

**AMBEDKAR** **INSTITUTE** **OF** **TECHNOLOGY** **GOVT.** **OF** **NCT** **OF** **DELHI**

**Shakarpur,** **Delhi** **-** **110092**

**ITES&M** **6th** **SEMESTER**

**MAJOR** **PROJECT**

**Car and Pedestrian Detection**

**Submitted** **to** **:-**Dr. Dipankar Mishra

(HOD)

**Submitted** **by**: **-**

Sachin Kumar

Roll no: 1602111050

**Acknowledgement**

I would like to express my special thanks of gratitude to my HOD Dr. Dipankar Mishra as who gave me the golden opportunity to do this wonderful project on the topic of “MACHINE LEARNING NAME CAR & PEDESTRIAN DETECTION”, which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them.

Secondly I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

**Content**

1. Introduction to python
2. Data structure in python
3. Regular Expression
4. File Handling
5. Introduction to data Science
   * Introduction to Numpy and pandas.
   * Introduction to Matplotlib
6. Introduction to Machine Learning

* Introduction to sklearn

1. Introduction to OpenCV
2. Introduction to Haar cascade classifiers
3. Introduction to car and pedestrian
4. Application of car and Pedestrian detection
5. Source code and screenshot

**Introduction of Python**

**Open source:**Python is publicly available open source software, anyone can use source code that doesn't cost anything.

**Easy-to-learn:** Popular (scripting/extension) language, clear and easy syntax, no type declarations, automatic memory management, high-level data types and operations, design to read (more English like syntax) and write (shorter code compared to C, C++, and Java) fast.

**High-level Language:**   
High-level language (closer to human) refers to the higher level of concept from machine language (for example assembly languages). Python is an example of a high-level language like C, C++, Perl, and Java with low-level optimization.

**Portable:**  
High level languages are portable, which means they are able to run across all major hardware and software platforms with few or no change in source code. Python is portable and can be used on Linux, Windows, Macintosh, Solaris, FreeBSD, OS/2, Amiga, AROS, AS/400 and many more.

**Object-Oriented:** Python is a full-featured object-oriented programming language, with features such as classes, inheritance, objects, and overloading.

**Python is Interactive :**  
Python has an interactive console where you get a Python prompt (command line) and interact with the interpreter directly to write and test your programs. This is useful for mathematical programming.

**Interpreted:** Python programs are interpreted, takes source code as input, and then compiles (to portable byte-code) each statement and executes it immediately. No need to compiling or linking

**Extendable:** Python is often referred to as a "glue" language, meaning that it is capable to work in mixed-language environment. The Python interpreter is easily extended and can add a new built-in function or modules written in C/C++/Java code.

**Libraries:** Databases, web services, networking, numerical packages, graphical user interfaces, 3D graphics, others.

**Supports:** Support from online Python community

**Python Interpreter**

* In interactive mode, type Python programs and the interpreter displays the result:
* Type python into your terminal's command line
* After a short message, the >>> symbol will appear
* The above symbol signals the start of a Python interpreter's command line.
* Python interpreter evaluates inputs (For example >>> 4\*(6-2) return 16)

**How stable is Python?**

Very stable. New, stable releases have been coming out roughly every 6 to 18 months since 1991, and this seems likely to continue. Currently there are usually around 18 months between major releases.

The latest stable releases can always be found on the [Python download page](https://www.python.org/download/). There are two recommended production-ready versions at this point in time, because at the moment there are two branches of stable releases: 2.x and 3.x. Python 3.x may be less useful than 2.x, since currently there is more third party software available for Python 2 than for Python 3. Python 2 code will generally not run unchanged in Python 3.

**History**

The name Python was selected from "Monty Python Flying Circus" which was a British sketch comedy series created by the comedy group Monty Python and broadcast by the BBC from 1969 to 1974.

Python was created in the early 1980s by Guido van Rossum at the National Research Institute for Mathematics and Computer Science in Netherlands.

Python was created as a successor of a language called ABC (All Basic Code) and released publicly in1991. Guido remains Python's principal author, although it includes many contributions from active user community.

Between 1991 and 2001 there are several versions released, current stable release is 3.2. In 2001 the Python Software Foundation (PSF) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

**Major uses of Python**

* System utilities (system admin tools, command line programs).
* Web Development.
* Graphical User Interfaces (Tkinter, gtk, Qt).
* Internet scripting.
* Embedded scripting.
* Database access and programming.
* Game programming.
* Rapid prototyping and development.
* Distributed programming

**Organizations Using Python (sector wise)**

**Web Development:** Yahoo Maps, Yahoo Groups, Google, Zope Corporation, Ultraseek, Linux Weekly News, ElasticHosts Cloud Servers, Mojam.com, hunch, Shopzilla, Movieplayer.it, Multiplayer.it.

**Games:**Battlefield 2, Crystal Space, Star Trek Bridge Commander, The Temple of Elemental Evil, Vampire: The Masquerade: Bloodlines, Civilization 4, QuArK (Quake Army Knife)

**Graphics:**Industrial Light & Magic, Walt Disney Feature Animation, HKS, Inc. (ABAQUS/CAE), RoboFog, Caligari Corporation, Blender 3D, Jasc Software, Paint Shop Pro.

**Financial:** Altis Investment Management, ABN AMRO Bank, Treasury Systems, Bellco Credit Union, Journyx Timesheet and Resource Management Software.

