Computer vision

*By*

ROHIT KUMAR (Roll No.-CSE/16031/186)



*Bachelor Thesis submitted to*

Indian Institute of Information Technology Kalyani

*for the partial fulfillment of the degree of*

**Bachelor of Technology**

**in**

**Computer Science and Engineering/Information Technology**

**Dec, 2019**

Certificate

This is to certify that the thesis entitled “Computer vision” being submitted by Rohit Kumar, an undergraduate student, Reg. No- 186 Roll No-16031, in the Department of Computer Science and Information Engineering, Indian Institute of Information Technology Kalyani, West Bengal 741235, India, for the award of Bachelors of Technology in Computer Science and Information Engineering /Information Technology is an original research work carried by him under my supervision and guidance. The thesis has fulfilled all the requirements as per the regulations of Indian Institute of Information Technology Kalyani and in my opinion, has reached the standards needed for submission. The work, techniques and the results presented have not been submitted to any other University or Institute for the award of any other degree or diploma.

-------------------------------------

Dr.sandip karmakar

Ph.D. [IIT Kharagpur ]  
Dept : Computer Science and Engineering

Indian Institute of Information Technology, Kalyani

Declaration

I hereby declare that the work being presented in this thesis entitled, “Smart Education System”, submitted to Indian Institute of Information Technology Kalyani in partial fulfillment for the award of the degree of **Bachelor of Technology** in Computer Science and Engineering during the period from July, 2018 to Dec, 2018 under the supervision of Dr.Sanjay Chatterji , Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, West Bengal 741235, India, does not contain any classified information.

--------------------------------------------------

ROHIT KUMAR (Roll No.-CSE/16031/186)

Name of the Department: Computer Science and Engineering

Institute Name: Indian Institute of Information Technology, Kalyani, W.B

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

..................................

Dr.sandip karmakar

Ph.D. [IIT Kharagpur ]  
Dept : Computer Science and Engineering

Indian Institute of Information Technology, Kalyani

Acknowledgments

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Dr. sandip karmakar for his invaluable advice, guidance and his enormous patience throughout the development of the research project.

In addition, I would also like to express my gratitude to my loving parent and friends who had helped and given me encouragement.

---------------------------------------------------

ROHIT KUMAR (Roll No.-CSE/16031/186)

Name of the Department: Computer Science and Engineering

Institute Name: Indian Institute of Information Technology,

**Content**

Chapters ( Computer vision)

1.Introduction

2.Objective

3.Face Detection Intuition

* vilon-jones algorithm
* haar-like- Features
* integral image
* Training classifiers
* Boosting and AdaBoost

4. Object Detection Intuition

# 

Introduction Computer vision

Computer vision is an [interdisciplinary field](https://en.wikipedia.org/wiki/Interdisciplinarity) that deals with how computers can be made to gain high-level understanding from [digital images](https://en.wikipedia.org/wiki/Digital_image) or [videos](https://en.wikipedia.org/wiki/Video). From the perspective of [engineering](https://en.wikipedia.org/wiki/Engineering), it seeks to automate tasks that the [human visual system](https://en.wikipedia.org/wiki/Human_visual_system) can do. "Computer vision is concerned with the automatic extraction, analysis and understanding of useful information from a single image or a sequence of images. It involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding.  As a [scientific discipline](https://en.wikipedia.org/wiki/Scientific_discipline), computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, or multi-dimensional data from a [medical scanner](https://en.wikipedia.org/wiki/Medical_scanner).  As a technological discipline, computer vision seeks to apply its theories and models for the construction of computer vision systems

Objective

Computer vision is a subfield of artificial intelligence. The purpose of computer vision is to program a computer to "understand" a scene or features in an image.

# Typical goals of computer vision include:

1.The detection, segmentation, localisation, and recognition of certain objects in images (e.g., human faces)

2.The evaluation of results (e.g., segmentation, registration)

3.Registration of different views of the same scene or object

4.Tracking an object through an image sequence

5.Mapping a scene to a three-dimensional model of the scene; such a model might be used by a robot to navigate the imaged scene

6.Estimation of the three-dimensional pose of humans and their limbs

7.Searching for digital images by their content (content-based image retrieval)

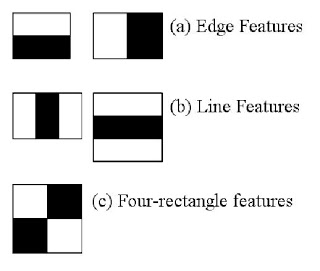
vilon-jones algorithm

**Face Detection** using **Viola**-**Jones Algorithm**. The **Viola**-**Jones algorithm** is a widely used mechanism for object detection Detection happens inside a detection window. A minimum and maximum window size is chosen, and for each size a sliding step size is chosen.

haar-like- Features

Haar-like features are digital image features used in object recognition. They owe their name to their intuitive similarity with Haar wavelets and were used in the first real-time face detector. Historically, working with only image intensities made the task of feature calculation computationally expensive.

