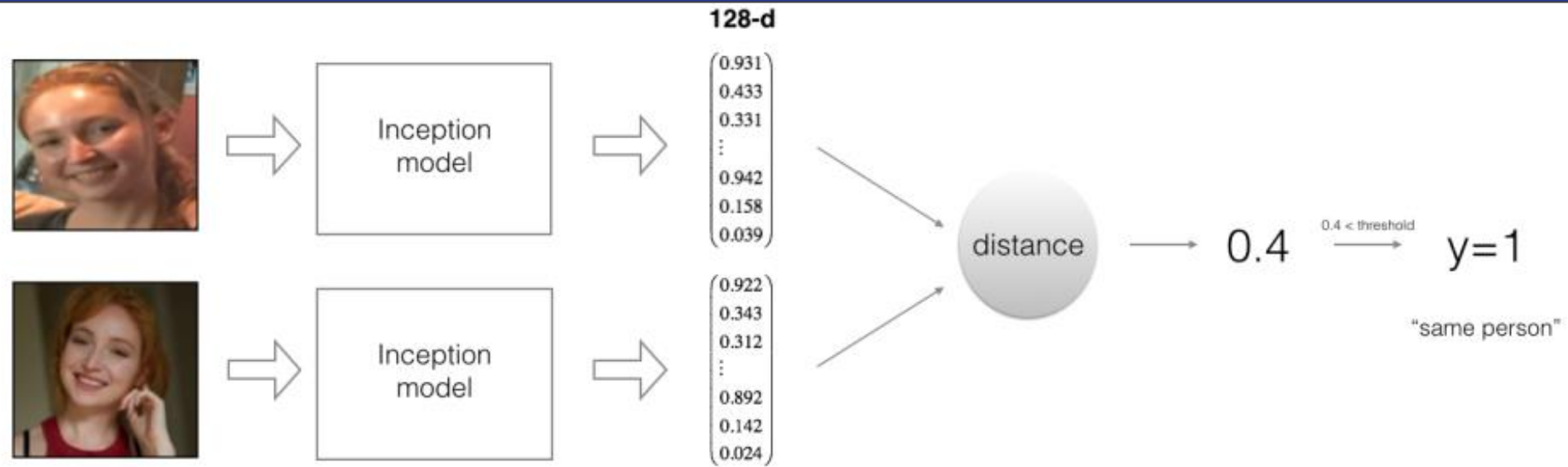
- HOG + Linear SVM based face detection is used to detect the faces in each frame.
- For each detected face image localize 68 (x, y) coordinates that map to facial structures on the face.
- The angle of the face image is computed by taking the difference between left and right eye centers.
- The face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