**Science:** National Weather Service, Radar Remote Sensing Group, Applied Maths, Biosoft, The National Research Council of Canada, Los Alamos National Laboratory (LANL) Theoretical Physics Division, AlphaGene, Inc., LLNL, NASA, Swedish Meteorological and Hydrological Institute (SMHI), Environmental Systems Research Institute (ESRI), Objexx Engineering, Nmag Computational Micromagnetics

**Electronic Design Automation:**Ciranova, Productivity Design Tools, Object Domain, Pardus, Red Hat, SGI, Inc., MCI Worldcom, Nokia,

**Education**: University of California, Irvine, Smeal College of Business, The Pennsylvania State University, New Zealand Digital Library, IT Certification Exam preparation, SchoolTool,

**Business Software:**Raven Bear Systems Corporation, Thawte Consulting, Advanced Management Solutions Inc., IBM, Arakn<E9>, RealNetworks, dSPACE, Escom, The Tiny Company, Nexedi, Piensa Technologies - Bufete Consultor de Mexico, Nektra, WuBook.

**Is Python a good language for beginning programmers?**

Yes. It is still common to start students with a procedural and statically typed language such as Pascal, C, or a subset of C++ or Java. Students may be better served by learning Python as their first language. Python has a very simple and consistent syntax and a large standard library and, most importantly, using Python in a beginning programming course lets students concentrate on important programming skills such as problem decomposition and data type design. With Python, students can be quickly introduced to basic concepts such as loops and procedures. They can probably even work with user-defined objects in their very first course.

For a student who has never programmed before, using a statically typed language seems unnatural. It presents additional complexity that the student must master and slows the pace of the course. The students are trying to learn to think like a computer, decompose problems, design consistent interfaces, and encapsulate data. While learning to use a statically typed language is important in the long term, it is not necessarily the best topic to address in the students’ first programming course.

Many other aspects of Python make it a good first language. Like Java, Python has a large standard library so that students can be assigned programming projects very early in the course that do something. Assignments aren’t restricted to the standard four-function calculator and check balancing programs. By using the standard library, students can gain the satisfaction of working on realistic applications as they learn the fundamentals of programming. Using the standard library also teaches students about code reuse. Third-party modules such as PyGame are also helpful in extending the students’ reach.

**Python**

Python is a cross-platform programming language, meaning, it runs on multiple platforms like Windows, Mac OS X, Linux, Unix and has even been ported to the Java and .NET virtual machines. It is free and open source.

**Starting The Interpreter**

After installation, the python interpreter lives in the installed directory.

By default, it is /usr/local/bin/pythonX.X in Linux/Unix and C:\PythonXX in Windows, where the 'X' denotes the version number. To invoke it from the shell or the command prompt we need to add this location in the search path.

Search path is a list of directories (locations) where the operating system searches for executables. For example, in Windows command prompt, we can type set path=%path%;c:\python33 (python33 means version 3.3, it might be different in your case) to add the location to path for that particular session.

In Mac OS we need not worry about this as the installer takes care about the search path.

Now there are various ways to start Python.

**1. Immediate mode**

Typing python in the command line will invoke the interpreter in immediate mode. We can directly type in Python expressions and press enter to get the output.

>>>

is the Python prompt. It tells us that the interpreter is ready for our input. Try typing in 1 + 1 and press enter. We get 2 as the output. This prompt can be used as a calculator. To exit this mode type exit() or quit() and press enter.

**2. Script mode**

This mode is used to execute Python program written in a file. Such a file is called a **script**. Scripts can be saved to disk for future use. Python scripts have the extension .py, meaning that the filename ends with .py.

For example: helloWorld.py

To execute this file in script mode we simply write python helloWorld.py at the command prompt.

**Integrated Development Environment (IDE)**

We can use any text editing software to write a Python script file.

We just need to save it with the .py extension. But using an IDE can make our life a lot easier. IDE is a piece of software that provides useful features like code hinting, syntax highlighting and checking, file explorers etc. to the programmer for application development.

Using an IDE can get rid of redundant tasks and significantly decrease the time required for application development.

IDEL is a graphical user interface (GUI) that can be installed along with the Python programming language and is available from the official website.

We can also use other commercial or free IDE according to our preference. We have used the PyScripter IDE for our testing and we recommend the same. It is free and open source.

**Python Keywords**

Keywords are the reserved words in Python.

We cannot use a keyword as variable name, function name or any other identifier. They are used to define the syntax and structure of the Python language.

In Python, keywords are case sensitive.

There are 33 keywords in Python 3.3. This number can vary slightly in course of time.

All the keywords except True, False and None are in lowercase and they must be written as it is. The list of all the keywords are given below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Keywords in Python programming language | | | | |
| False | Class | Finally | is | return |
| None | Continue | For | lambda | Try |
| True | Def | From | nonlocal | while |
| And | Del | Global | not | With |
| As | Elif | If | or | yield |
| Assert | Else | Import | pass |  |
| Break | Except | In | raise |  |

Looking at all the keywords at once and trying to figure out what they mean might be overwhelming.

If you want to have an overview, here is the complete list of all the keywords with examples.

**Python Identifiers**

Identifier is the name given to entities like class, functions, variables etc. in Python. It helps differentiating one entity from another.

**Rules for writing identifiers**

1. Identifiers can be a combination of letters in lowercase (a to z) or uppercase (A to Z) or digits (0 to 9) or an underscore (\_). Names like myClass, var\_1 and print\_this\_to\_screen, all are valid example.
2. An identifier cannot start with a digit. 1variable is invalid, but variable1 is perfectly fine.
3. Keywords cannot be used as identifiers.
4. >>> global = 1
5. File "<interactive input>", line 1
6. global = 1
7. ^

SyntaxError: invalid syntax

1. We cannot use special symbols like !, @, #, $, % etc. in our identifier.
2. >>> a@ = 0
3. File "<interactive input>", line 1
4. a@ = 0
5. ^

SyntaxError: invalid syntax

1. Identifier can be of any length.

**Things to care about**

Python is a case-sensitive language. This means, Variable and variable are not the same. Always name identifiers that make sense.