# This worka all futures, (a)Edge Features (b) Line Features (c) Four-rectangle Features



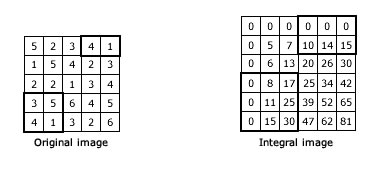
integral image

An integral image lets you calculate summations over image subregions. Rapidly.These summations are useful in many applications, like calculating HAAR wavelets. These are used in face recognition and other similar algorithms.

## How it works

Suppose an image is w pixels wide and h pixels high. Then the integral of this will be w+1 pixels wide and h+1pixels high. The first row and column of the integral image are all zeros.

All other pixels have a value equal to the sum of all pixels before it.



See the integral in the above image? Every pixel is the summation of the pixels before it (above and to the left).

Now, to calculate the summation of the pixels in the black box, you take the corresponding box in the integral. You sum as follows: (Bottom right + top left - top right - bottom left).

So for the 3,5,4,1 box, the calculations would go like this: (30+0-17-0 = 13). For the 4,1 box, it would be (0+15-10-0 = 5).

This way, you can calculate summations in rectangular regions rapidly.

# The parameters are, as always, self explanatory:

image: the source image

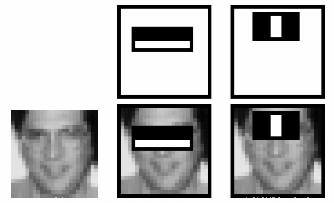
sum: the sum summation integral image

sqsum: the square sum integral image

tiled\_sum: image is rotated by 45 degrees and then its integral is calculated

Training classifiers

machine **learning** algorithm that uses a bunch of images of faces and non-faces to **train** a**classifier** that can later be used to detect faces in realtime. The algorithm implemented in **OpenCV** can also be used to detect other things, as long as you have the right **classifiers**.



Some limitations of the current visualisation tool

Only handles cascade classifier models, trained with the opencv\_traincascade tool, containing **stumps** as decision trees [default settings].

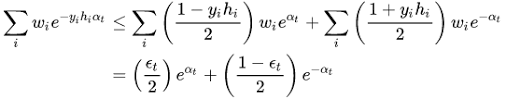
The image provided needs to be a sample window with the original model dimensions, passed to the --image parameter.

Example of the HAAR/LBP face model ran on a given window of Angelina Jolie, which had the same preprocessing as cascade classifier files –>24x24 pixel image, grayscale conversion and histogram equalisation:

Boosting Ensemble Method

# Training

AdaBoost refers to a particular method of training a boosted classifier. A boost classifier is a classifier in the form



{\displaystyle F\_{T}(x)=\sum \_{t=1}^{T}f\_{t}(x)\,\!}

[Boosting](https://en.wikipedia.org/wiki/Boosting_(machine_learning)) is a general ensemble method that creates a strong classifier from a number of weak classifiers.

This is done by building a model from the training data, then creating a second model that attempts to correct the errors from the first model. Models are added until the training set is predicted perfectly or a maximum number of models are added.

[AdaBoost](https://en.wikipedia.org/wiki/AdaBoost) was the first really successful boosting algorithm developed for binary classification. It is the best starting point for understanding boosting.

Modern boosting methods build on AdaBoost, most notably [stochastic gradient boosting machines](https://en.wikipedia.org/wiki/Gradient_boosting).

Cascading

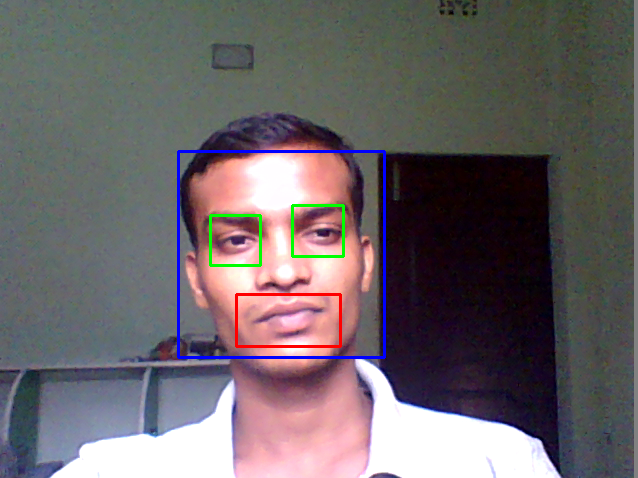
A **Haar Cascade** is basically a **classifier** which is used to detect the object for which it has been trained for, from the source. The **Haar Cascade** is trained by superimposing the positive image over a set of negative images. The training is generally done on a server and on various stages.