FEATURE VECTOR REPRESENTATION

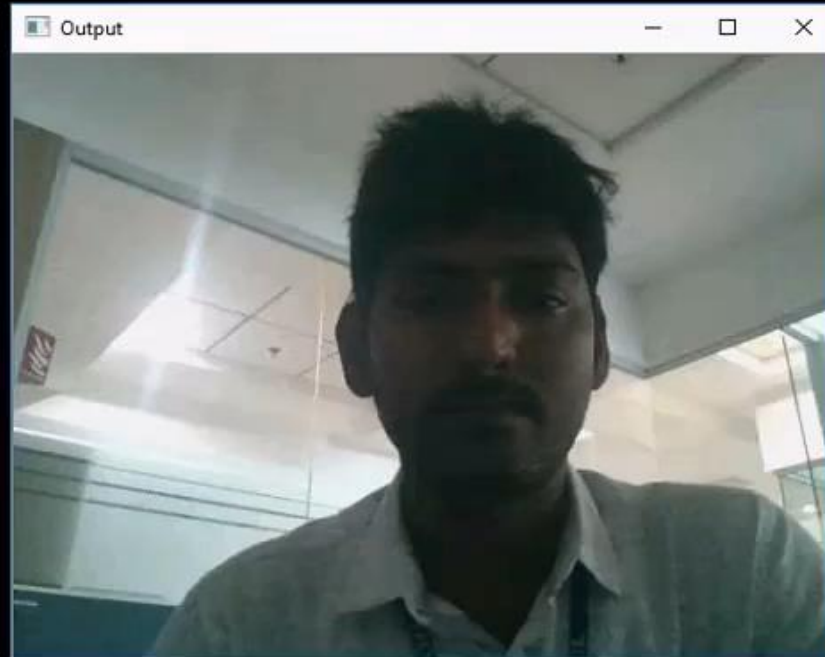


FEATURE MATCHING



FACE RECOGNITION DEMO

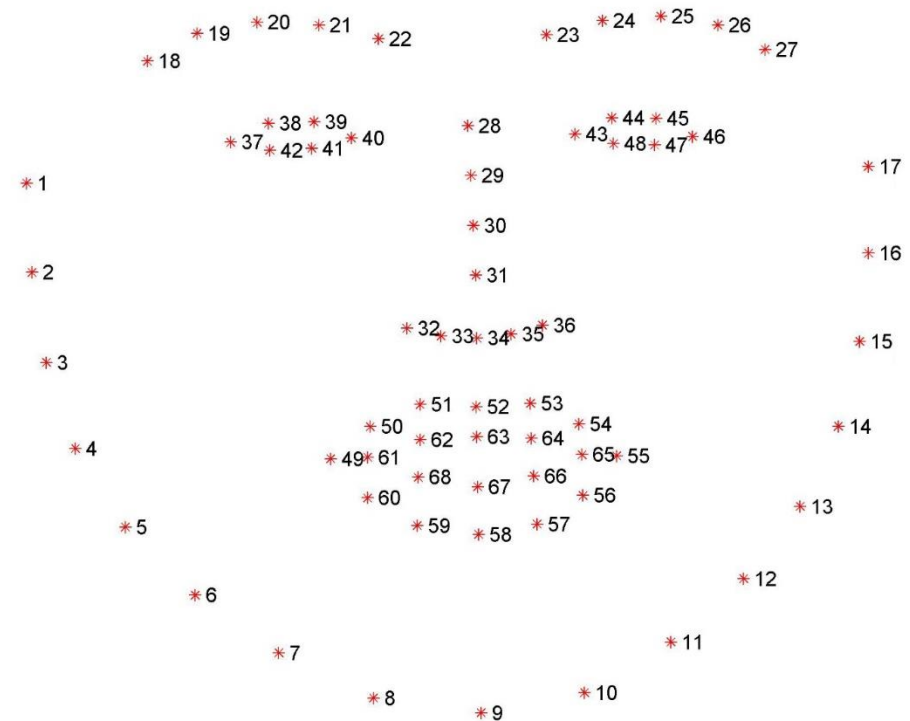
```
mohanraj_featrures.h5
mohanraj_labels.h5
(12, 128)
(12,)
mohanrajv_featrures.h5
mohanrajv_labels.h5
(13, 128)
(13,)
mohanrajvv_featrures.h5
mohanrajvv_labels.h5
(70, 128)
(70,)
shobana_featrures.h5
shobana_labels.h5
(35, 128)
(35,)
(337, 128)
(337,)
[INFO] creating model...
[INFO] dumping classifier...
```



```
C:\HCL\E-Learning\Face Recongition\code>python Test.py --shape-predictor shape_predictor_68_face_landm
arks.dat
```

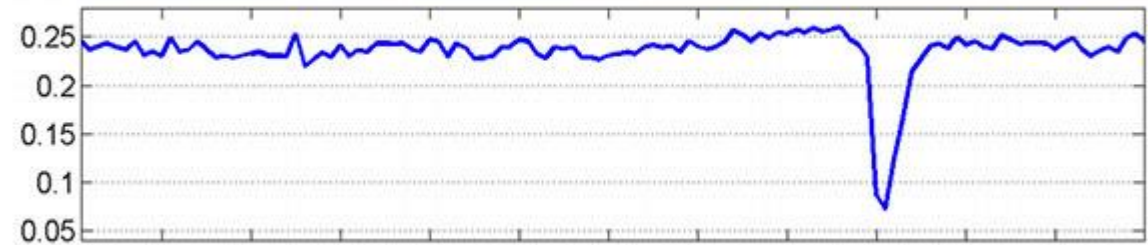
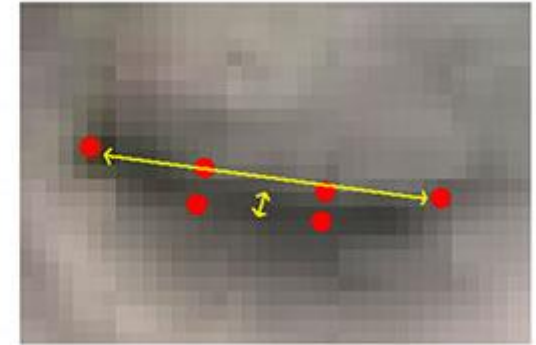
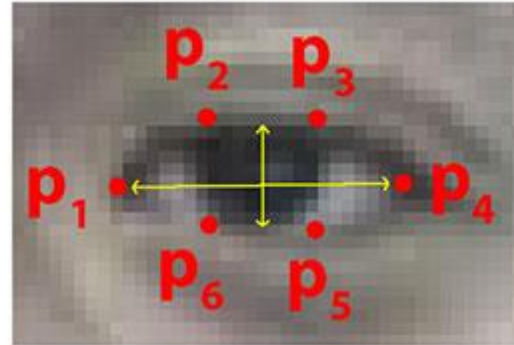
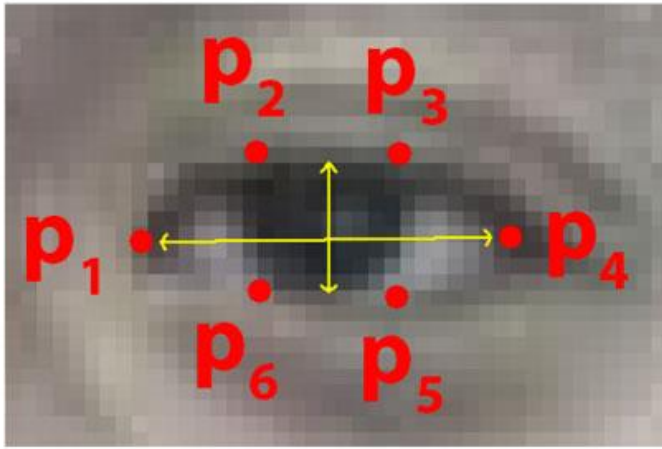
DRIVER DROWSINESS DETECTION

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