While, c = 10 is valid. Writing count = 10 would make more sense and it would be easier to figure out what it does even when you look at your code after a long gap.

Multiple words can be separated using an underscore, this\_is\_a\_long\_variable.

We can also use camel-case style of writing, i.e., capitalize every first letter of the word except the initial word without any spaces. For example: camelCaseExample

**Python Statement**

Instructions that a Python interpreter can execute are called statements. For example, a = 1is an assignment statement. if statement, for statement, while statement etc. are other kinds of statements which will be discussed later.

**Multi-line statement**

In Python, end of a statement is marked by a newline character. But we can make a statement extend over multiple lines with the line continuation character (\). For example:

a = 1 + 2 + 3 + \

4 + 5 + 6 + \

7 + 8 + 9

This is explicit line continuation. In Python, line continuation is implied inside parentheses ( ), brackets [ ] and braces { }. For instance, we can implement the above multi-line statement as

a = (1 + 2 + 3 +

4 + 5 + 6 +

7 + 8 + 9)

Here, the surrounding parentheses ( ) do the line continuation implicitly. Same is the case with [ ] and { }. For example:

colors = ['red',

'blue',

'green']

We could also put multiple statements in a single line using semicolons, as follows

a = 1; b = 2; c = 3

**Python Indentation**

Most of the programming languages like C, C++, Java use braces { } to define a block of code. Python uses indentation.

A code block (body of a function, loop etc.) starts with indentation and ends with the first unindented line. The amount of indentation is up to you, but it must be consistent throughout that block.

Generally, four whitespaces are used for indentation and is preferred over tabs. Here is an example.

for i in range(1,11):

print(i)

if i == 5:

break

The enforcement of indentation in Python makes the code look neat and clean. This results into Python programs that look similar and consistent.

Indentation can be ignored in line continuation. But it's a good idea to always indent. It makes the code more readable. For example:

if True:

print('Hello')

a = 5

and

if True: print('Hello'); a = 5

both are valid and do the same thing. But the former style is clearer.

Incorrect indentation will result into IndentationError.

Data structure in python

1. A data structure represents logical arrangement of elements in memory in a particular model, Data structures are also known as abstract data types(ADTs).
2. Stacks, linked lists and queues are important data structure which are most used in software.
3. To create stacks, linked and queues are important data structures which are most used in software.
4. A linked list is a set of nodes such that each node contains a data field to store data and two link fields to refer to the previous node and next node.
5. Insertion, deletion and replacing the elements are important operation in case of a linked list.
6. A stack represents a group of elements arranged in memory in LIFO (LAST IN FIRST OUT) manner.
7. Push, pop and peep (or peek) operations are important in case of stacks.
8. A queue is a data structure where the first element which entered the queue will come out first, this is called FIFO (FIRST IN FIRST OUT) order.
9. In queue, the elements are added only at the rear (or back) side of the queue and they are removed from the front of the queue.
10. Adding the elements and removing the elements are the two important operations on queues.
11. A double-ended queue (or deque) is a queue where elements can be inserting or deleted from both ends.
12. Deques are more efficient than the normal queues in terms of memory usage and speed.
13. Adding elements at the front and at the rear, deleting the elements at the front and rear are the two important operations that one can perform on a deque.

**File Handling**

To store data in a computer, we need files.

* File:- The data stored on a secondary storage media like hard disk or CD is called a file.
* Once the data is stored in a file, the same data can be shared by several programs.

Advantages of storing data in a file.

1. When the data is stored in a file, it stored permanently. This mean that even though the computer is switched off, the data is not removed from the memory since the file is sorted on hard disk or CD.
2. It is possible to update the file data. For example, we can add new data to the existing file. delete unnecessary data from the file and modify the available data of the file. This makes the file more useful.
3. Once the data is stored in a file, the same data can be shared by various programs.
4. Files are highly useful to store huge amount of data. For example, Voter list or census data.

Types of files

1. Text files
2. Binary files

Text files: text file store the data in the form of characters.

Binary files: Binary files store entire data in the form of bytes,

i.e. a group of 8 bit each. Binary files can be used to store text, image, audio and video.

Working with binary files

f1 = open('fuji.jpg','rb')  
f2 = open('laptop.jpg','wb')  
  
# read byte file f1 and write into f2  
  
bytes = f1.read()  
f2.write(bytes)  
  
f1.close()  
f2.close()

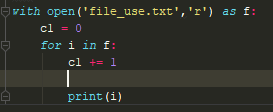
Opening file format

|  |  |
| --- | --- |
| w | To write data into the file. |
| r | To read data from a file. |
| a | To append data to the file. |
| w+ | To write and read data of a file. |
| r+ | To read and write data of a file. |
| a+ | To append and read data of a file. |
| x | To open the file in exclusive creation mode. The file creation fails if the file already  Exists. |

The with statement: The ‘with’ statement can be used while opening a file.

The advantage of with statement is that it will take care of closing a file which is open by it.

Hence, we need not close the file explicitly.



Pickle (Serialization):- is the concept of storing objects into a binary file (abc.dat).

* Pickle is a process of converting a class object into a byte stream so that it can be stored into a file.
* This is also called serialization.
* Pickling is done using the dump() method of ‘pickle’ module as:

Pickle.dump(object,file)

Unpickle (de-serialization):- is the process whereby a byte stream is converted back into a class object.

* It means, Unpickling represents reading the class objects from the file.
* It is also called de-serialization.
* Unpickling is done using the load() method of ‘pickle’ as:

Object = pickle.load(file)

Seek():- In binary file, to move the file pointer to any position, we can use seek() method.

Tell() :- in binary file, to know the position of the file pointer, we can use tell() method.

encode() :- The encode() method converts the string into bytes so that it can be written into a binary file.

decode() :- The decode() method use to convert binary string into ordinary string.