# How does cascading work

# C:\Users\Rohit kumar\Desktop\images (2).png

machine learning based approach where a **cascade** function **is** trained from a lot of positive and negative images. It **is** then used to detect objects in other images. The algorithm has four stages: **Haar** Feature Selection

This is a final result ,used cascade classifier haarcascade\_eye, haarcascade\_frontalface\_default and haarcascade\_smile

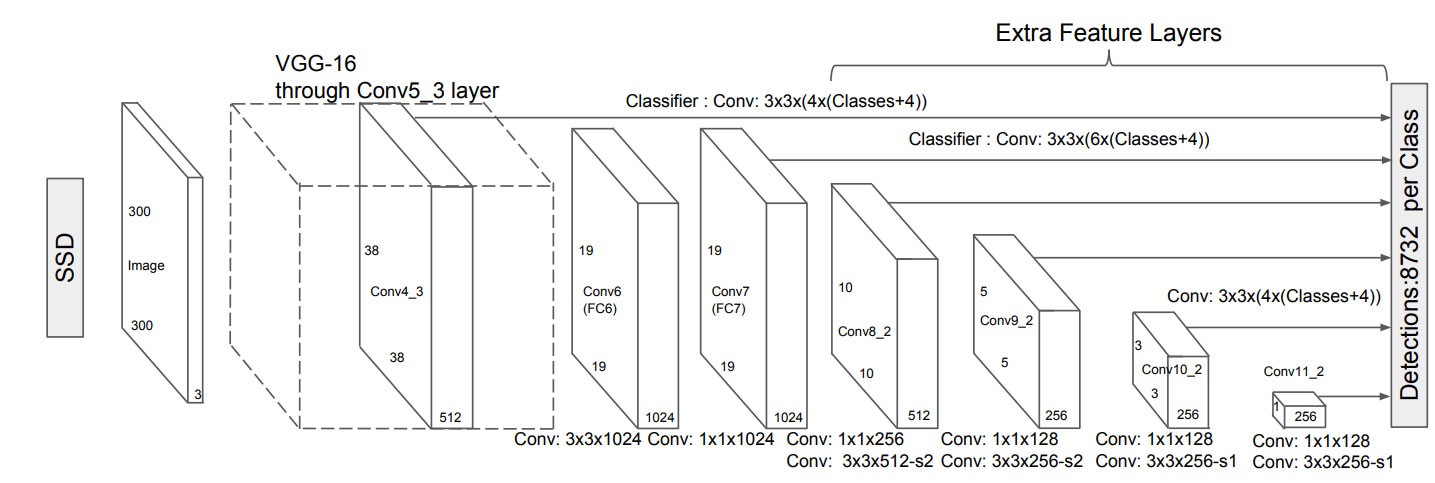


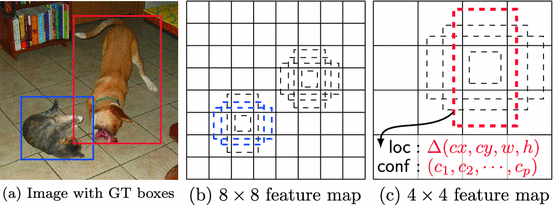
Object Detection Intuition

Object detection in computer vision. Object detection is the process of finding instances of real-world objects such as faces, bicycles, and buildings in images or videos. Object detection **algorithms** typically use extracted features and learning**algorithms** to recognize instances of an object category.