Random accessing:- Accessing the contents of file randomly by moving the file pointer to any byte in the file is called random accessing. Seek() and tell() methods are used in random accessing of a file.

Zipping and unzipping of file can be done using ZipFile class of zipfile module.

**REGULAR EXPRESSION**

Regular expression: - a regular expression is a string that contains special symbols and characters to find and extract the information needed by us from the given data.

Generally, we should use raw strings that start with ‘r’ in creation of regular expressions. We can also use normal strings but they are not advisable as they retain the meaning of escape characters.

Regular expressions are used to perform the following important operations:

1. Matching strings: match()
2. Searching for strings: search()
3. Finding all strings: findall()
4. Splitting a string into pieces: split()
5. Replacing string: sub()

Sequence Characters in Regular Expressions

|  |  |
| --- | --- |
| Char | Its description |
| \d | Represent any digit{0 to 9} |
| \D | Represents any non-digit |
| \s | Represents white space. Ex :- \t\n\r\f\v |
| \S | Represents non- white space character |
| \w | Represents any alphanumeric (a to z, a to z, 0 to 9 ) |
| \W | Represents non- alphanumeric |
| \b | Represents a space around words |
| \A | Matches only at start of the string |
| \Z | Matches only at end of the string |

Quantifier: In regular expression, some characters represent more than one character to be matched in the string. Such characters are called ‘quantifiers’

Quantifiers in regular expression

|  |  |
| --- | --- |
| Character | Its description |
| + | 1 or more repetitions of the preceding regexp |
| \* | 0 or more repetitions of the preceding regexp |
| ? | 0 or 1 repetitions of the preceding regexp |
| {m} | Exactly m occurrences |
| {m,n} | From m to n . m default to 0, n to infinity |

Special Character in regular Expressions

|  |  |
| --- | --- |
| Character | Its description |
| \ | Escape special character nature |
| . | Matches any character except new line |
| ^ | Matches beginning of a string |
| $ | Matches ending of a string |
| […..] | Denotes a set of possible characters. Ex: [6b-d] matches any characters ‘6’, ‘b’,’ c’ or ‘d’ |
| [^……] | Matches every character except the ones inside brackets. Ex[^a-c6]  Matches any character except ‘a’,’b’,’c’ and ‘6’ |
| (……) | Matches the regular expression inside the parentheses and the result can be captured |
| R|S | Matches either regex R or regx S |

# **Data Science Introduction**

Data science is the process of deriving knowledge and insights from a huge and diverse set of data through organizing, processing and analyzing the data. It involves many different disciplines like mathematical and statistical modelling, extracting data from it source and applying data visualization techniques. Often it also involves handling big data technologies to gather both structured and unstructured data. Below we will see some example scenarios where Data science is used.

## Python in Data Science

The programming requirements of data science demands a very versatile yet flexible language which is simple to write the code but can handle highly complex mathematical processing. Python is most suited for such requirements as it has already established itself both as a language for general computing as well as scientific computing. Moreover, it is being continuously upgraded in form of new addition to its plethora of libraries aimed at different programming requirements. Below we will discuss such features of python which makes it the preferred language for data science.

**Introduction to Numpy and Pandas**

NumPy is a Python package which stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

## Operations using NumPy

Using NumPy, a developer can perform the following operations −

* Mathematical and logical operations on arrays.
* Fourier transforms and routines for shape manipulation.
* Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

## NumPy – A Replacement for MatLab

NumPy is often used along with packages like **SciPy** (Scientific Python) and **Mat−plotlib** (plotting library). This combination is widely used as a replacement for MatLab, a popular platform for technical computing. However, Python alternative to MatLab is now seen as a more modern and complete programming language.

It is open source, which is an added advantage of NumPy.

**Introduction to Pandas**

Pandas is an open-source Python Library used for high-performance data manipulation and data analysis using its powerful data structures. Python with pandas is in use in a variety of academic and commercial domains, including Finance, Economics, Statistics, Advertising, Web Analytics, and more. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, organize, manipulate, model, and analyses the data.

## Key Features of Pandas

* Fast and efficient Data Frame object with default and customized indexing.
* Tools for loading data into in-memory data objects from different file formats.
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of date sets.
* Label-based slicing, indexing and subletting of large data sets.
* Columns from a data structure can be deleted or inserted.
* Group by data for aggregation and transformations.
* High performance merging and joining of data.
* Time Series functionality.

**Introduction to Matplotlib**

Matplotlib is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. It supports a very wide variety of graphs and plots namely - histogram, bar charts, power spectra, error charts etc.

Matplotlib Example

The following script produces the **sine wave plot** using matplotlib.

Example

import numpy as np

import matplotlib.pyplot as plt

# Compute the x and y coordinates for points on a sine curve

x = np.arange(0, 3 \* np.pi, 0.1)

y = np.sin(x)

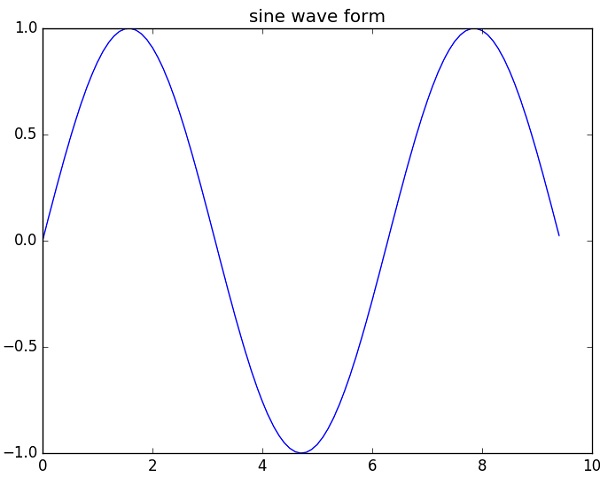
plt.title("sine wave form")

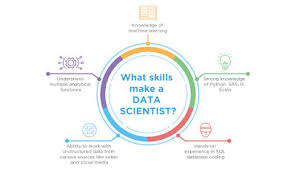
# Plot the points using matplotlib

plt.plot(x, y)

plt.show()

Its **output** is as follows −





# **Machine Learning with Python Tutorial**

## Python in Machine Learning

Python has libraries that enables developers to use optimized algorithms. It implements popular machine learning techniques such as recommendation, classification, and clustering. Therefore, it is necessary to have a brief introduction to machine learning before we move further.

## What is Machine Learning?

Data science, machine learning and artificial intelligence are some of the top trending topics in the tech world today. Data mining and Bayesian analysis are trending and this is adding the demand for machine learning. This tutorial is your entry into the world of machine learning.

Machine learning is a discipline that deals with programming the systems so as to make them automatically learn and improve with experience. Here, learning implies recognizing and understanding the input data and taking informed decisions based on the supplied data. It is very difficult to consider all the decisions based on all possible inputs.

## Applications of Machine Learning Algorithms

The developed machine learning algorithms are used in various applications such as −

* Vision processing
* Language processing
* Forecasting things like stock market trends, weather
* Pattern recognition
* Games
* Data mining
* Expert systems
* Robotics

## Steps Involved in Machine Learning

A machine learning project involves the following steps −

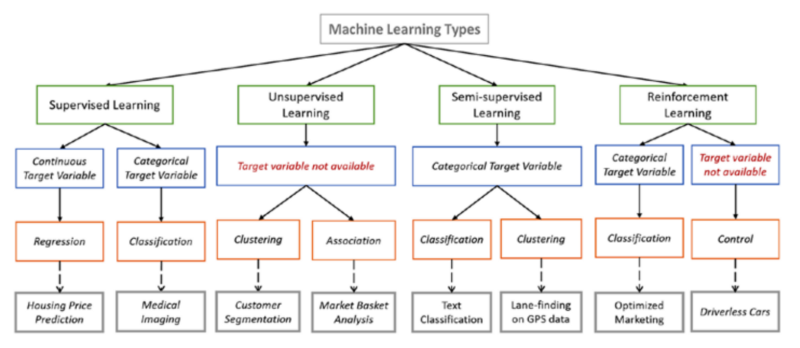
* Defining a Problem
* Preparing Data
* Evaluating Algorithms
* Improving Results
* Presenting Results

## Concepts of Learning

Learning is the process of converting experience into expertise or knowledge.

Learning can be broadly classified into three categories, as mentioned below, based on the nature of the learning data and interaction between the learner and the environment.

* Supervised Learning
* Unsupervised Learning
* Reinforcement learning



**Introduction to OpenCV**

OpenCV is a cross-platform library using which we can develop real-time **computer vision applications**. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

Let’s start the chapter by defining the term "Computer Vision".

## Computer Vision

Computer Vision can be defined as a discipline that explains how to reconstruct, interrupt, and understand a 3D scene from its 2D images, in terms of the properties of the structure present in the scene. It deals with modeling and replicating human vision using computer software and hardware.

Computer Vision overlaps significantly with the following fields −

* **Image Processing** − It focuses on image manipulation.
* **Pattern Recognition** − It explains various techniques to classify patterns.
* **Photogrammetry** − It is concerned with obtaining accurate measurements from images.

### Computer Vision Vs Image Processing

**Image processing** deals with image-to-image transformation. The input and output of image processing are both images.

**Computer vision** is the construction of explicit, meaningful descriptions of physical objects from their image. The output of computer vision is a description or an interpretation of structures in 3D scene.

## Applications of Computer Vision

Here we have listed down some of major domains where Computer Vision is heavily used.

### Robotics Application

* Localization − Determine robot location automatically
* Navigation
* Obstacles avoidance
* Assembly (peg-in-hole, welding, painting)
* Manipulation (e.g. PUMA robot manipulator)
* Human Robot Interaction (HRI) − Intelligent robotics to interact with and serve people

### Medicine Application

* Classification and detection (e.g. lesion or cells classification and tumor detection)
* 2D/3D segmentation
* 3D human organ reconstruction (MRI or ultrasound)
* Vision-guided robotics surgery

### Industrial Automation Application

* Industrial inspection (defect detection)
* Assembly
* Barcode and package label reading
* Object sorting
* Document understanding (e.g. OCR)

### Security Application

* Biometrics (iris, finger print, face recognition)
* Surveillance − Detecting certain suspicious activities or behaviors

### Transportation Application

* Autonomous vehicle
* Safety, e.g., driver vigilance monitoring

## Features of OpenCV Library

Using OpenCV library, you can −

* Read and write images
* Capture and save videos
* Process images (filter, transform)
* Perform feature detection
* Detect specific objects such as faces, eyes, cars, in the videos or images.
* Analyze the video, i.e., estimate the motion in it, subtract the background, and track objects in it.

OpenCV was originally developed in C++. In addition to it, Python and Java bindings were provided. OpenCV runs on various Operating Systems such as windows, Linux, OSx, FreeBSD, Net BSD, Open BSD, etc.

## OpenCV Library Modules

Following are the main library modules of the OpenCV library.

### Image Processing

This module covers various image processing operations such as image filtering, geometrical image transformations, color space conversion, histograms, etc. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.imgproc**.

### Video

This module covers the video analysis concepts such as motion estimation, background subtraction, and object tracking. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.video**.

### Video I/O

This module explains the video capturing and video codecs using OpenCV library. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.videoio**.