**SSD (Single Shot Detector)**

**SSD (Single Shot Detector)** is reviewed. By using SSD, we only need to **take one single shot to detect multiple objects within the image**, while regional proposal network (RPN) based approaches such as [R-CNN](https://medium.com/coinmonks/review-r-cnn-object-detection-b476aba290d1) series that need two shots, one for generating region proposals, one for detecting the object of each proposal. Thus, SSD is much faster compared with two-shot RPN-based approaches

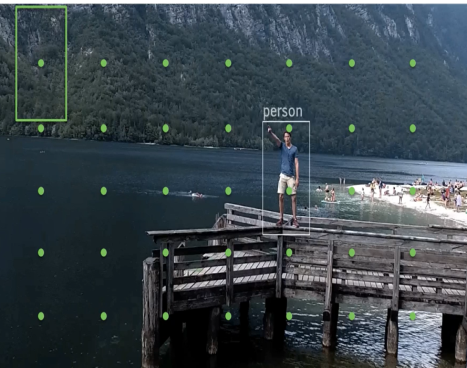
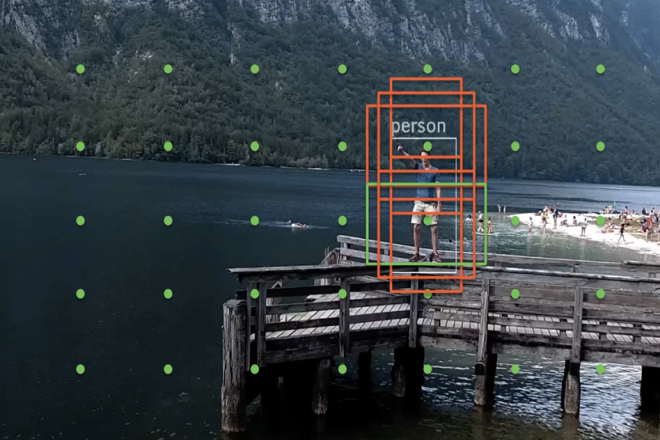
****

****

SSD framework. (a) SSD only needs an input image and ground truth boxes for each object during training. In a convolutional fashion, we evaluate a small set (e.g. 4) of default boxes of different aspect ratios at each location in several feature maps with different scales (e.g. 8 × 8 and 4 × 4 in (b) and (c)). For each default box, we predict both the shape offsets and the confidences for all object categories ((c1, c2, · · · , cp)). At training time, we first match these default boxes to the ground truth boxes. For example, we have matched two default boxes with the cat and one with the dog, which are treated as positives and the rest as negatives. The model loss is a weighted sum between localization loss (e.g. Smooth L1 [6]) and confidence loss (e.g. Softmax).

multi-box concept

* After going through a certain of convolutions for feature extraction, we obtain **a feature layer of size *m*×*n* (number of locations) with *p*channels**, such as 8×8 or 4×4 above. And a 3×3 conv is applied on this *m*×*n*×*p* feature layer.

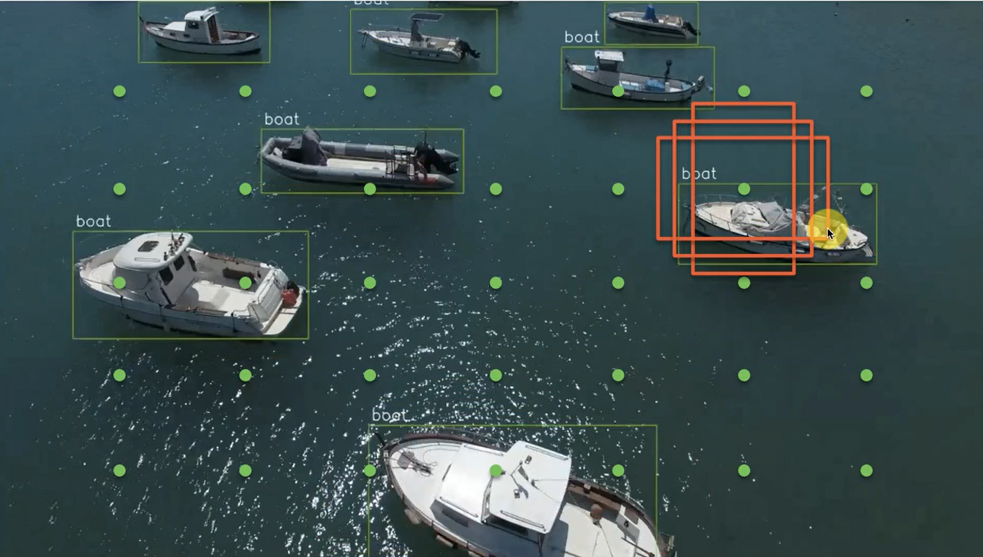
* **For each location, we got *k* bounding boxes.**These k bounding boxes have different sizes and aspect ratios. The concept is, maybe a vertical rectangle is more fit for human, and a horizontal rectangle is more fit for car.
* **For each of the bounding box, we will compute *c* class scores and 4 offsets relative to the original default bounding box shape.**
* Thus, we got **(*c*+4)*kmn* outputs.**

That’s why the paper is called “SSD: Single Shot **MultiBox** Detector”. But the above it’s just a part of SSD.