### calib3d

This module includes algorithms regarding basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence and elements of 3D reconstruction. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.calib3d**.

### features2d

This module includes the concepts of feature detection and description. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.features2d**.

### Objdetect

This module includes the detection of objects and instances of the predefined classes such as faces, eyes, mugs, people, cars, etc. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.objdetect**.

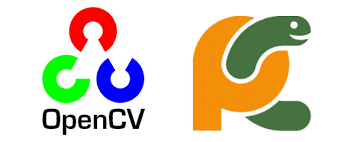
### Highgui

This is an easy-to-use interface with simple UI capabilities. In the Java library of OpenCV, the features of this module is included in two different packages namely, **org.opencv.imgcodecs** and **org.opencv.videoio**.

## A Brief History of OpenCV

OpenCV was initially an Intel research initiative to advise CPU-intensive applications. It was officially launched in 1999.

* In the year 2006, its first major version, OpenCV 1.0 was released.
* In October 2009, the second major version, OpenCV 2 was released.
* In August 2012, OpenCV was taken by a nonprofit organization OpenCV.org.



**Pedestrian detection**

is an essential and significant task in any intelligent [video surveillance](https://en.wikipedia.org/wiki/Video_surveillance) system, as it provides the fundamental information for [semantic](https://en.wikipedia.org/wiki/Semantic) understanding of the [video](https://en.wikipedia.org/wiki/Video) footages. It has an obvious extension to automotive applications due to the potential for improving safety systems. Many car manufacturers (e.g. Volvo, Ford, GM, Nissan) offer this as an [ADAS](https://en.wikipedia.org/wiki/Advanced_driver-assistance_systems) option in 2017.

## Challenges

* Various style of clothing in appearance
* Different possible articulations
* The presence of occluding accessories
* Frequent occlusion between pedestrians

Despite the challenges, pedestrian detection still remains an active research area in [computer vision](https://en.wikipedia.org/wiki/Computer_vision) in recent years. Numerous approaches have been proposed.

### Holistic detection

* Detectors are trained to search for pedestrians in the video frame by scanning the whole frame. The detector would “fire” if the image features inside the local search window meet certain criteria. Some methods employ global features such as edge template ,[[1]](https://en.wikipedia.org/wiki/Pedestrian_detection#cite_note-1) others uses local features like [histogram of oriented gradients](https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients) [[2]](https://en.wikipedia.org/wiki/Pedestrian_detection#cite_note-2)descriptors. The drawback of this approach is that the performance can be easily affected by background clutter and occlusions.

### Part-based detection[[edit](https://en.wikipedia.org/w/index.php?title=Pedestrian_detection&action=edit&section=4)]

* Pedestrians are modeled as collections of parts. Part hypotheses are firstly generated by learning local features, which include edgelet[[3]](https://en.wikipedia.org/wiki/Pedestrian_detection#cite_note-3) and orientation features.[[4]](https://en.wikipedia.org/wiki/Pedestrian_detection#cite_note-4) These part hypotheses are then joined to form the best assembly of existing pedestrian hypotheses. Though this approach is attractive, part detection itself is a difficult task. Implementation of this approach follows a standard procedure for processing the image data that consists of first creating a densely sampled image pyramid, computing features at each scale, performing classification at all possible locations, and finally performing non-maximal suppression to generate the final set of bounding boxes.[[5]](https://en.wikipedia.org/wiki/Pedestrian_detection#cite_note-5)

### Patch-based detection[[edit](https://en.wikipedia.org/w/index.php?title=Pedestrian_detection&action=edit&section=5)]

* Recently Leibe et al.[[6]](https://en.wikipedia.org/wiki/Pedestrian_detection#cite_note-6) proposed an approach combining both the detection and segmentation with the name Implicit Shape Model (ISM). A codebook of local appearance is learned during the training process. In the detecting process, extracted local features are used to match against the codebook entries, and each match casts one vote for the pedestrian hypotheses. Final detection results can be obtained by further refining those hypotheses. The advantage of this approach is only a small number of training images are required.

# **Histogram of oriented gradients**

The **histogram of oriented gradients (HOG)** is a [feature descriptor](https://en.wikipedia.org/wiki/Feature_descriptor) used in [computer vision](https://en.wikipedia.org/wiki/Computer_vision) and [image processing](https://en.wikipedia.org/wiki/Image_processing) for the purpose of [object detection](https://en.wikipedia.org/wiki/Object_detection). The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of [edge orientation histograms](https://en.wikipedia.org/w/index.php?title=Edge_orientation_histogram&action=edit&redlink=1), [scale-invariant feature transform](https://en.wikipedia.org/wiki/Scale-invariant_feature_transform) descriptors, and [shape contexts](https://en.wikipedia.org/wiki/Shape_context), but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy

# **Haar-like feature**

**Haar-like features** are [digital image](https://en.wikipedia.org/wiki/Digital_image) [features](https://en.wikipedia.org/wiki/Feature_(computer_vision)) used in [object recognition](https://en.wikipedia.org/wiki/Object_recognition). They owe their name to their intuitive similarity with [Haar wavelets](https://en.wikipedia.org/wiki/Haar_wavelet) and were used in the first real-time face detector.[[1]](https://en.wikipedia.org/wiki/Haar-like_feature#cite_note-Viola_2001-1)

Historically, working with only image intensities (i.e., the [RGB](https://en.wikipedia.org/wiki/RGB) [pixel](https://en.wikipedia.org/wiki/Pixel) values at each and every pixel of image) made the task of feature calculation [computationally expensive](https://en.wikipedia.org/wiki/Computationally_expensive). A publication by Papageorgiou et al.[[2]](https://en.wikipedia.org/wiki/Haar-like_feature#cite_note-2) discussed working with an alternate feature set based on Haar wavelets instead of the usual image intensities. Viola and Jones[[1]](https://en.wikipedia.org/wiki/Haar-like_feature#cite_note-Viola_2001-1) adapted the idea of using Haar wavelets and developed the so-called Haar-like features. A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums.

For example, let us say we have an image database with human [faces](https://en.wikipedia.org/wiki/Face). It is a common observation that among all faces the region of the eyes is darker than the region of the cheeks. Therefore, a common Haar feature for face detection is a set of two adjacent rectangles that lie above the eye and the cheek region. The position of these rectangles is defined relative to a detection window that acts like a bounding box to the target object (the face in this case).

The key advantage of a Haar-like feature over most other features is its calculation speed. Due to the use of [*integral images*](https://en.wikipedia.org/wiki/Integral_image), a Haar-like feature of any size can be calculated in constant time (approximately 60 microprocessor instructions for a 2-rectangle feature).

## Rectangular Haar-like features

A simple rectangular Haar-like feature can be defined as the difference of the sum of pixels of areas inside the rectangle, which can be at any position and scale within the original image. This modified feature set is called *2-rectangle feature*. Viola and Jones also defined 3-rectangle features and 4-rectangle features. The values indicate certain characteristics of a particular area of the image. Each feature type can indicate the existence (or absence) of certain characteristics in the image, such as edges or changes in texture. For example, a 2-rectangle feature can indicate where the border lies between a dark region and a light region.

**Abstract** – Robust and efficient vehicle detection from images is an important task in Intelligent Transportation Systems. With the

development of computer vision techniques and consequent accessibility of video image data, new applications have been enabled to

on-road vehicle detection algorithms

**Vehicle Detection Approaches**

To be useful, vehicle detection methods need to be fast enough to operate in real-time, be insensitive to illumination

change and different weather conditions, and be able to separate vehicles from images sequences in an accurate and

**efficient manner**. With the deployment of video cameras, vehicle detection can be categorized as feature-based

methods and model-based methods.

**Vehicle Detection Approaches**

To be useful, vehicle detection methods need to be fast enough to operate in real-time, be insensitive to illumination

change and different weather conditions, and be able to separate vehicles from images sequences in an accurate and

efficient manner. With the deployment of video cameras, vehicle detection can be categorized as feature-based

methods and model-based methods.

# **Pedestrian and Car detection**

The aim of this project is to develop an application which can detect pedestrians effectively. The problem of motion-based object detection can be divided into two parts:  
a) Classifying pedestrians and non pedestrians features  
a) Detecting pedestrians in each frame  
b) Associating the detections corresponding to the same object over time

**Tool: This** project is based on Machine learning, We can provide image data set of pedestrians and non-pedestrians as an training data to the software tool which will extract important features using An object detection method that inputs **Haa**r features into a series of classifiers (cascade) to identify objects in an image. We can use Python or Machine Learning as a building tool for this system.

**Implementation :** The Implementation of such a tool depends on two factors – Feature extraction and object detection methods.  
So we use HAAR [classifiers](https://in.mathworks.com/matlabcentral/fileexchange/?utf8=%E2%9C%93&term=Image+Classifier)  and also read about basic feature extraction algorithm.

**Research:** Detecting humans in images is a challenging task owing to their variable appearance. This is a booming research topic which is still going on for surveillance of large crowds in real time applications. Research areas include image processing, artificial Intelligence and machine learning.

The objective of the program given is to detect object of interest(Car) in video frames and to keep tracking the same object. This is an example of how to detect vehicles in Python.

**Why Vehicle Detection?**

* The startling losses both in human lives and finance caused by vehicle accidents.
* Detecting vehicles in images acquired from a moving platform is a challenging problem.

#### **Haar Cascades for Human Detection**

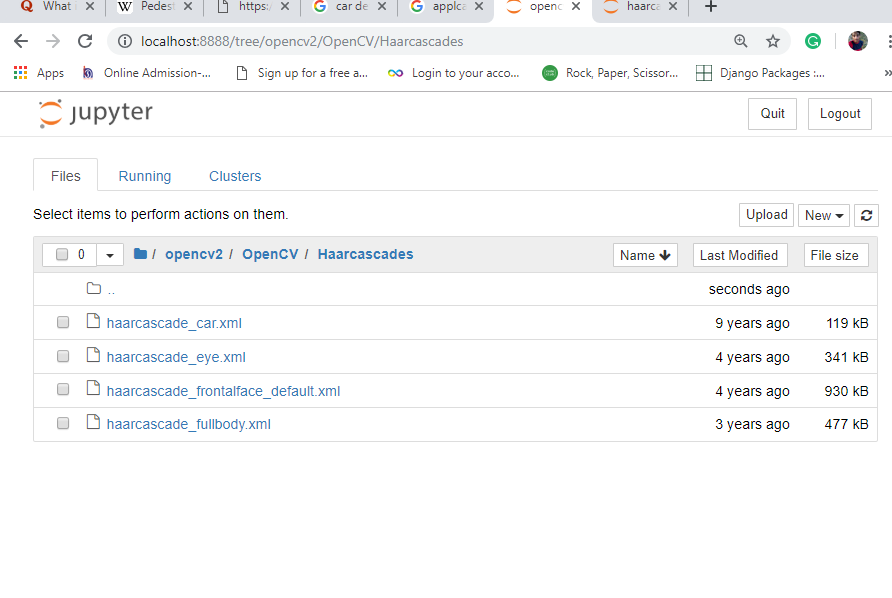
Haar feature based approach for object detection is proposed by **Paul Viola** and **Michael Jones** in their paper “[**Rapid Object Detection using a Boosted Cascade of Simple Features**](https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf)**”** published in 2001. This approach is widely used for Face Detection.