Predicting Object Positions

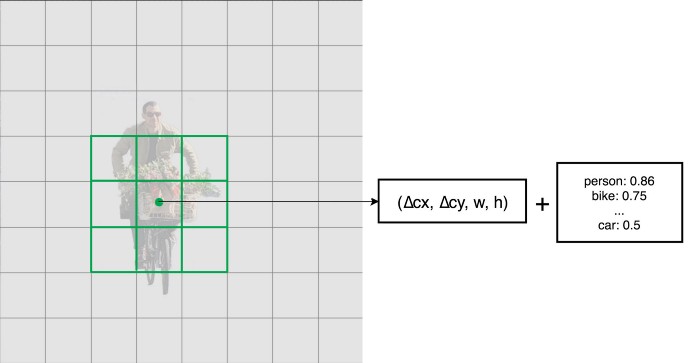
**Object detection** in computer vision. **Object detection** is the **process** of finding instances of real-world **objects** such as faces, bicycles, and buildings in **images** or videos. **Object detection** algorithms typically use extracted features and learning algorithms to recognize instances of an **object** category.



**Object detection** in computer vision. **Object detection** is the **process** of finding instances of real-world **objects** such as faces, bicycles, and buildings in **images** or videos. **Object detection** algorithms typically use extracted features and learning algorithms to recognize instances of an **object** category.

SSD does not use a delegated region proposal network. Instead, it resolves to a very simple method. It computes both the location and class scores using **small convolution filters**. After extracting the feature maps, SSD applies 3 × 3 convolution filters for each cell to make predictions. (These filters compute the results just like the regular CNN filters.) Each filter outputs 25 channels: 21 scores for each class plus one boundary box.

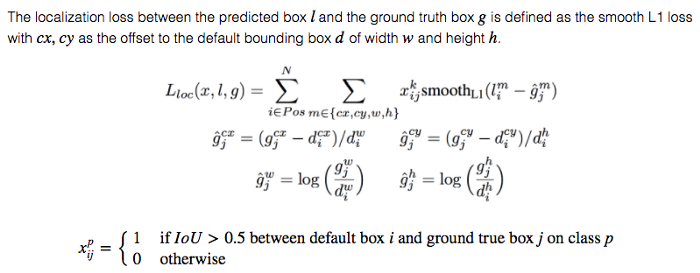


For example, in Conv4\_3, we apply four 3 × 3 filters to map 512 input channels to 25 output channels.

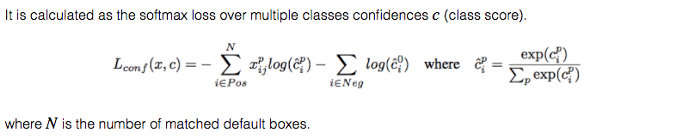
https://miro.medium.com/max/700/1*fxy_hGpFB8V9ZA3KYTGgfg.jpeg

Loss function

The **localization loss** is the mismatch between the ground truth box and the predicted boundary box. SSD only penalizes predictions from positive matches. We want the predictions from the positive matches to get closer to the ground truth. Negative matches can be ignored.



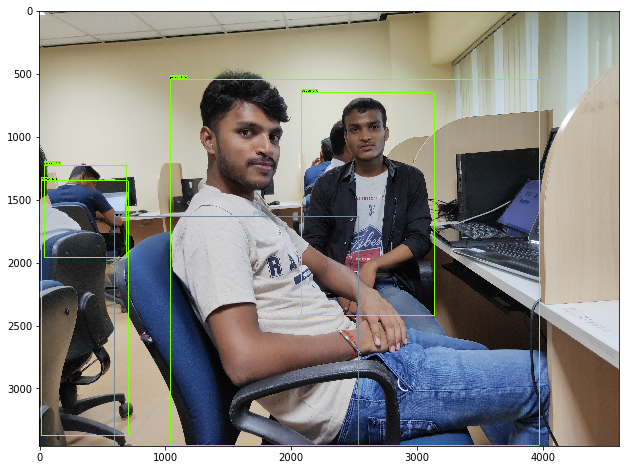
The **confidence loss** is the loss in making a class prediction. For every positive match prediction, we penalize the loss according to the confidence score of the corresponding class. For negative match predictions, we penalize the loss according to the confidence score of the class “0”: class “0” classifies no object is detected.



The final loss function is computes as:

https://miro.medium.com/max/573/1*I79wnE3675HYqcMKMm7yHg.pngwhere N is the number of positive

computer vision object\_detection\_tutorial – final output



object\_detection\_webcam-final output