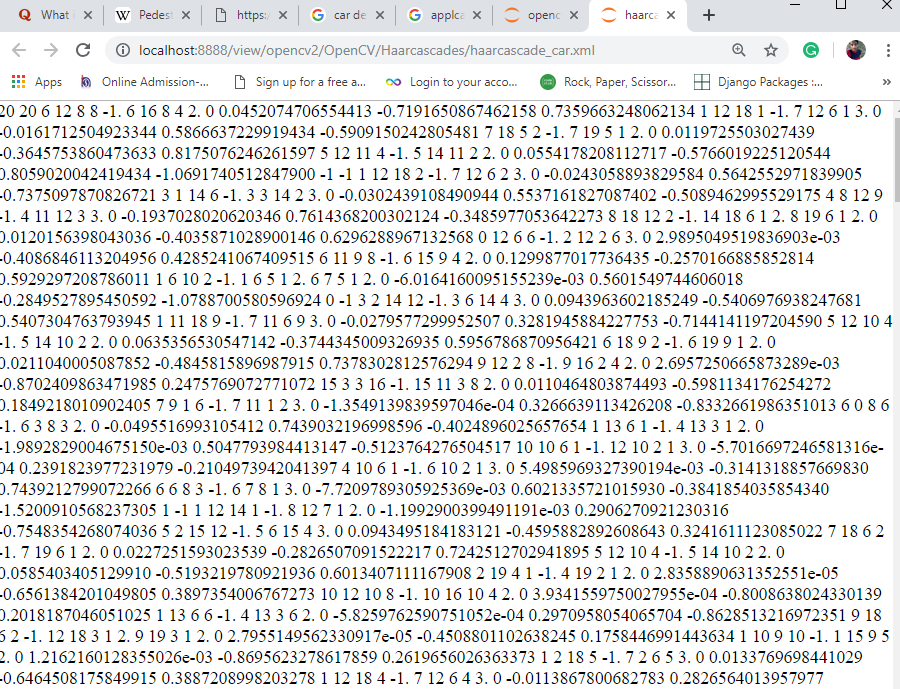
**OpenCV** includes inbuilt functionality to provide Haar cascade based object detection. **Pre-trained models**provided by OpenCV for “Full Body Detection”, “Upper Body Detection” and “Lower Body Detection” are available [here](https://github.com/opencv/opencv/tree/master/data/haarcascades).

[**This Python code snippet**](https://gist.github.com/madhawav/cd913f68b9405b438833b614ccb49b57) shows application of Haar cascade for Human Detection using Open CV 3.4. It shows a frame time of approximately 90 — 100 milliseconds per frame (equivalent to 11 frames-per-second) in my test bench.

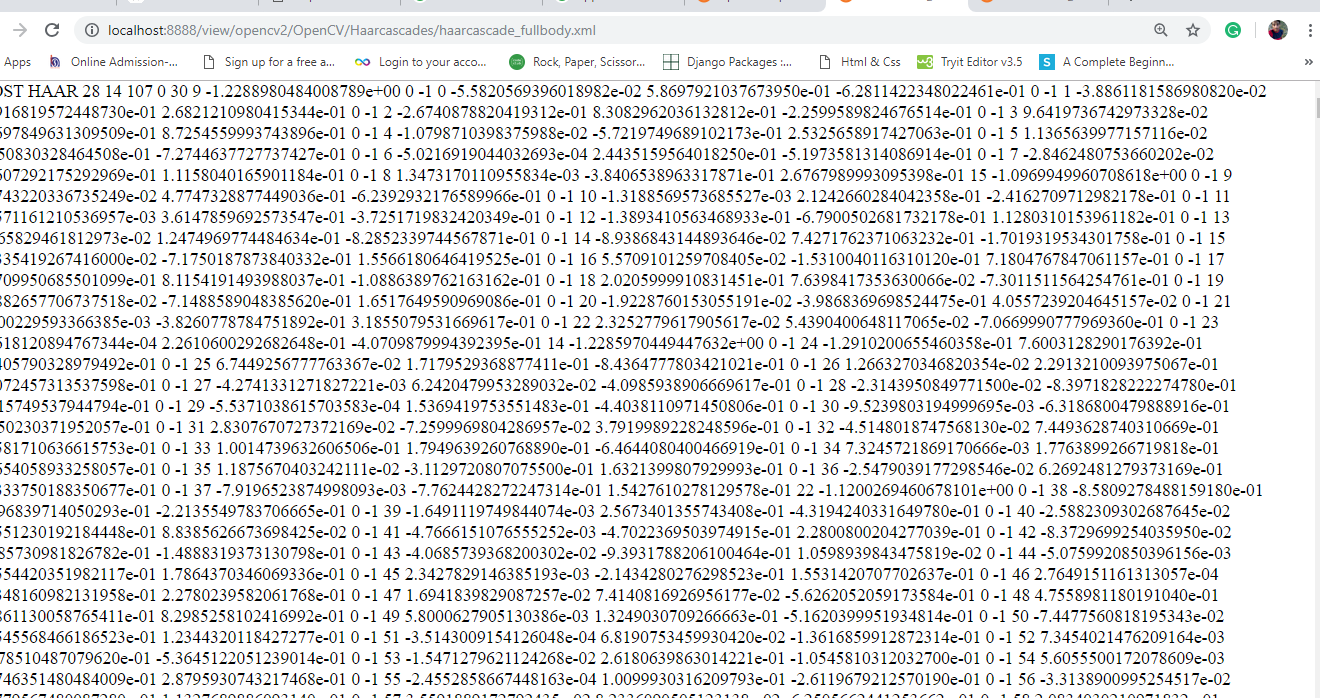
**Screenshot of project**

Haar cascade: dir





XMl for car detection



Project source code: