CHAPTER 1
INTRODUCTION

1.1 OBJECT DETECTION

Object detection involves detection instances of objects from a specific category in a picture. The goal of object sight on is to detect all instances of objects from a renowned category, like folks, cars or faces in a picture. Typically solely a little range of instances of the thing area unit gift within the image, but there is a very large number of possible locations and scales at which they can occur which got to somehow be explored. Each detection is according with some sort of create data. This could be as easy because the location of the thing, a location and scale, or the extent of the object defined in terms of a bounding box.

In alternative things the create data is a lot of elaborated and contains the parameters of a linear or non-linear transformation. For example a face detector could reckon the locations of the eyes, nose and mouth, in addition to the bounding box of the face. Object detection ways fall under 2 major classes, generative [1,2,3,4,5] and discriminative [6,7,8,9,10]. The first consists of a chance model for the create variability of the objects along with AN look model, a probability model for the image appearance conditional on a given pose, along with a model for background, i.e. non-object images.

The model parameters are often calculable from coaching information and also the selections area unit supported ratios of posterior possibilities. The second usually builds a classifier which will discriminate between pictures (or sub-images) containing the thing and people not containing the thing. The parameters of the classifier area unit hand-picked to reduce mistakes on the coaching information, typically with a regularization bias to avoid overfitting.

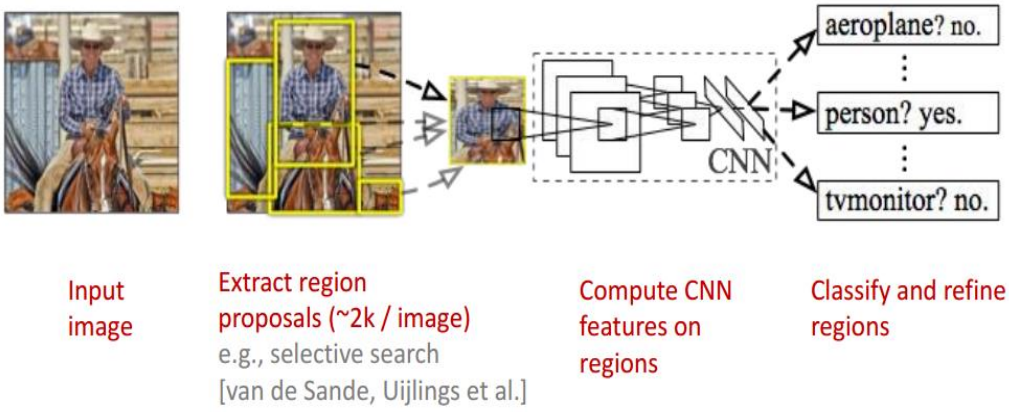


Fig 1.1 Object Detection

1.2 MACHINE LEARNING

Machine learning is an application of computing (AI) that gives systems the flexibility to mechanically learn and improve from expertise while not being expressly programmed. Machine learning focuses on the event of pc programs which will access information and use it learn for themselves. The process of learning starts with observations or information, like examples, direct expertise, or instruction, so also seem for patterns in information and build higher choices in the future supported the examples that offer below. The primary aim is to permit the computers learn mechanically while not human intervention or help and modify actions consequently.

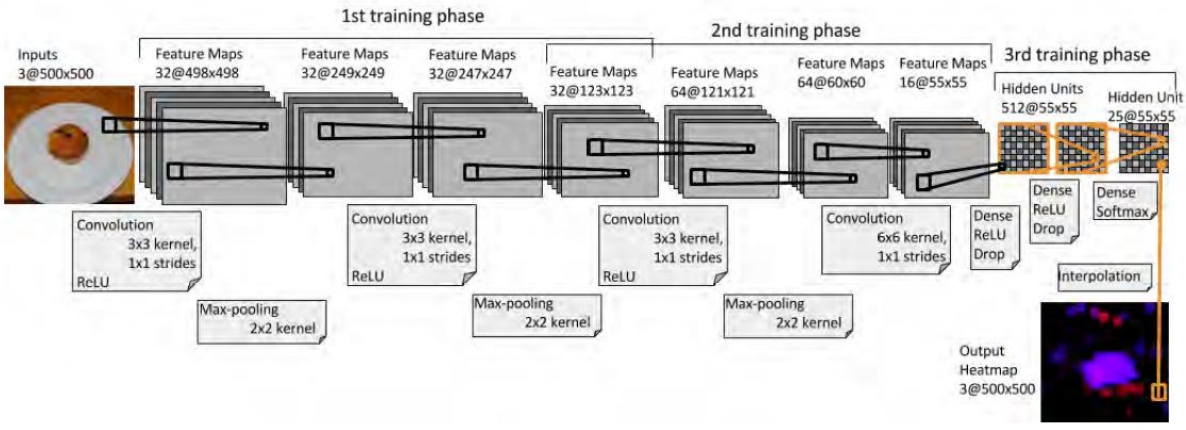


Fig 1.2 Machine Learning Training Phase

1.3 DEEP LEARNING

Deep Learning has evolved hand-in-hand with the digital era, that has caused associate explosion of informational together forms and from each region of the globe. This data, familiar merely as huge information, is drawn from sources like social media, net search engines, e-commerce platforms, on-line cinemas and a lot of. This monumental quantity of information is instantly accessible and may be shared through fintech applications like cloud computing. However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unraveling this wealth of information, and are increasingly adapting to Artificial Intelligence (AI) systems for automated support. One of the most communal AI techniques used for handing out Big Data is Machine Learning, a self-adaptive algorithm that gets gradually better analysis and patterns with experience or with new added data.

If a digital payments company wished to notice the incidence of or potential for fraud in its system, it could employ machine learning tools for this purpose. The process algorithmic rule designed into a pc model can method all transactions happening on the digital platform, find patterns in the data set and point out any anomaly detected by the pattern. Deep learning, a subset of machine learning, utilizes a graded level of artificial neural networks to carry out the process of machine learning.

The artificial neural networks square measure designed just like the human brain, with neuron nodes connected together like a web. While ancient programs build analysis with information during a linear approach, the hierarchical function of deep learning systems enables machines to process data with a nonlinear approach.

A traditional approach to detective work fraud or concealment would possibly think about the number of group action that ensues, while a deep learning nonlinear technique would include time, geographic location, IP address, variety of distributor and the other feature that's seemingly to purpose to a fallacious activity. The first layer of the neural network processes an information input just like the quantity of the group action and passes it on to consecutive layer as output. The second layer processes the previous layer’s data by as well as extra data just like the user's science address and passes on its result.

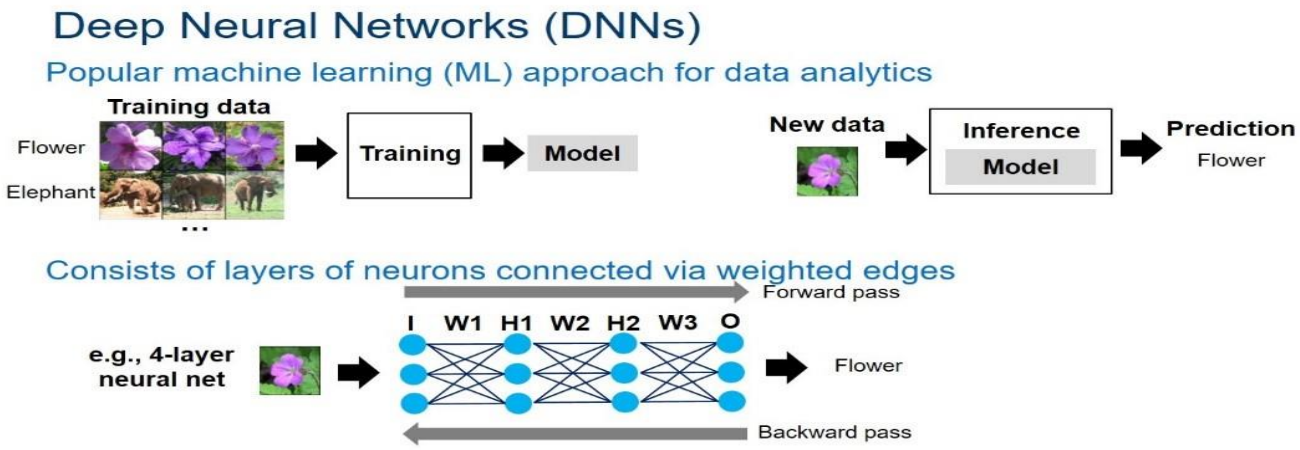


Fig 1.3 Deep Neural Networks

1.4 TENSORFLOW

The central unit of information in TensorFlow is that the tensor. A tensor consists of a group of primitive values formed into associate in nursing array of any variety of dimensions. In straight forward terms, a tensor is simply a multi-dimensional array. A rank of a tensor is that the dimension of the array. For example: a pair of two-dimensional array may be a rank two tensor, a three-dimensional array may be a rank three tensor. So, [1, 2, 3] is 1-D array, thus, a rank 1 tensor, [[1, 2, 3], [4, 5, 6]] is a 2-D array, thus, a rank 2 tensor. TensorFlow is an open-source software library.

TensorFlow was originally developed by analyzers and engineers functioning on the Google Brain Team at intervals Google's Machine Intelligence research organization for the needs of conducting machine learning and deep neural networks analysis, however the system is general enough to be applicable during a big variety of alternative domains as well. TensorFlow is essentially a package library for numerical computation exploitation information flow graphs where: nodes within the graph represent mathematical operations.

1.5 OPENCV

OpenCV (Open Source computer vision) could be a library of programming functions chiefly aimed toward period computer vision. Originally developed by Intel, it had

been later supported by Willow Garage then it sees the library is cross-platform and free to be used below the ASCII text file under the BSD license. The first alpha version of OpenCV was free to the general public at the IEEE Conference on Computer Vision and Pattern Recognition in 2000, and 5 betas were released between 2001 and 2005. The version 1.0 was released in 2006. A version 1.1 "pre-release" was released in 2008.

The second major unharness of the OpenCV was in Gregorian calendar month 2009. OpenCV version 2 includes major changes to the object oriented interface, aiming at easier, more type-safe patterns, new functions, and better implementations for existing ones in terms of performance (especially on multi-core systems). Official releases currently occur each six months associated in constant development is currently done by an freelance Russian team supported by industrial firms.

1.6 NUMPY

NumPy may be a library for the Python artificial language, adding support for big, multi-dimensional arrays and matrices, together with an out sized assortment of high-level mathematical functions to operate on these arrays. The relative of NumPy, Numeric, was originally created by Jim Hugunin with contributions from many alternative developers. In 2005, Travis Oliphant created NumPy by incorporating options for the competitive Num array into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

NumPy targets the CPython location implementation of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms written for this version of Python typically run abundant slower than compiled equivalents. NumPy addresses the gradualness downside partially by providing dimensional arrays and functions and operators that operate expeditiously on arrays, requiring rewriting some code, mostly inner loops using NumPy.

Using NumPy in Python provides practicality admire MATLAB since they're each taken, and they both allow the user to write fast programs as long as most operations work on arrays or matrices instead of scalars. In comparison, MATLAB vaunts a large number of additional toolboxes, notably Simulink, whereas NumPy is intrinsically integrated with Python, a more modern and complete programming language.

CHAPTER 2

LITERATURE REVIEW

In different fields there is a need to distinguish the focused on article and furthermore track them viably while taking care of impediments and other included complexities. Numerous scientists (Almeida and Guting 2004, Hsiao-Ping Tsai 2011 , Nicolas Papadakis and Aure lie Bugeau 2010) endeavored for different methodologies in item following. The idea of the strategies to a great extent relies upon the application area. A portion of the exploration works which made the development to proposed work in the field of item following are portrayed as pursues.

2.1 OBJECT DETECTION

Article location is an imperative, yet difficult vision undertaking. It is a basic part in numerous applications, for example, picture look, picture auto-comment, scene comprehension and item following. Moving article following of video picture groupings was a standout amongst the most vital subjects in PC vision. It has just been connected in numerous PC vision field, for example, keen video observation (Arun Hampapur 2005), man-made consciousness, military direction, security recognition and robot route, medicinal and organic application. As of late, various fruitful single-object following framework showed up, however within the sight of a few items, object identification wound up troublesome and when objects are completely or somewhat stopped, they are obtruded from human vision which further builds the issue of discovery.

Tie Liu (2010) gave another methodology for the discovery of articles by format coordinating from the huge database gathering. The methodology was appropriate for multi-scale differentiate foundations and it utilizes a shading spatial technique to distinguish objects. This methodology neglected to recognize numerous items in a given client situation and furthermore worthless when the articles were in non-direct movement. This issue of fizzled recognitions was adequately defeated in the proposed framework via preparing the framework to recognize the items through a viable framework learning system and the articles in non-direct movement are followed utilizing the proposed molecule gathering approach.

Scott McCloskey et al (2011), gave a bit based technique to identify questions even within the sight of fractional impediments. The haze piece strategy was utilized to analyze extensive incomplete impediments when closer view object was out of core interest. The key value of this strategy was that it given an accurate answer for fractional impediments through foundation mapping. Be that as it may, this was conceivable just through a progression of suppositions made on foundation power, for example, the static idea of the foundation, the brightening changes, the shade of the items, and so forth. Likewise it was discovered suitable just for self impediment. A superior answer for this issue was to use a functioning appearance model to recognize the articles moving in powerfully evolving foundation. And furthermore the separation detailing method dependent on transport topology is utilized to identify the nearness of incomplete and complete impediments with high exactness in proposed MLP based item following.

Aniruddha Kembhavi et al (2011), exhibited an article recognition strategy utilizing shading likelihood maps to catch the shading insights of vehicles and their environment. Fractional Least Squares (PLS) to extend the information onto a much lower dimensional subspace was the technique featured and utilized. The benefit of this technique was that PLS empowers in the determination of a little subset of highlight information. In any case, the identifier's execution debases with diminishing enlightenment and procurement edge. The proposed MLP based article following framework is made vigorous by ideal determination of extraordinary highlights and furthermore by actualizing the adaboost solid grouping strategy.

2.1.1 Background Subtraction

The foundation subtraction technique by Horprasert et al (1999), had the capacity to adapt to nearby enlightenment changes, for example, shadows and features, even globe light changes. In this technique, the foundation demonstrate was factually displayed on every pixel. A computational shading mode, incorporate brilliance mutilation and chromaticity contortion was utilized to recognize shading foundation from the customary foundation or moving forefront objects. The foundation and closer view subtraction strategy utilized the accompanying methodology. A pixel was displayed by a 4-tuple [Ei, si, ai, bi], where Ei-a vector with expected shading esteem, si - a vector with the standard deviation of shading esteem, ai - the variety of the brilliance contortion and bi was the variety of the chromaticity bending of the ith pixel. In the

subsequent stage, the contrast between the foundation picture and the present picture was assessed. Every pixel was at long last characterized into four classifications: unique foundation, shaded foundation or shadow, featured foundation and moving frontal area object.

Liyuan Li et al (2003), contributed a technique for recognizing frontal area questions in non stationary complex conditions containing moving foundation objects. A Bayes choice principle was utilized for characterization of foundation and forefront changes dependent on between edge shading co-event insights. A way to deal with store and quick recover shading co-event insights was additionally settled. In this technique, forefront objects were recognized in two stages. To begin with, both closer view and foundation changes are separated utilizing foundation subtraction and fleeting differencing. The successive foundation changes were then perceived utilizing the Bayes choice rule dependent on the scholarly shading co-event insights. Both present moment and long haul procedures to become familiar with the regular foundation changes were utilized.

A calculation concentrated on getting stationary forefront districts as said by Álvaro Bayona et al (2010), which was valuable for applications like the recognition of surrendered/stolen articles and left vehicles. This calculation fundamentally utilized two stages. Right off the bat, a sub-examining plan dependent on foundation subtraction methods was executed to acquire the stationary frontal area districts. This identifies closer view changes at various time moments in a similar pixel areas. This was finished by utilizing a Gaussian appropriation work. Also, a few changes were presented on this base calculation, for example, sift holding the beforehand the past registered subtraction. The fundamental motivation behind this calculation was lessening the measure of stationary frontal area distinguished.

2.1.2 Template Matching

Format Matching is a system for discovering little pieces of a picture which coordinate a layout picture. It slides the layout from the upper left to the base right of the picture and contrast for the best match and the format. The format measurement ought to be equivalent or littler than the reference picture. It perceives a fragment with the most noteworthy relationship as an objective. Given a picture S and a picture T, where the component of S was both bigger than T, yield whether S contains a subset picture I where I and T are reasonably comparable in example and if such I exists, yield the area of I in S as in Hager and Bellhumear (1998).

Schweitzer et al (2011), inferred a calculation which utilized both upper and lower bound to recognize 'k' best matches. Euclidean separation and Walsh change portions are utilized to compute coordinate measure. The positive things included utilization of need line improved nature of choice as to which bound-improved and when great matches exist inalienable expense was overwhelming and it improved execution. Be that as it may, there were limitations like nonappearance of good matches that lead to line cost and the number-crunching activity cost was higher. The proposed strategies dint use line in this manner staying away from the line cost rather utilized layout coordinating.

Visual following techniques can be generally sorted in two different ways to be specific, highlight based and area based strategy as proposed by Ken Ito and Shigeyuki Sakane (2001). Highlight based methodology appraises the 3D posture of an objective article to fit to the picture includes the edges, given a 3D geometrical model of an item. This technique requires much computational expense. District based can be ordered into two classifications to be specific, parametric strategy and view based technique. The parametric technique expect a parametric model of the pictures in the objective picture and computes ideal fitting of the model to pixel information in an area. View based strategy was utilized to locate the best match of a district in a pursuit zone given the reference layout. This has the preferred standpoint that it doesn't require much computational multifaceted nature as in the component based methodology.

2.2 OBJECT TRACKING

An example for following moving items proficiently was overviewed by Yilmaz et al (2006), through mining bunch development. The work centers around different gathering following dependent on nearby relationship and moving example. In this methodology, comparable article moving example could be productively gathered however unique moving item couldn't be assembled and therefore following bombed in such situations prompting vitality squander. A suitable arrangement received here is to utilize singular trackers for the items in the video under examination.

An example for following moving items productively displayed by Hsiao-Ping Tsai (2011), through mining bunch development. This work is concentrated on various gathering following dependent on neighborhood relationship and moving example. In this methodology, comparable article moving example could be effectively assembled yet extraordinary moving

item couldn't be gathered and along these lines following bombed in such situations prompting vitality squander. In the proposed Rk-d direction chart, moving example lead in the basic district distinguishing proof which structure gatherings to maintain a strategic distance from crash. The assembled items and diversely moving article followed effectively.

A framework to recognize and follow various people from a solitary uncalibrated camera was expressed by Michael D. Breitenstein et al (2011). The framework utilized a bootstrap channel to follow different people and target explicit classifiers to choose the items and partner them to targets. The benefits included powerful treatment of false positive discoveries however at the expense of a high computational unpredictability. The genuine downside was that no foundation displaying was finished. In the proposed framework, molecule gathering limited mapping unpredictability and dynamic scene displaying done as such as to keep away from additional calculation.

2.2.1 Filter Based Tracking

Jin Wang (2010), displayed a model that improved the current Kalman Filter to follow questions even within the sight of impediments. This work was a mix of appearance and movement data to follow objects. The upsides of the technique were that it was fruitful in finding objects precisely notwithstanding when they were incompletely or completely blocked and had the capacity to effectively recognize disparately shaded items. The hindrance of the technique was its untrustworthiness and less efficienct alongside wrong correspondence when diverse articles have comparative appearance. A powerful answer for defeat these issues are given in the proposed framework by choosing movement parameters alongside highlight extraction and defining the separation based transport topology system.

Ibrahima and Patricio (2011), utilized EKF alone to gauge the situation of deformable articles. In light of bend speeds forecasts were finished. This strategy simply utilized a solitary channel for the development of articles. The proposed programmed KFB utilizes a bank of channels which yield preferable outcomes over utilizing only a solitary channel for a development. St-Pierre (2004), had completed an examination between the UKF and the EKF for the position estimation module of an incorporated route data framework was performed. It concentrated on the utilization of EKF for situating the route data framework and UKF when forecast and refresh capacities are exceptionally non straight estimation ends up complex.

Despite the fact that this strategy improved the exactness of estimation, it renders the framework futile when there is no GPS arrangement. And furthermore the computational time of the combined EKF and UKF was turned out to be 3% more noteworthy than the EKF alone. An adept answer for this issue is given in the proposed framework by molecule gathering based methodology for item following to limit the calculation.

2.2.2 Agent Based Tracking

Lately specially appointed parallel information handling has risen to be one of the executioner applications for foundation as-an administration mists. Specific undertakings of a preparing employment can be doled out to various kinds of virtual machines. Goldszmidt and Yemini (1998), characterized an abnormal state language named DEAL(DElegated Agent Language). The primary point was to limit utilization of the assets at the system level and handling time at the chief dimension.

Daniel Warneke et al (2011), talked about the chances and difficulties in parallel information preparing in the cloud and present Nephele, the primary information handling structure to unequivocally abuse the dynamic asset assignment. Nephele was cost-productive as it dispensed and de-allotted virtual machines as per the present place of employment execution stage. It likewise gave the criticism information. Notwithstanding, the structure does not permit correspondence between the different errand administrators. Also, the client must choose the activity diagram and the code to be executed. Proposed work bolsters association and correspondence between the specialists utilizing FIPA ACL for crash shirking.

Kohsia Huang et al (2001), planned a system for organizing disseminated camcorder to follow human development. Anyway this system did not give the coordination technique to the operators. Proposed work makes a planned bunch of specialists which communicate utilizing FIPA ACL. The specialists trade messages to arrange crafted by article relocation.

Hideaki Takeda et al (1997), thought of an operator based structure where clients speak with a robot by means of a camcorder situated outside of the robot. The system gave an omnipresent access technique that investigates the operator as a way to build up correspondence among clients and robots by means of circulated sensors. In spite of the fact that this system is intended for Human-Robot-Interface utilizing specialist innovation, it didn't take care of the

disconnected issue. The disconnected issue is brought about by the temporary nearness of a robot. Proposed work utilizes a spatio-transient database to recover data about the article. This evacuates the requirement for direct physical association between the item to be controlled and the operator.

Valter Silva et al (2011), examined a system to furnish customers of area based administrations with a more extravagant end-client experience by methods for administration creation, personalization, gadget adjustment and congruity of administration. This methodology depends on a multi-specialist framework made out of intermediary operators that go about as middle people and suppliers of personalization meta-administrations, gadget adjustment and coherence of administration for customers of prior area based administrations. These intermediary operators, which have Web administrations interfaces to guarantee an abnormal state of interoperability, perform administration organization and take in thought the inclinations of the clients, the impediments of the client gadgets, making the use of various sorts of gadgets consistent for the end-client.

Daniela Bordencea et al (2011), structured a framework that comprise of implanting ultralow control, Wi-Fi transmission abilities in an extremely little bundle. This framework kept running on batteries having a trademark lifetime of several years and offered a stage for sensor estimations. The checking framework can utilize any current foundation, with huge lessening of execution costs.

2.2.2.1 Data retrieval using agents

Arun Kumar Yadav and Ajay Agarwal (2010), have talked about the different simultaneousness control calculations for distinguishing and settling the contentions. The calculations - Distributed 2PL, Wound-Wait, Basic Timestamp orders and a Distributed hopeful calculation has been concentrated utilizing a nitty gritty model of a Distributed DBMS.

Sergio Ilarri et al (2006), displayed a framework that bolsters dispersed preparing of consistent area subordinate questions in portable conditions without over-burdening the client's remote gadgets. The proposed framework additionally bolsters ceaseless inquiries, moving questions and area mindful questions. An example area subordinate question is appeared in

Figure 2.2.1. The inquiry requests police units that are inside 0.56 miles around car38 (a stolen vehicle) and the squad cars inside 0.42 miles around policeCar5.

```
SELECT   policeUnit.id, policeCar.id
  FROM   policeUnit, policeCar
 WHERE   inside(0.56 miles, 'car38', policeUnit) AND inside(0.42 miles, 'policeCar5', policeCar)
         AND policeCar.id<>'policeCar5'
```

Fig 2.2.1. Query for area subordinate question

2.2.2.2 Autonomous agents

An autonomous agent is an intelligent agent operating on an owner's behalf but without any interference of that ownership entity.Ruth Aguilar-Ponce et al (2005), had studied about Autonomous Decentralized Systems (ADS) and utilized it to form a cluster of networked visual sensors. The ADS detects and tracks the objects. The Autonomous Observer Cells (AOC) send all the tracking information to the Autonomous Analysis Cell (AAC) which further analyzes the object. Based on the analysis the AAC decides what action to pursue further. The architecture diagram is shown in Figure 2.2.2.

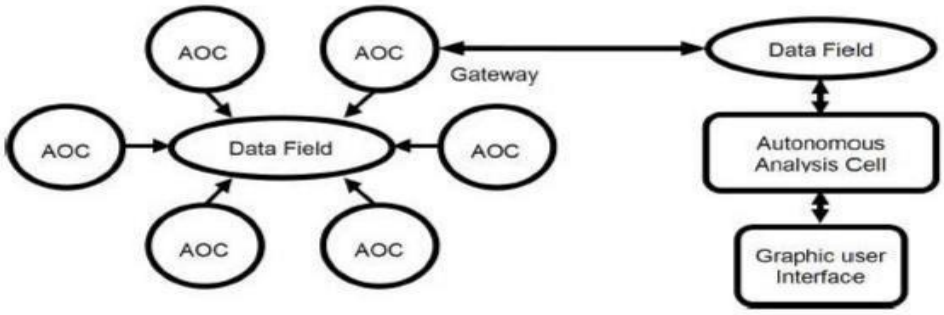


Fig 2.2.2. Architecture Diagram of Autonomous Agents

Notwithstanding, this work has decentralized just the data gathering errand. The primary strategy to be taken should even now be chosen by the AAC. This diminishes the dimension of

decentralization. The work does not consider the impact shirking of the articles. Proposed work decentralizes the work by part the earth into spaces. Every space has a MA and a CA dealing with the crash shirking.

Markus Dietl et al (2001), exhibited strategies for following items from boisterous and questionable information. The various specialists assemble information about their condition and trade them with a module for multi-operator sensor combination. In this work, the data gets adjusted by the operator's perspective on the world. Proposed work does not adjust the information and utilizations the first information to anticipate and keep the crashes in a space.

2.2.3 Tree Based Tracking

The expansion in PC applications has consequently prompted the expansion in database application and multidimensional database itself. At the point when the database turns out to be expansive, seek turns into a repetitive work. The questioning appears by then of time and it the diverse sorts of ordering and questioning were given by Volker Gaede (1998).

Q+Rtree (Yuni Xia and Sunil Prabhakar 2004) depended on the perception that I) most moving items are in semi static state more often than not and ii) the moving examples of articles are firmly identified with the geology of the space. The Q+R tree is a half and half tree structure which comprises of both a R-tree and a Quadtree. The R tree lists semi static items. The uadtree files quick moving articles. Utilizing quad trees for ordering dynamic articles lessens visit refreshes.

R k-d direction trie (Priyadasini et al 2011) proposed as a two layered structure with R k-d tree on the upper layer and direction trie on the lower layer. It likewise bolsters full, incomplete and repetitive example coordinating for inquiries. Be that as it may, the inconvenience being high memory utilization and questioning expense as direction trie is kept up for each article in the R k-d tree. Improved work utilizes a typical direction chart speaking to the directions everything being equal. A wide scope of diagram explicit inquiries are upheld.

Dmitri V. Kalashnikov et al (2002), investigated the execution of ordering approaches R-tree, R*-tree, matrix tree and quad tree. R-tree and R*-tree file structures are intended to be circle based structures. R-tree is an unadulterated information parceling list structure. An ideal access is accomplished by picking the hub size to be a various of circle space. R*-tree is a

variation of R-tree where the meaning of cover of square shapes and part calculation is extraordinary.

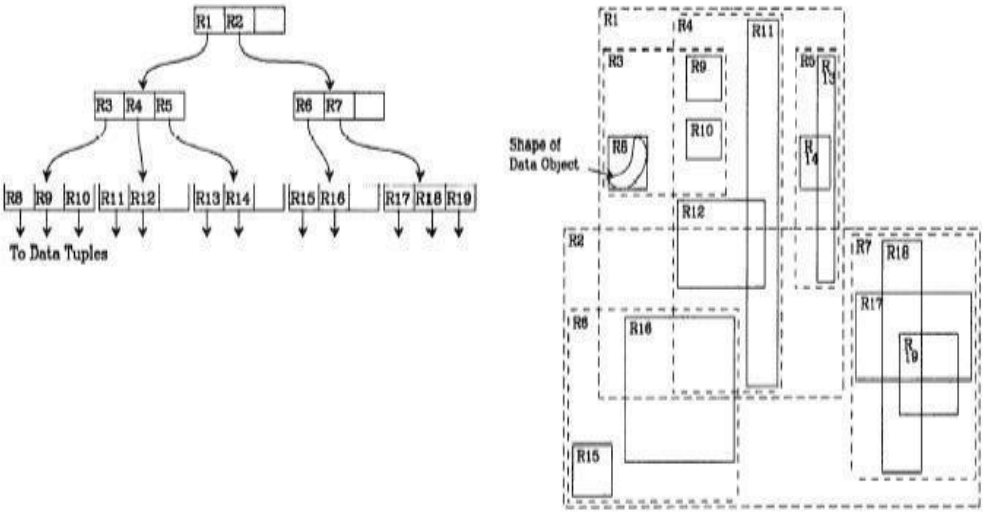


Fig 2.2.3 R-tree Example

The network record and the quad-tree are all around firmly related. They are both space parceling ordering procedures and handle the overfull locale by split activity. k-dtree is an unadulterated space parceling list structure. k-dtree has fan out free of dimensionality and no cover between subspaces.

Benetis et al (2002), thought of Time Parametrized R tree (TPR) for quering Reverse Nearest Neighbor (RNN) utilizing speed vectors. The primary favorable position was that it underpins relentless and ceaseless inquiries. The real downside was it restricts development outside course structure. The proposed technique does not lay any principles dependent on the development.

2.2.4 Quantum Based Tracking

Glenn Beach et al (2003), had a methodology that utilizes a quantum seek calculation in Image Processing. They had utilized Quantum calculations for looking through a specific incentive in a n x n cluster of qualities. In a N x N cluster, on the off chance that a n x n exhibit is looked, at that point the time multifaceted nature is O( ). In any case, Quantum look calculation

requires just O(Nn2) time multifaceted nature. They had demonstrated that the time multifaceted nature of quantum calculation is far less when contrasted with other regular pursuit calculations.

Mitja Perus et al (2004), has a potential elective execution into a quantum-wave mechanism for a superior picture acknowledgment technique. Quantum-net's abilities of network, parallelism, stockpiling, associativity, speed and scaling down are huge. The Hopfield display with genuine esteemed (along these lines not really parallel) exercises of units/neurons, having straight (not sigmoid or signum) enactment work, can be changed into a quantum-holographic technique where the Hebbian memory stockpiling is supplanted by numerous self-impedances of quantum plane waves. This interpretation prevailing by the least complex variable trade in the Hopfield's genuine esteemed factors with the complex-esteemed factors changing as sinusoids (waves). In this way, all contribution to-yield changes are protected. In this manner quantum-wave picture acknowledgment works proportionately to Hopfield's one.

2.2.5 Linear Programming Approach

Hao Jiang et al (2010), had thought of a methodology that utilizes direct programming unwinding plan for the class of various article following issues. This plan models object following as a multi-way looking issue. It demonstrates track collaboration, for example, object spatial format consistency and enhances numerous item following all the while. It utilizes Network display for following numerous items. They had demonstrated that exactness rate is 94% per outline.

2.3 COLLISION AVOIDANCE

At the point when numerous articles are followed, one of the real objectives is to foresee and keep the impacts.

Wei You et al (2008), examined a strategy for numerous article following in shrewd conditions where the items may have similar highlights. The work consolidates facial highlights and the entire body histogram to particularly distinguish and name an item. The framework does not think about the crashes between the items. Proposed work has a structure for impact shirking dependent on bunched specialists. Both between area and intra-space crashes have been dealt with.

Mei Han et al (2004), built up a different article following calculation that looks for the ideal state grouping which amplifies the joint state-perception likelihood. It evaluated the state grouping or "direction" rather than the present state. The calculation is equipped for following various items whose number is obscure and differs amid following. They utilize a Hidden Markov Model as the probabilistic model to augment the joint likelihood between the state grouping and the perception arrangement

Zhengya Xu et al (2009), proposed an ongoing item following framework dependent on multi-see cameras is proposed in this paper. So as to follow a moving item, a functioning camera mounted on a skillet/till stage constrained by static cameras is utilized in this framework. A foundation displaying based moving article discovery and investigation method for the multi cameras based continuous item following framework have been proposed for acquiring the important flag for the control.

Ulrich (2003), demonstrated that conduct of portable robot is confused. This was quantitatively demonstrated utilizing the disparity property of the directions of the robots. In proposed work, disparity and union property of the items has been utilized to distinguish the articles adding to the basic district. The basic area is where there is a high likelihood of impact. More the level of assembly of different directions at a moment, more the likelihood of collision.The proposed work goes for keeping away from impact better by thinking about gathering of nature.

2.3.1 Visual Angle

Kahlouche Souhila et al (2007), built up a calculation for visual hindrance evasion of independent portable robot. They have utilized optical stream data removed from the picture succession for this reason. The opportunity to crash is discovered utilizing the optical stream data. The significant confinement of this technique is that it is touchy to commotion. The proposed work gives another geometric technique to ascertain the TTC which is less touchy to clamor. Steffen Gormer et al (2009), created a framework with Forward Collision Warning, Lane Departure Warning and Electronic Stability Control. They have additionally proposed a novel vehicle recognition approach and a powerful TTC estimation framework. The primary weakness in the TTC count system given in this work is that the information of the genuine separation of the main vehicle and relative speed of the two vehicles is required. The proposed work does not

require the real separation or speed to be known. Rather it computes the TTC utilizing the rate of edge sub tense of a moving toward article.

2.4 MANAGING DYNAMIC UPDATES

With the quick advances in situating frameworks, for example, worldwide situating frameworks and portable figuring advances, overseeing modern data about the areas of monstrous moving items has turned into a basic territory of research. Kwon et al 2002, created sluggish updates performed in a base up way by embracing an optional list on the R-tree.

All the more as of late, Biveinis et al (2007), anticipated RR-tree which abuses in-memory task cushion as another R-tree and backings mass addition calculations. MoonBae Song et al (2009), displayed a R-tree-based list structure (called Rsb-tree, R-tree with semi mass stacking) for proficiently overseeing regular updates from gigantic moving articles. The idea of semi mass stacking is abusing a little in-memory cradle to concede, support and gathering the approaching updates and mass addition these updates at the same time. Proposed work stretches out the semi bulk stacking procedure to R k-d direction chart for amplifying the refresh throughput of moving articles. Proposed flush calculation was stretched out so as to improve refresh and inquiry execution together.

Lee et al (2003), built up an in-memory rundown structure to improve the two updates and questions. In-memory support contains an article vault, histogram and decimation list. Article Registry (OR) is a lot of item tuples (objID; p; cid; tstamp;Nhit) hashed dependent on item id, where p is area of item; cid is an identifier of a histogram cell, tstamp is a period stamp doled out by GlobalClock and Nhit is the quantity of updates performed inside OR (called OR hit). The limit of OR (ORSize) is characterized as a small amount of the all out number of moving articles, N. The histogram is two dimensional hash structure that partitions the information space into cells (g*g). Every cell contains the esteem hist and item ID of OR sections whose area are totally encased by it. Demolition list contains data about old sections to be erased.

Jing Zhou et al (2007), examined a separation based area refresh conspire by finding an ideal separation edge. To decide an ideal edge adaptively, this work proposes two improvement

calculations, to be specific, approximate calculation and dynamic calculation. Approximate calculation surmises the present framework condition dependent on which it straightforwardly decides the most plausible ideal esteem though the dynamic calculation begins with a specific edge esteem and modifies it slowly towards the ideal point.

2.4.1 Query Processing

An expanding number of utilizations require handling of moving Points of Interest (POI) in light of the fundamental system. They require preview questions instead of persistent checking. Spatial information preparing is a functioning examination field. K. Mouratidis et al 2005, centered handling POIs and went for constantly observing a lot of moving closest neighbors. In any case, two fundamental difficulties when supporting POI portability on a system are (an) effectively overseeing item area updates and (b) give quick system remove calculations.

To address these issues Haojun Wang et al (2010), structured a novel framework to process area put together questions with respect to moving items. The objective is to bi-directionally outline two structures on-plate R*-tree and the in-memory matrix file and recover an insignificant arrangement of information for preparing the inquiries. It helps in pruning the pursuit zone. In view of the arrangement of network cells covering with a given edge, calculations execute go just as k closest neighbor questions were introduced. Be that as it may, in this methodology nonstop inquiries are not bolstered and consolidating dynamic system refreshes are basic. This is overwhelmed by executing preview based inquiries in unique conditions and expanding the usefulness of R k-d trees to help consistent questions. Constant client needs are likewise fulfilled through pertinence input condition dependent on the client feeling and question results. The question preparing will be proceeded with increasingly pertinent inquiry conditions from the past outcomes criticism from the clients, in this manner accomplishing better question execution.

Prabhakar et al (2002), had thought about speed as limitation and concocted question based ordering. The fundamental favorable position was that it dealt with arrange inquiries and the updates productively. A requirement in articles conduct was the negative in this strategy. The proposed Rk-d technique does not put any requirements for the article development.

2.4.2 Spatial Database

In different fields there is a need to oversee geometric, geographic, or spatial information, which implies information identified with space. A spatial database is a database that is streamlined to store and question these information identified with articles in space, including 2D and 3D focuses.

Austin Parker et al (2009), talked about the idea of "idealistic" choice, where there is a productive ordering system to execute hopeful determination inquiries over SPOT (Spatial Probabilistic Temporal database) databases. Besides, spatial subtleties are contended to be listed adequately with a careful question determination utilizing information structures like BSP tree and SPOT trees. In any case, it is seen that while tending to questions, for example, choice inquiries, by recovering information from circulated databases, a test stays as there are no effective joins or total tasks on the structures considered by them. Proposed work bolsters all the conceivable spatio-worldly questions with join activities and propelled sort of direction design coordinating inquiries.

Kolahdouzan (2004), offered two system tending to C-KNN (Continous k-Nearest Neighbor) inquiries in SNDB (Spatial Network DataBase) in particular convergence examination and upperbound calculation. The primary favorable position is proficiently finding the area of articles. The drawback being a reduction in exhibition when POI (Point Of Interest) were thickly conveyed in systems. The proposed strategies utilize spatial and fleeting database thus even the thickly disseminated system can find the articles.

CHAPTER 3
SYSTEM ANALYSIS

3.1 PROBLEM DEFINITION

As object recognition is a key output of machine learning and deep learning, the existing systems does not give the accuracy of the recognized object and it also needs a web application to execute. This project is used to give the accuracy of the objects with pre-trained models and it can also detect the accuracy of real time running objects with the help of an Android Application.

3.2 EXISTING SYSTEM

The web application is conveyed to a VM occurrence running on Compute Engine. At the point when the customer transfers a picture to the application, the application runs the surmising work locally. The pre-prepared model returns the names of distinguished items, and the picture directions of the comparing objects. Utilizing these qualities, the application produces new pictures populated with square shapes around the recognized articles. Separate pictures are created for each item class, enabling the customer to segregate between chosen objects. This can send the pre-prepared model on Google Cloud Machine Learning Engine to give an API administration to deduction. On the off chance that is done, the web application sends an API ask for to distinguish protests in the transferred picture, rather than running the derivation work locally. TensorFlow enables to pick which stage to execute deduction occupations on relying upon our business needs. This adaptability demonstrates the upside of Google Cloud Platform and TensorFlow as an open stage for AI. It can be utilized with five pre-prepared models with the Object Detection API. Also are prepared with the COCO dataset and are equipped for distinguishing general items in 80 classes.

3.2.1 Drawbacks

- As they are utilizing the web application for the acknowledgment of pictures, it will be simple for the outsider individuals to duplicate the pictures.

- It can acknowledge just the pictures of certain size. It can't be extended with careful quality any longer.
- If the picture is compacted the measure of the pixels are additionally diminished which influences the nature of the picture.
- It is preposterous to expect to recognize questions in low lighting.
- It is beyond the realm of imagination to expect to catch the ongoing moving pictures.

3.3 PROPOSED SYSTEM

Building up an android application for versatile stages called the Object Recognition System utilizing TensorFlow, which pictures the article from a picture or a video document through pre prepared item show.

This proposed system not just identifies the articles from the picture, additionally it names the recognized picture with its basic name and determined exactness, this framework likewise gives mean ongoing video following where the client can follow an item which is moving in the video

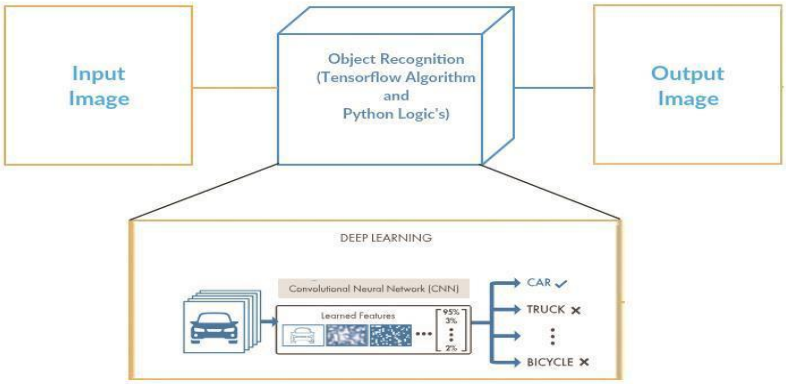


Fig 3.3.1 System Design for Object detection model

3.3.1 Advantages

- It will give the right picture thickness and difference.
- It serves to effortlessly store and recover the pictures in the PC.
- It is additionally equipped for perceiving objects even in low lighting.

- The pictures can be perceived in any ideal structures like highly contrasting, negative picture and so on.

3.4 SYSTEM REQUIREMENTS

3.4.1 Hardware Requirements

The hardware used for the development of the project is

MONITOR : 15 COLOR – 1280 x 800 minimum screen resolution

RAM : 4 GB & above

HARDDISK : 150 GB & above

PROCESSOR : Intel core i3 or above.

3.4.2 Software Requirements

The software used for the development of the project is

OPERATING SYSTEM : Microsoft Windows 7/8/10 (64 bit)

SOFTWARE TOOLS : Jupyter Notebook, Android Studio IDE

LANGUAGE : Anaconda, Keras, Tensorflow.js

CHAPTER 4
SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

The system architecture of Object Recognition using TensorFlow gives a detailed overview about the model of the system.

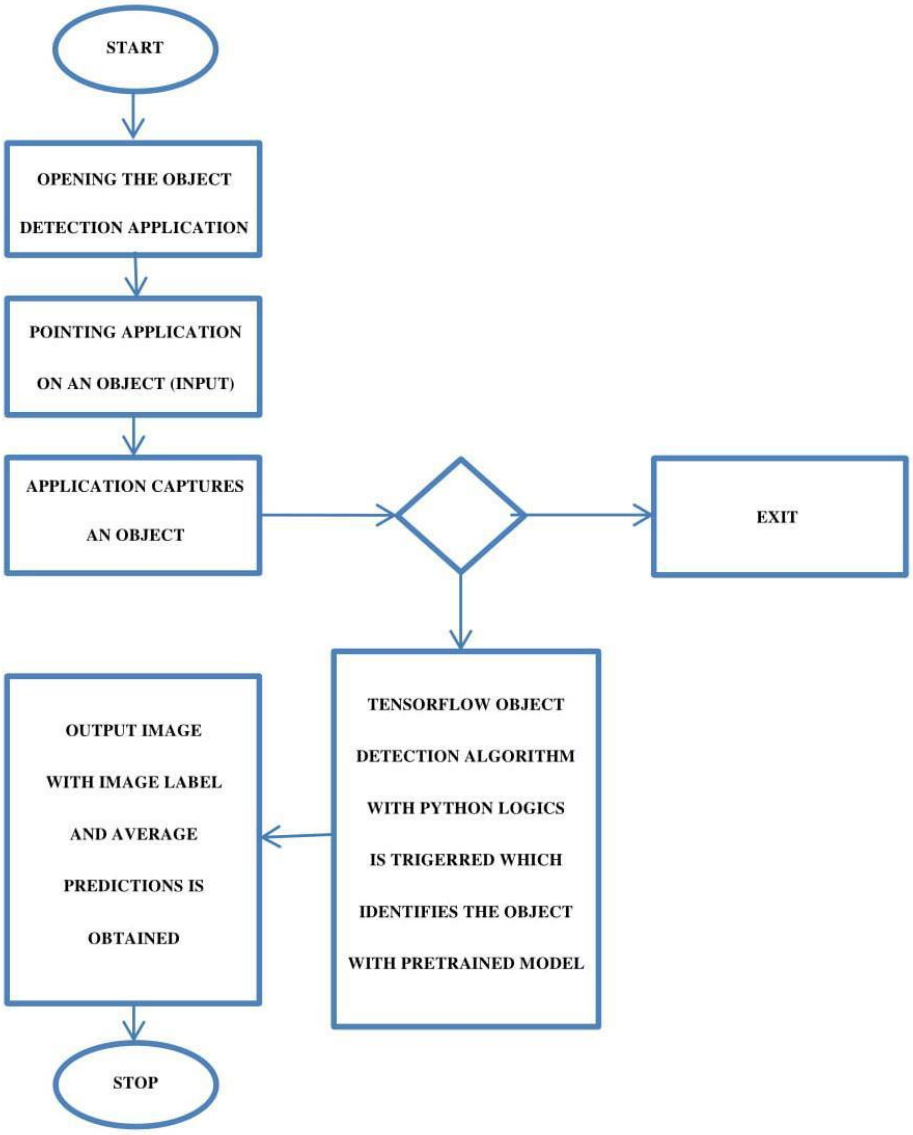


Fig 4.1 System Architecture

At first, open the application that focuses to the item to be identified. The item ought to be recognized through that specific application through which the information layers from the picture are ordered. By utilizing the TensorFlow item identification calculation, the information design from the info picture is contrasted and the pre-prepared models through Convolutional Neural Networks (CNN). On the off chance that the example of the info picture precisely coordinates with any one example of the concealed layer of a pre-prepared model, the yield layer is acquired that portrays the picture mark and its normal expectation. In the event that the catch picture does not coordinates any of the pictures of pre-prepared models the procedure will stop and exit. This framework gives the precision if the yield layer through which the picture exactness is inferred, this has the capacity of creating yields even in low lights and furthermore can be utilized to follow questions progressively movement recordings.

4.2 SYSTEM MODULES

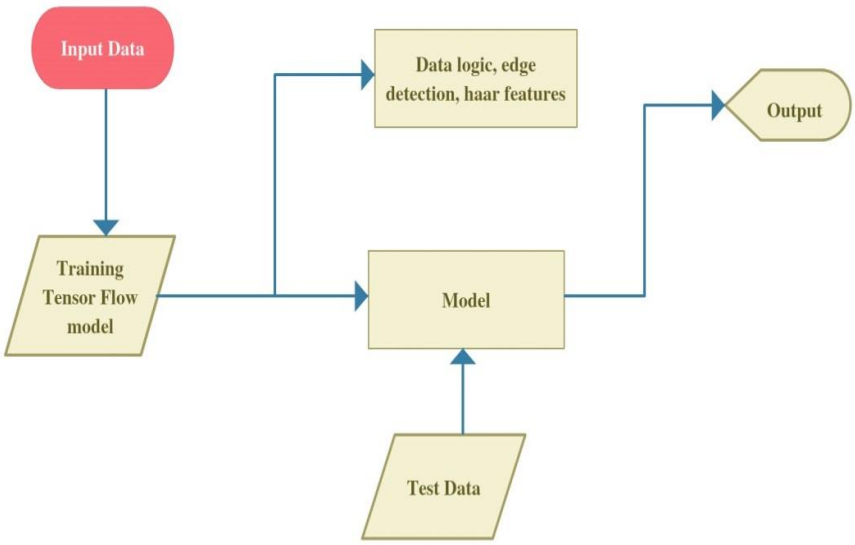


Fig 4.2.1 Test and Training data flow

At first, open the application that points to the object to be detected. The application will capture the object which has to be detected for various purpose. The captured image is considered as the input data. With the help of TensorFlow the captured image is compared with the pre-trained models of COCO dataset by Convolutional Neural

Networks(CNN). The comparison of input image with pre-trained models are made by using various algorithms and logics like edge detection, Haar detection feature etc,. If the captured image matches anyone of the pre-trained patterns of COCO dataset, the output will describe the label for the image along with its accuracy. If the captured image doesn't matches with any of the pre-trained models the process will exit. With the help of this system one can capture the images even in low dense lights and one can also track the real time motion videos.

4.2.1 Training the module using Tensorflow for still images

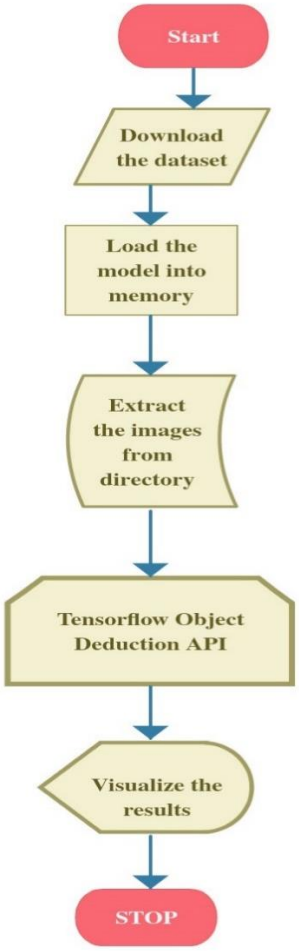


Fig 4.2.2 Still Image Detection

The proposed system can also be used to detect the captured images and give its accuracy. Steps for the detection of object and its accuracy for already captured images are similar to the steps used for the detection of already captured videos. As usual first step is to download the dataset followed by loading the module into memory. Next step is to extract the stored images from the directory and with the application of TensorFlow object detection API the output will be visualized.

4.2.2 Training the module using Tensorflow for recorded video

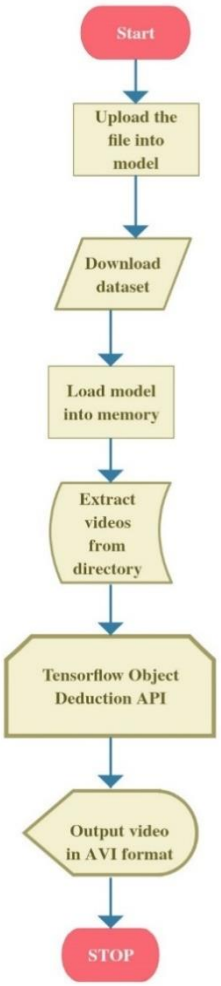


Fig 4.2.3 Recorded Video Detection

The proposed system not only used to detect the capturing images but with the proposed system it can also detect and label the already captured videos. To detect the already captured

images, the file of the image should be uploaded in the corresponding system. After that dataset should be downloaded from the respective site through which the object can be compared. All the modules of the dataset need some memory. Hence modules are loaded into the memory. If the user need to detect the object from already capture video, it should be extracted from the respective directory. The main objective of the proposed system is achieved only after the application of TensorFlow object detection API. Finally the output for the video will be in the AVI format.

4.2.4 Training the module using Tensorflow for live video detection

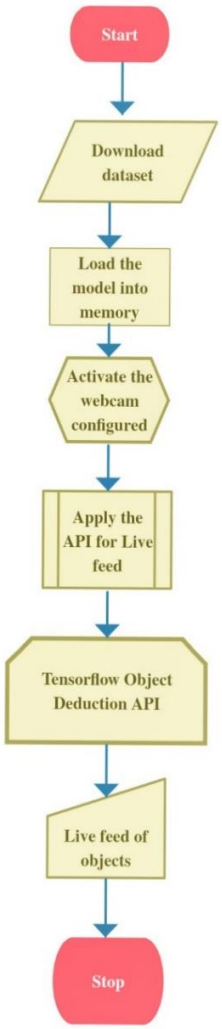


Fig 4.2.4 Live Video Detection

For training the modules in the TensorFlow platform several procedures have to be followed. At first, dataset has to be downloaded from the respective site. In this proposed system COCO is used. After downloading the corresponding dataset it is mandatory to load the modules into the memory from where it can be fetch the modules in future. Once all these prerequisites are ready the next most important thing is to activate the configured webcam which will be used to capture the images Application Program interface should be applied for feeding the dataset from the memory. The process will be incomplete until the TensorFlow object detection API is applied. Hence the API for TensorFlow object detection is applied. After the API of TensorFlow object detection is applied the system is ready to feed the object which has to be detected.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 STATIC IMAGE DETECTION

In the first module of the proposed system, instance images will be captured and compared with the pre-trained models using TensorFlow. First step is to download the dataset followed by loading the module into memory. Next step is to extract the stored images from the directory and with the application of TensorFlow object detection API the output will be visualized. Here Haar object detection algorithm is used for the calculations of accuracy.

5.1.1 Haar Detection Cascade

Haar Detection Cascade is to eliminate negative examples with very little processing. A chain of classifiers is computed to every sub-region in the image. If the alternate part of the large region does not pass all of the classifiers than that image is liberated and further computation is performed. If a sub-region passes the first stage which requires little computation, it is then passed onto the next stage where a little more computation is needed. If the image goes by this classifier it is then passed to another part of a large region. In order for face detection to occur, an image subregion must pass all of these classifiers. One can train a classifier to improve its accuracy.

Haar-like features are an over the complete set of two dimensional (2D) Haar functions, which can be used to encode local appearance of objects. They consist of more than two rectangular regions confined in a template. The feature value f of a Haar-like feature which has k rectangles is obtained as in the following equation:

$$f = \sum_{i=1}^{k} w^{(i)}.\mu^{(i)} \qquad (1)$$

where μ(i) is the mean intensity of the pixels in an unknown image enclosed by the ith rectangle. The quantity μ refers as the rectangle mean. In the given equation (1), μ(i) is the

heaviness assigned to the ith rectangle. Traditionally, the weights assigned to the rectangles of a Haar-like feature are set to default integer numbers

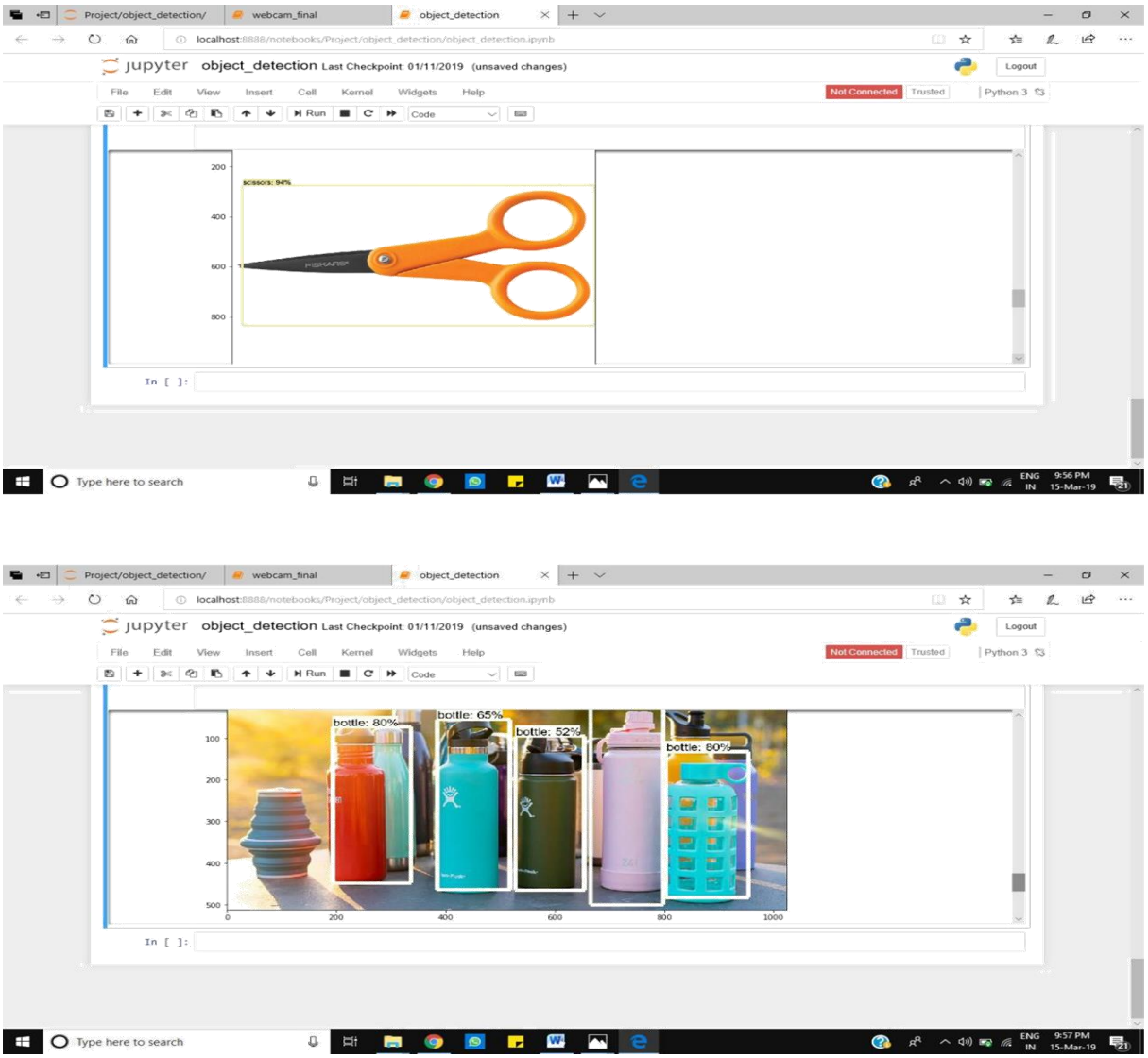


Fig 5.1.1 Static Image Detection

5.2 RECORDED VIDEO RECOGNITION

The first module of the proposed system is to train the models to detect and compare the object which was already captured with the pre-trained model using TensorFlow. For training the models, dataset has to be downloaded from the respective site. In the proposed system COCO dataset is used. After downloading the corresponding dataset it is mandatory to load the modules into the memory from where can fetch the modules in future. Once all these prerequisites are ready the next most important thing is to activate the configured webcam which will be used to capture the images Application Program interface should be applied for feeding the dataset from the memory. The process will be incomplete until the TensorFlow object detection API is applied. Hence the API for TensorFlow object detection is applied. After the API of TensorFlow object detection is applied the system is ready to feed the object which has to be detected. By using several algorithms like “Haar object detection algorithm”, the accuracy of the image will be calculated.

Fig. 5.2.1 Recorded video Detection

5.3 LIVE VIDEO IDENTIFICATION

The next module of the proposed system is to train the model for the instance videos. Here, to detect the live videos and images, the file of the image/video should be uploaded in the corresponding system. After that dataset should be downloaded from the respective site through which the object can be compared. All the modules of the dataset need some memory. Hence modules are loaded into the memory. If the user need to detect the object from already capture video, it should be extracted from the respective directory. The main objective of the proposed system is achieved only after the application of TensorFlow object detection API. Finally the output for the video will be in the AVI format.

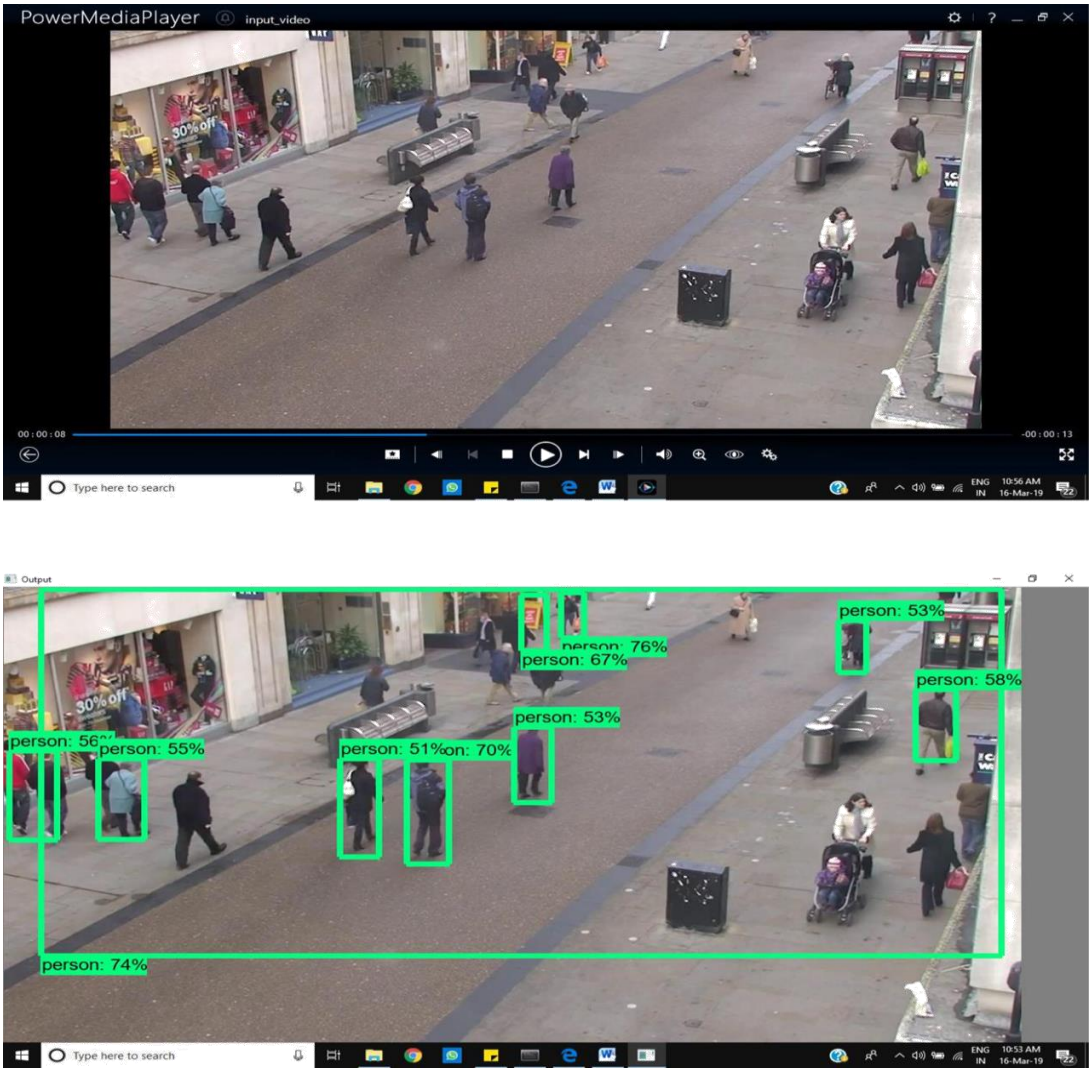


Fig 5.3.1 Live video Detection

5.4 DEVELOPING AN APPLICATION

The final module of the proposed system is developing an Android application. For developing an android application, the environment setup is the mandatory thing. The first step is to make environmental setup for the android studio and then download the required SDK's which should support the TensorFlow. Code will be developed according to the requirements of the proposed system. Before executing the code one should confirm that TensorFlow and other necessary software are synced with the android studio. After that run the project to build the gradle. The application will be ready for the detection of object and give its accuracy.

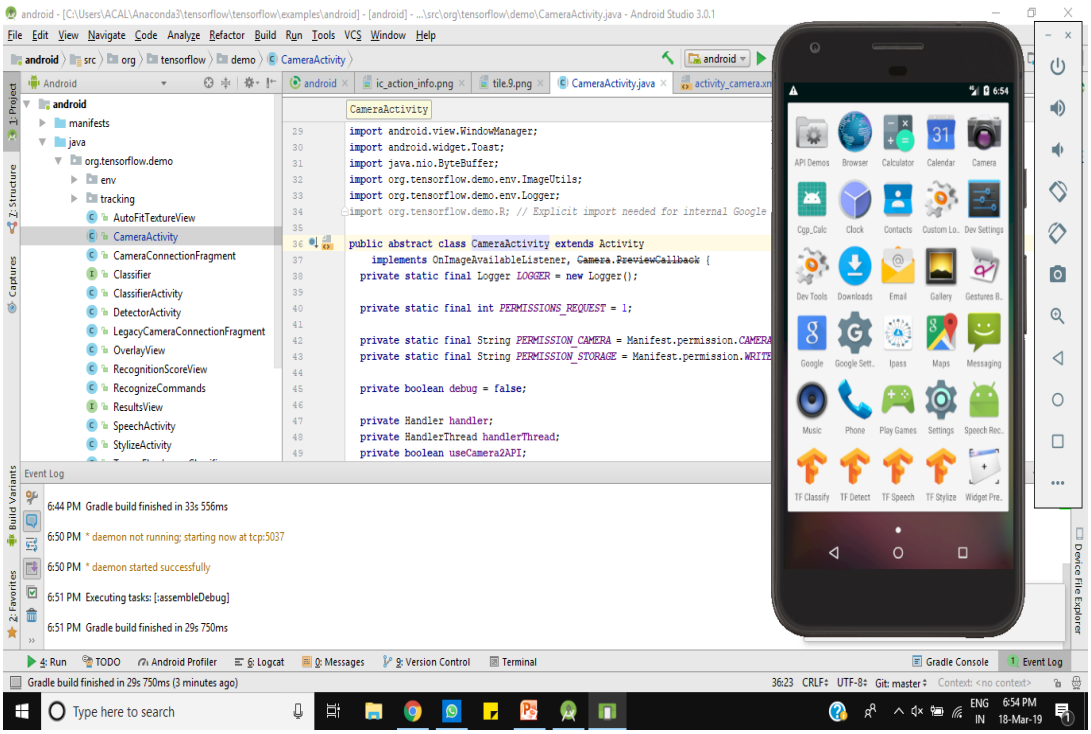


Fig 5.4.1 Mobile Application Development

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMEMT

The results show that the main aim is to detect objects in any real time scenes of surveillance, the face detection algorithm can be used to detect and follow people in case of surveillance for some other purpose. The object detection algorithm is used to detect and track the object instances in real world and also by labeling the class to which the object belong. This work is done in Python – OpenCV using the Tensorflow Machine Learning Library, an open source library to develop and deploy machine learning models and can also be performed with Matlab, Python combined with Tensorflow is highly preferred because this can be included in OpenCV programs and the execution time in Python is lesser, simple and is much more efficient. It is concluded that the work is reported in pointing its use in areas like face detection, object classification, surveillance and soon. The program can be incorporated in UAV's for better detection and navigation of the aircrafts in the near future. It could be fourthly upgraded to reduce the process time of the controller to a different methodology which can be tried and implemented. Hoping that the researches in the computer vision fields will get advancement and can be much more efficient with higher accuracy rate, the intuitive visualizations can help in much more advancements.

FUTURE ENHANCEMENTS

The proposed system is used to capture the live videos and images. In future the proposed system can be used for further more purposes. The rise of Special Purpose Integrated Circuits can help the researchers build much more efficient models than the ones present now, The Object Detection can also be used in domestic surveillance in case like people head count during festivals, identifying a specific person in such huge crowds, etc. For example, if a child lost his/her parent in a huge crowd, the image of the child will be matched with already captured CCTV photos. From the matched video, the image of the parents will be taken. Then with the help of that image the system will easily identify the particular object. With the help of the proposed system the register number of a high speed vehicle can also be taken when the system is enhanced in future. The detection can be more enhanced in near future.

PUBLICATIONS

[1]. Presented a paper titled “Object Detection with pretrained model using tensorflow” by Abishek H, Monisha D.S, Farhana S at 11th National Conference on Networks, Intelligence and Computing Systems – NCNICS 19 organized by SNS College of Technology - Coimbatore held on 07 March 2019.

[2]. Published a paper titled “Object Detection with pretrained model using tensorflow” in IJSER – International Journal of Scientific and Engineering Research in Volume 10, Issue 3, March 2019 Edition by Abishek H, Monisha D.S, Farhana S.

NCNICS 2019

11th National Conference on Networks, Intelligence and Computing Systems

SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Accredited by NBA & NAAC-UGC with 'A+' Grade
Coimbatore - 641 035

IJIRST
Innovation redefined

Department of Information Technology

CERTIFICATE

Paper ID: NCNICS19-68

This is to certify that Dr/Mr/Ms/Mrs ABISHEK.H

of

K.S.RANGASAMY COLLEGE OF TECHNOLOGY

has presented the paper titled

OBJECT DETECTION USING PRE-TRAINED MODEL IN TENSOR FLOW

in the National Conference on "Networks, Intelligence and Computing Systems" organized by the Department of Information Technology held on 7 th March 2019.

HOD	Principal	Chairman
(Dr.L.M.Nithya)	(Dr.S.Chenthur Pandian)	(Dr.S.N.Subbramanian)

NCNICS 2019

11th National Conference on Networks, Intelligence and Computing Systems

SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Accredited by NBA & NAAC-UGC with 'A+' Grade
Coimbatore - 641 035

IJIRST
Innovation redefined

Department of Information Technology

CERTIFICATE

Paper ID: NCNIC819 - 68

This is to certify that Dr/Mr/Ms/Mrs FARHANA . S

of

K.S. RANGASAMY COLLEGE OF TECHNOLOGY

has presented the paper titled

OBJECT DETECTION USING PRE-TRAINED MODEL IN TENSOR FLOW

in the National Conference on "Networks, Intelligence and Computing Systems" organized by the Department of Information Technology held on 7 th March 2019.

HOD	Principal	Chairman
(Dr.L.M.Nithya)	(Dr.S.Chenthur Pandian)	(Dr.S.N.Subbramanian)

NCNICS 2019

11th National Conference on Networks, Intelligence and Computing Systems

NCNICS 2019

SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Accredited by NBA & NAAC-UGC with 'A+' Grade
Coimbatore - 641 035

IJIRST
Innovation redefined

Department of Information Technology

CERTIFICATE

Paper ID: NCNICS 19-68

This is to certify that Dr/Mr/Ms/Mrs MONISHA .D.S

of

K.S. RANGASAMY COLLEGE OF TECHNOLOGY

has presented the paper titled

OBJECT DETECTION USING PRE-TRAINED MODEL IN TENSOR FLOW

in the National Conference on "Networks, Intelligence and Computing Systems" organized by the Department of Information Technology held on 7 th March 2019.

HOD	Principal	Chairman
(Dr.L.M.Nithya)	(Dr.S.Chenthur Pandian)	(Dr.S.N.Subbramanian)

NCNICS 2019

ISSN 2229-5518

CERTIFICATE OF ACCEPTANCE

International Journal of Scientific & Engineering Research (IJSER)

THIS IS TO CERTIFY THAT OUR REVIEW BOARD HAS ACCEPTED RESEARCH PAPER OF

Abishek Hari

Object Detection using pretrained model with tensorflow

March 15,2019

Visit us at: **www.ijser.org**

Editor in Chief

ISSN 2229-5518

CERTIFICATE OF ACCEPTANCE

International Journal of Scientific & Engineering Research (IJSER)

THIS IS TO CERTIFY THAT OUR REVIEW BOARD HAS ACCEPTED RESEARCH PAPER OF

Monisha Srikanth

Object Detection using pretrained model with tensorflow

March 15,2019

Visit us at: **www.ijser.org**

Editor in Chief

ISSN 2229-5518

CERTIFICATE OF ACCEPTANCE

International Journal of Scientific & Engineering Research (IJSER)

THIS IS TO CERTIFY THAT OUR REVIEW BOARD HAS ACCEPTED RESEARCH PAPER OF

Farhana Hameed

Object Detection using pretrained model with tensorflow

March 15,2019

Visit us at: **www.ijser.org**

Editor in Chief

APPENDIX A
SOURCE CODE

```
# coding: utf-8

# In[ ]:

# Modele 1: Static Object Detection

# File Name: object_detection.ipynb


import numpy as np

import os

import six.moves.urllib as urllib

import sys

import tarfile

import tensorflow as tf

import zipfile

from collections import defaultdict

from io import StringIO

from matplotlib import pyplot as plt

from PIL import Image

sys.path.append("..")

from object_detection.utils import ops as utils_ops

from utils import label_map_util

from utils import visualization_utils as vis_util
```

```
# In[ ]:

MODEL_NAME = 'ssd_mobilenet_v1_coco_2017_11_17'

MODEL_FILE = MODEL_NAME + '.tar.gz'

DOWNLOAD_BASE = 'http://download.tensorflow.org/models/object_detection/'

PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'

PATH_TO_LABELS = os.path.join('data', 'mscoco_label_map.pbtxt')

NUM_CLASSES = 90


# In[ ]:

opener = urllib.request.URLopener()

opener.retrieve(DOWNLOAD_BASE + MODEL_FILE, MODEL_FILE)

tar_file = tarfile.open(MODEL_FILE)

for file in tar_file.getmembers():

        file_name = os.path.basename(file.name)

        if 'frozen_inference_graph.pb' in file_name:

                tar_file.extract(file, os.getcwd())

detection_graph = tf.Graph()

with detection_graph.as_default():

        od_graph_def = tf.GraphDef()

        with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:

                serialized_graph = fid.read()

                od_graph_def.ParseFromString(serialized_graph)
```

```
            tf.import_graph_def(od_graph_def, name='')


# In[ ]:

label_map = label_map_util.load_labelmap(PATH_TO_LABELS)

categories = label_map_util.convert_label_map_to_categories(label_map,
        max_num_classes=NUM_CLASSES, use_display_name=True)

category_index = label_map_util.create_category_index(categories)


# In[ ]:

def load_image_into_numpy_array(image):

        (im_width, im_height) = image.size

        return np.array(image.getdata()).reshape(

                (im_height, im_width, 3)).astype(np.uint8)


# In[ ]:

PATH_TO_TEST_IMAGES_DIR = 'test_images'

TEST_IMAGE_PATHS = [ os.path.join(PATH_TO_TEST_IMAGES_DIR,
        'image{}.jpg'.format(i)) for i in range(1,9) ]

IMAGE_SIZE = (12, 8)


# In[ ]:

def run_inference_for_single_image(image, graph):

        with graph.as_default():
```

```
with tf.Session() as sess:
        # Get handles to input and output tensors
        ops = tf.get_default_graph().get_operations()
        all_tensor_names = {output.name for op in ops for output in op.outputs}
        tensor_dict = {}
        for key in [
                'num_detections', 'detection_boxes', 'detection_scores',
                'detection_classes', 'detection_masks'
        ]:
if 'detection_masks' in tensor_dict:
        # The following processing is only for single image
        detection_boxes = tf.squeeze(tensor_dict['detection_boxes'], [0])
        detection_masks = tf.squeeze(tensor_dict['detection_masks'], [0])
        # Reframe is required to translate mask from box coordinates to image
coordinates and fit the image size.
        real_num_detection = tf.cast(tensor_dict['num_detections'][0], tf.int32)
        detection_boxes = tf.slice(detection_boxes, [0, 0], [real_num_detection, -1])
        detection_masks = tf.slice(detection_masks, [0, 0, 0], [real_num_detection, -1, -
1])
        detection_masks_reframed = utils_ops.reframe_box_masks_to_image_masks(
                detection_masks, detection_boxes, image.shape[0], image.shape[1])
        detection_masks_reframed = tf.cast(
                tf.greater(detection_masks_reframed, 0.5), tf.uint8)
```

```
            # Follow the convention by adding back the batch dimension
            tensor_dict['detection_masks'] = tf.expand_dims(
                detection_masks_reframed, 0)
        image_tensor = tf.get_default_graph().get_tensor_by_name('image_tensor:0')
        # Run inference
        output_dict = sess.run(tensor_dict,
            feed_dict={image_tensor: np.expand_dims(image, 0)})
        # all outputs are float32 numpy arrays, so convert types as appropriate
        output_dict['num_detections'] = int(output_dict['num_detections'][0])
        output_dict['detection_classes'] = output_dict[
            'detection_classes'][0].astype(np.uint8)
        output_dict['detection_boxes'] = output_dict['detection_boxes'][0]
        output_dict['detection_scores'] = output_dict['detection_scores'][0]
        if 'detection_masks' in output_dict:
            output_dict['detection_masks'] = output_dict['detection_masks'][0]
        return output_dict


# In[ ]:
for image_path in TEST_IMAGE_PATHS:
        image = Image.open(image_path)
        # the array based representation of the image will be used later in order to prepare the
        # result image with boxes and labels on it.
        image_np = load_image_into_numpy_array(image)
```

```
        # Expand dimensions since the model expects images to have shape: [1, None, None, 3]
        image_np_expanded = np.expand_dims(image_np, axis=0)
        # Actual detection.
        output_dict = run_inference_for_single_image(image_np, detection_graph)
        plt.figure(figsize=IMAGE_SIZE)
        plt.imshow(image_np)


# coding: utf-8
# In[ ]:
```

 # Modele 2: Video Object Detection from MP4 files

 # File Name: video_object_detection.ipynb

```
#import necessary packages
import sys
import numpy as np
import os
import six.moves.urllib as urllib
import sys
import tarfile
import tensorflow as tf
import cv2

# In[ ]:
```

```
from object_detection.utils import label_map_util

from object_detection.utils import visualization_utils as vis_util


# In[ ]:

sys.path.append("..")

cap = cv2.VideoCapture("input_video.mp4")

fps = 30

capSize = (1280, 720)

out = cv2.VideoWriter()

success = out.open('output_video.avi',-1,fps,capSize,True)


# In[ ]:

MODEL_NAME = 'ssd_mobilenet_v1_coco_11_06_2017'

MODEL_FILE = MODEL_NAME + '.tar.gz'

DOWNLOAD_BASE = 'http://download.tensorflow.org/models/object_detection/'

PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'

PATH_TO_LABELS = os.path.join('data', 'mscoco_label_map.pbtxt')

NUM_CLASSES = 90


# In[ ]:

opener = urllib.request.URLopener()

opener.retrieve(DOWNLOAD_BASE + MODEL_FILE, MODEL_FILE)

tar_file = tarfile.open(MODEL_FILE)
```

```
for file in tar_file.getmembers():
        file_name = os.path.basename(file.name)
        if 'frozen_inference_graph.pb' in file_name:
                tar_file.extract(file, os.getcwd())
detection_graph = tf.Graph()
with detection_graph.as_default():
        od_graph_def = tf.GraphDef()
        with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
                serialized_graph = fid.read()
                od_graph_def.ParseFromString(serialized_graph)
                tf.import_graph_def(od_graph_def, name='')


# In[ ]:
label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(label_map,
        max_num_classes=NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)


# In[ ]:
def load_image_into_numpy_array(image):
        (im_width, im_height) = image.size
        return np.array(image.getdata()).reshape(
                (im_height, im_width, 3)).astype(np.uint8)
```

```
# In[ ]:
with detection_graph.as_default():
        with tf.Session(graph=detection_graph) as sess:
        while cap.isOpened():
                ret, image_np = cap.read()
                if ret == True:
                        image_np_expanded = np.expand_dims(image_np, axis=0)
                        image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')
                        boxes = detection_graph.get_tensor_by_name('detection_boxes:0')
                        scores = detection_graph.get_tensor_by_name('detection_scores:0')
                        classes = detection_graph.get_tensor_by_name('detection_classes:0')
                vis_util.visualize_boxes_and_labels_on_image_array(
                        image_np,
                        np.squeeze(boxes),
                        np.squeeze(classes).astype(np.int32),
                        np.squeeze(scores),
                cv2.imshow('Output',image_np)
                if cv2.waitKey(1) & 0xFF == ord('q'):
                        cv2.destroyAllWindows()
                        break
        else:
                break
```

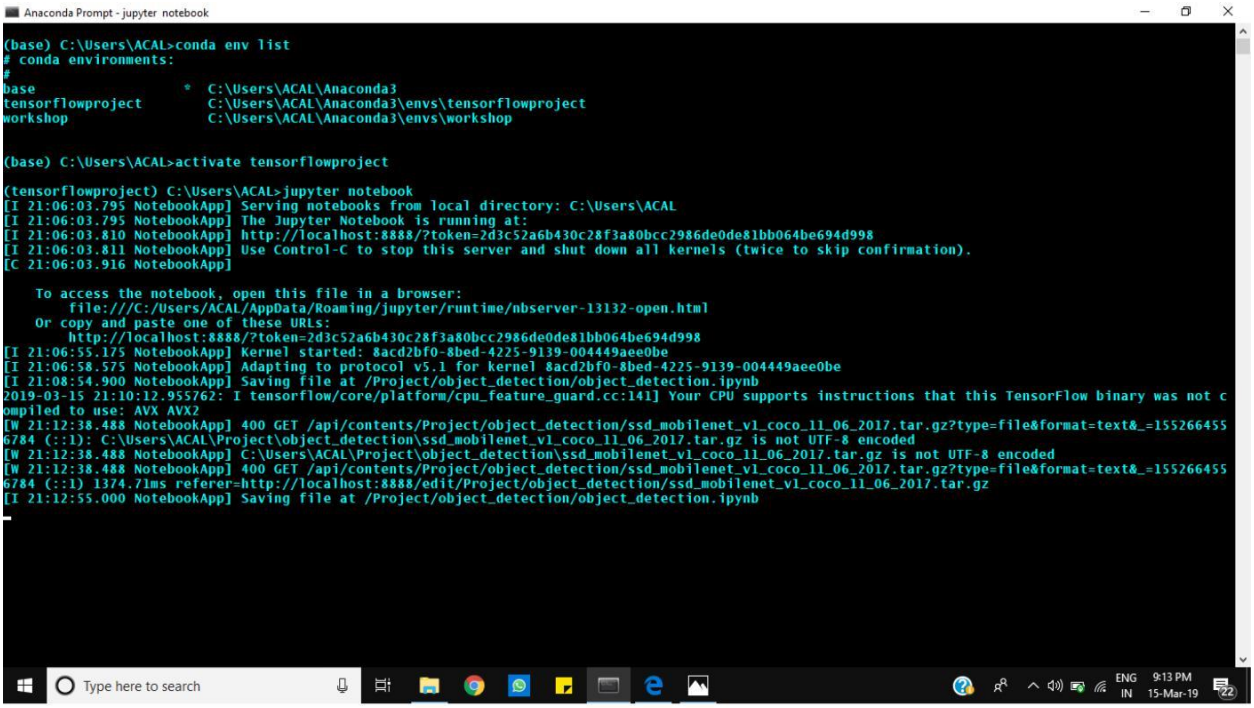
```
cap.release()

out.release()

cv2.destroyAllWindows()


# coding: utf-8

# In[ ]:

# Modele 3: Video Object Detection from Live Feed

# File Name: final_webcam.ipynb


import numpy as np

import os

import six.moves.urllib as urllib

import sys

import tarfile

import tensorflow as tf

import zipfile

from collections import defaultdict

from io import StringIO

from matplotlib import pyplot as plt

from PIL import Image

import cv2

cap = cv2.VideoCapture(0)

sys.path.append("..")
```

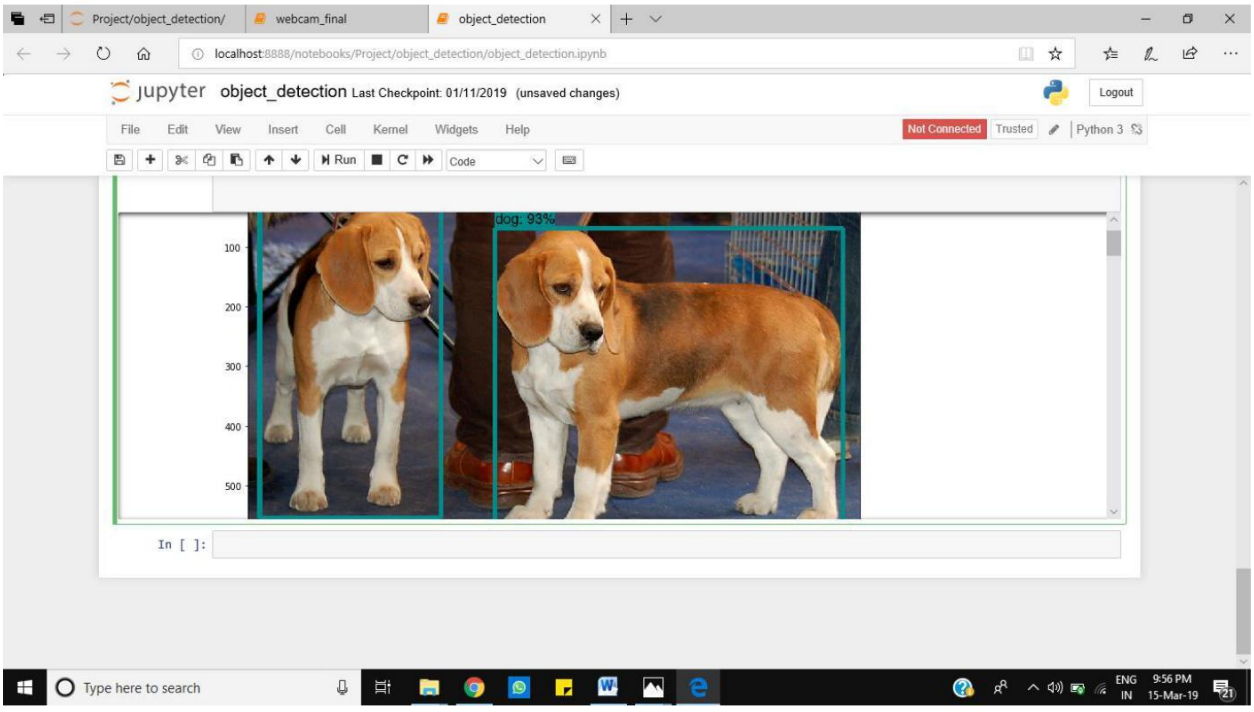
```
from object_detection.utils import ops as utils_ops

from utils import label_map_util

from utils import visualization_utils as vis_util


# In[ ]:

opener = urllib.request.URLopener()

opener.retrieve(DOWNLOAD_BASE + MODEL_FILE, MODEL_FILE)

tar_file = tarfile.open(MODEL_FILE)

for file in tar_file.getmembers():

        file_name = os.path.basename(file.name)

        if 'frozen_inference_graph.pb' in file_name:

                tar_file.extract(file, os.getcwd())

detection_graph = tf.Graph()

with detection_graph.as_default():

        od_graph_def = tf.GraphDef()

        with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:

                serialized_graph = fid.read()

                od_graph_def.ParseFromString(serialized_graph)

                tf.import_graph_def(od_graph_def, name='')


# In[ ]:

label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
```

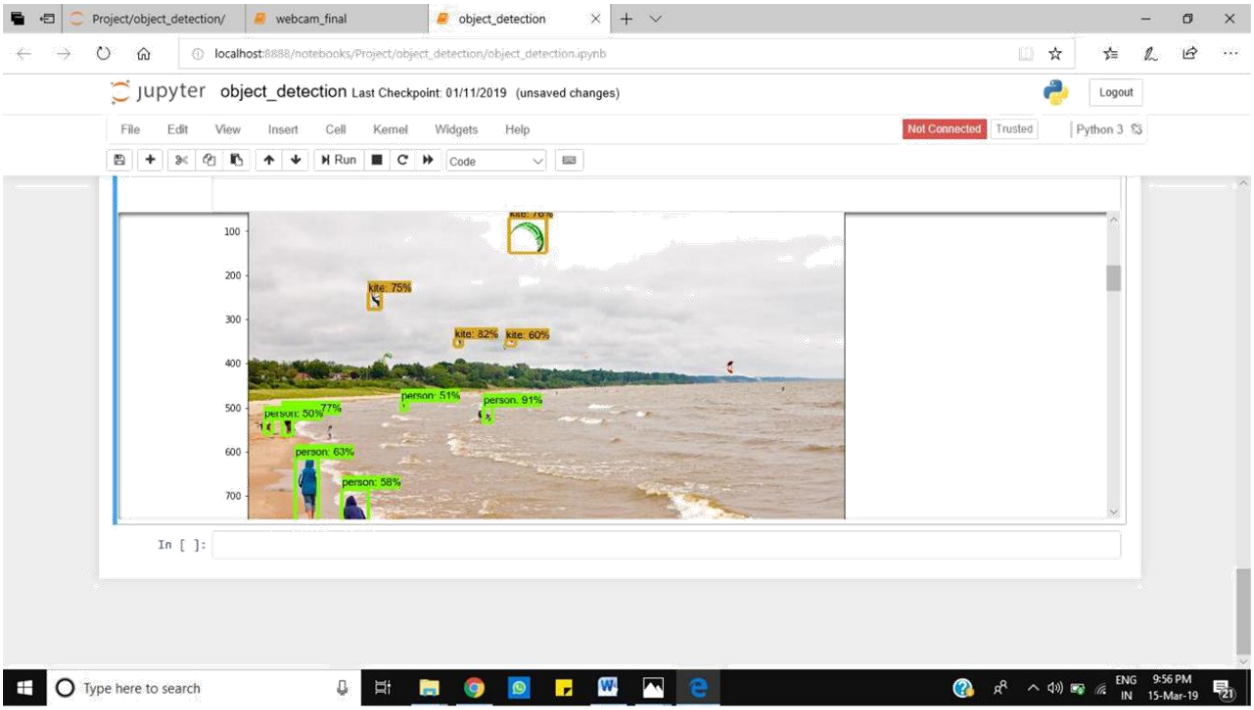
```
categories = label_map_util.convert_label_map_to_categories(label_map,
        max_num_classes=NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)


# In[ ]:
with detection_graph.as_default():
        with tf.Session(graph=detection_graph) as sess:
                while True:
                ret, image_np = cap.read()
                image_np_expanded = np.expand_dims(image_np, axis=0)
                image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')
                boxes = detection_graph.get_tensor_by_name('detection_boxes:0')
                scores = detection_graph.get_tensor_by_name('detection_scores:0')
                classes = detection_graph.get_tensor_by_name('detection_classes:0')
                vis_util.visualize_boxes_and_labels_on_image_array(
                        image_np,
                        np.squeeze(boxes),
                        np.squeeze(classes).astype(np.int32),
                        np.squeeze(scores),
                cv2.imshow('object detection', cv2.resize(image_np, (800, 600)))
                if cv2.waitKey(25) & 0xFF == ord('q'):
                        cv2.destroyAllWindows()
                        break
```

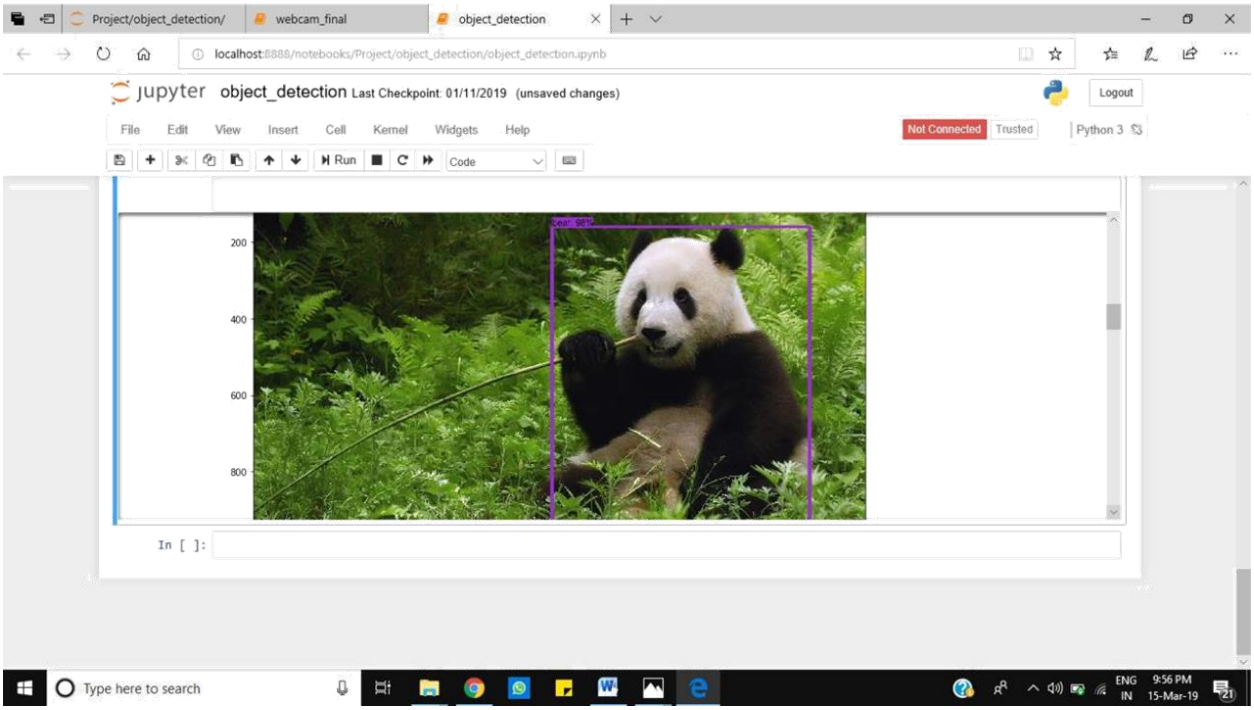
APPENDIX B

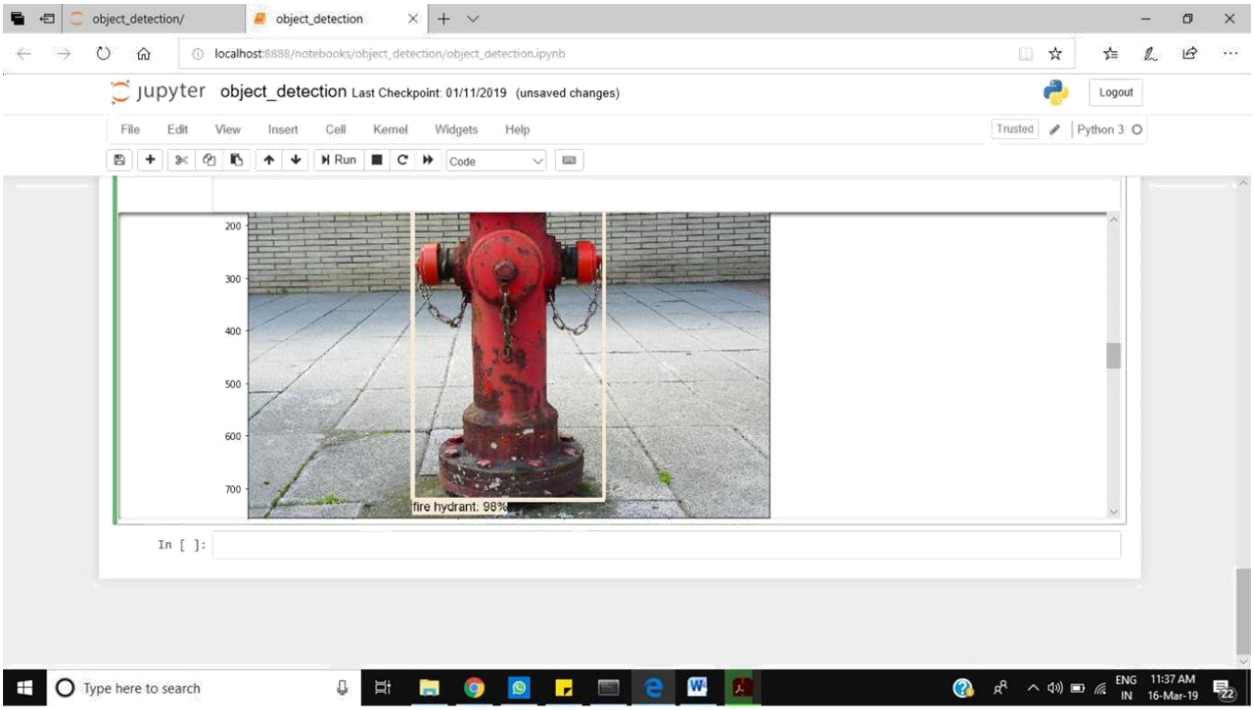
SCREENSHOTS OF OUTPUT

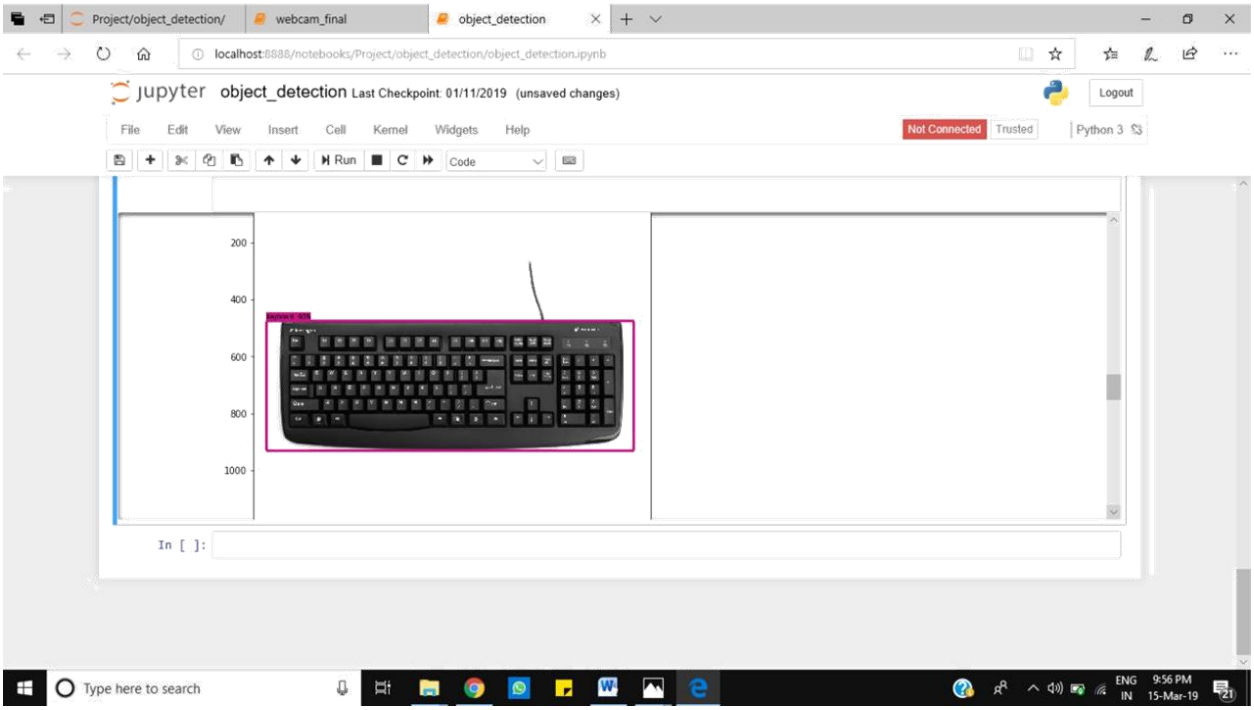


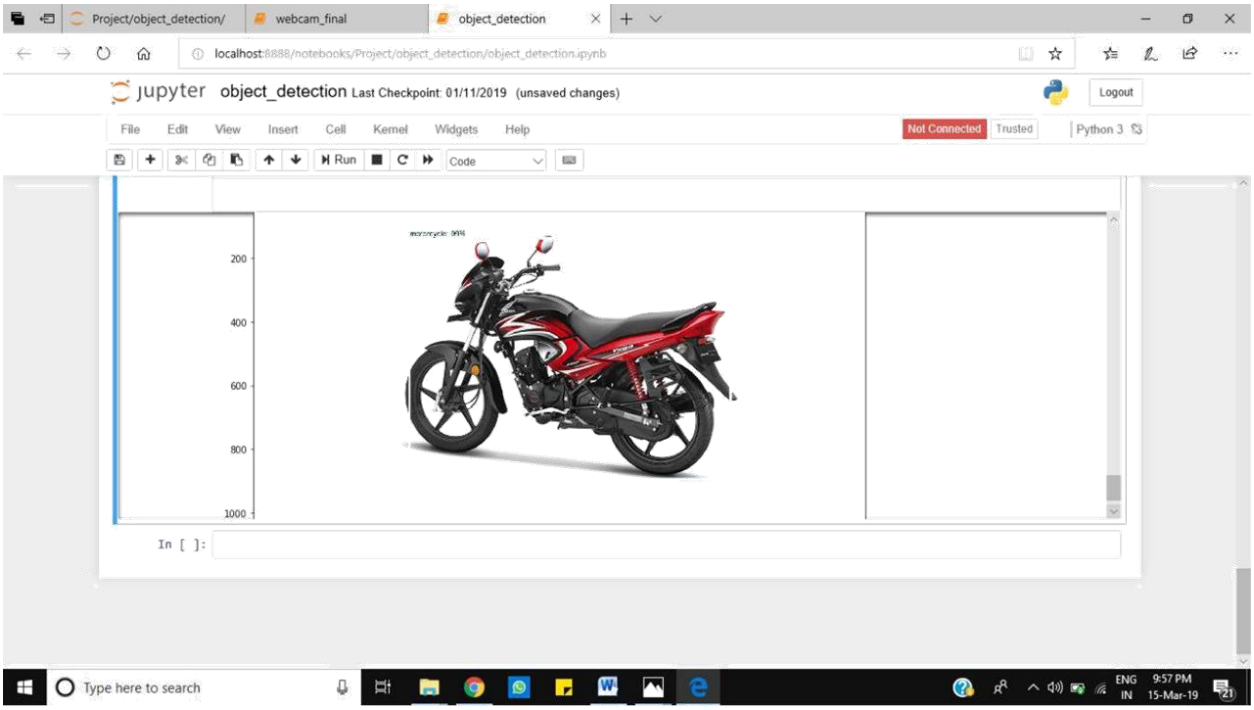


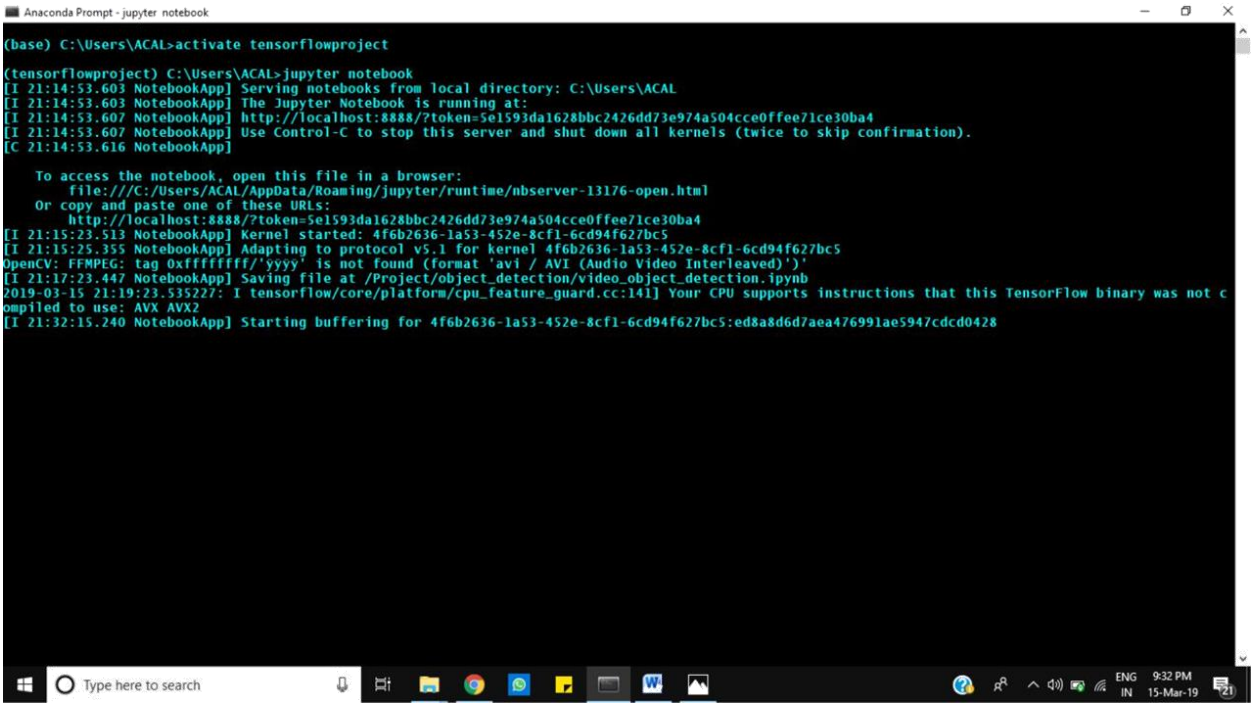

Project/object_detection/
webcam_final
object_detection
localhost:8888/notebooks/Project/object_detection/object_detection.ipynb
jupyter object_detection Last Checkpoint: 01/11/2019 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Connected
Trusted
Python 3
Run
Code
kite: 75%
kite: 82%
kite: 60%
person: 51%
person: 91%
person: 50%
person: 63%
person: 58%
In [ ]:
Type here to search
ENG IN
9:56 PM 15-Mar-19

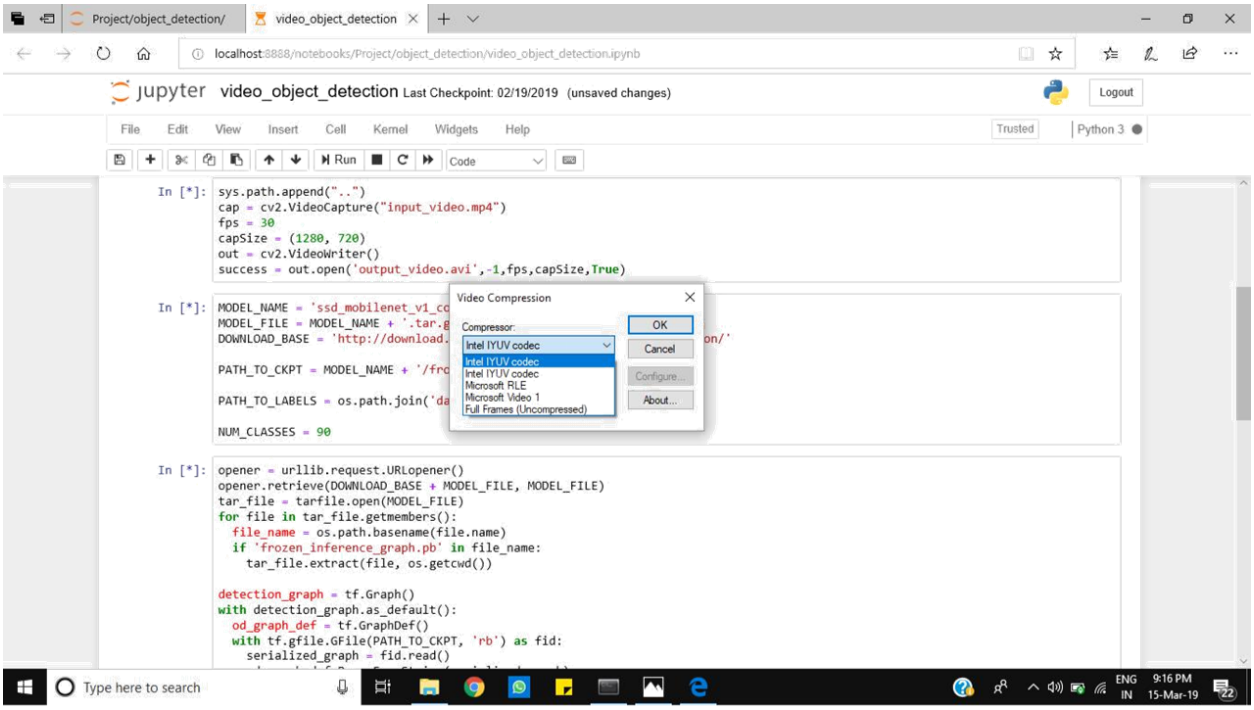

Project/object_detection/
webcam_final
object_detection
localhost:8888/notebooks/Project/object_detection/object_detection.ipynb
jupyter object_detection Last Checkpoint: 01/11/2019 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Connected
Trusted
Python 3
Run
Code
In [ ]:
Type here to search
ENG IN
9:56 PM 15-Mar-19

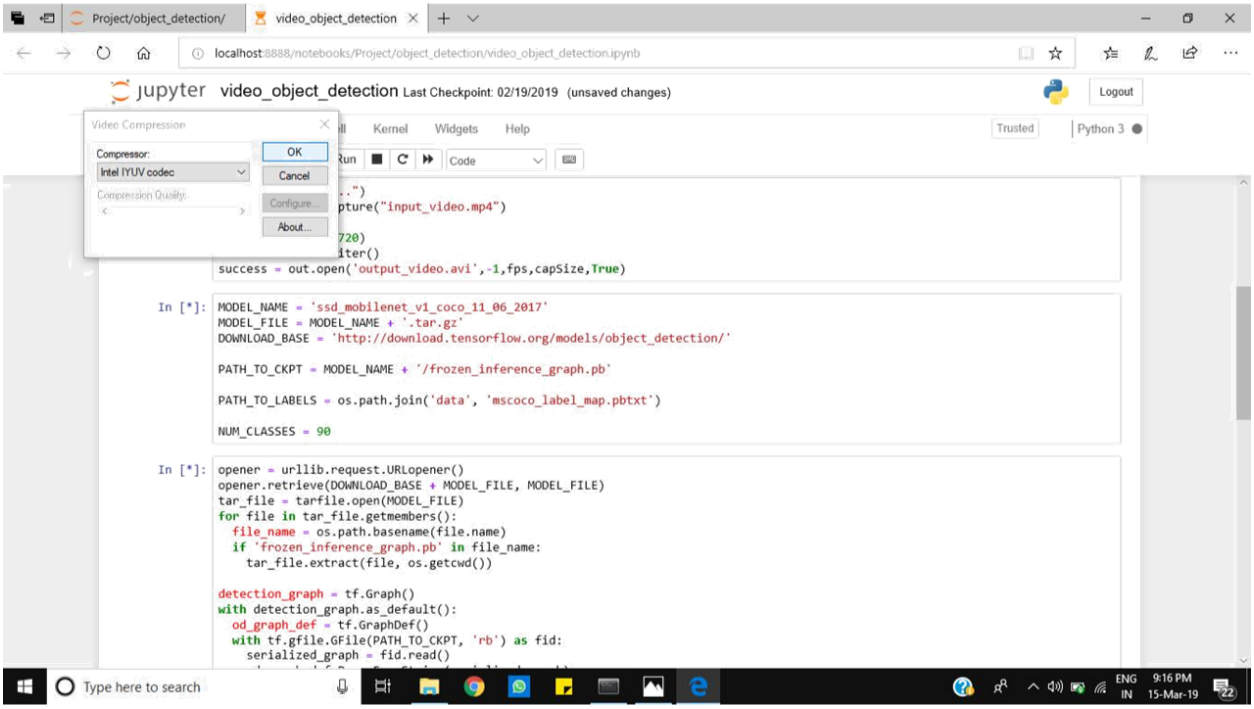

object_detection/
object_detection
localhost:8888/notebooks/object_detection/object_detection.ipynb
jupyter object_detection Last Checkpoint: 01/11/2019 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code
200
300
400
500
600
700
fire hydrant: 98%
In [ ]:
Type here to search
ENG IN 11:37 AM 16-Mar-19

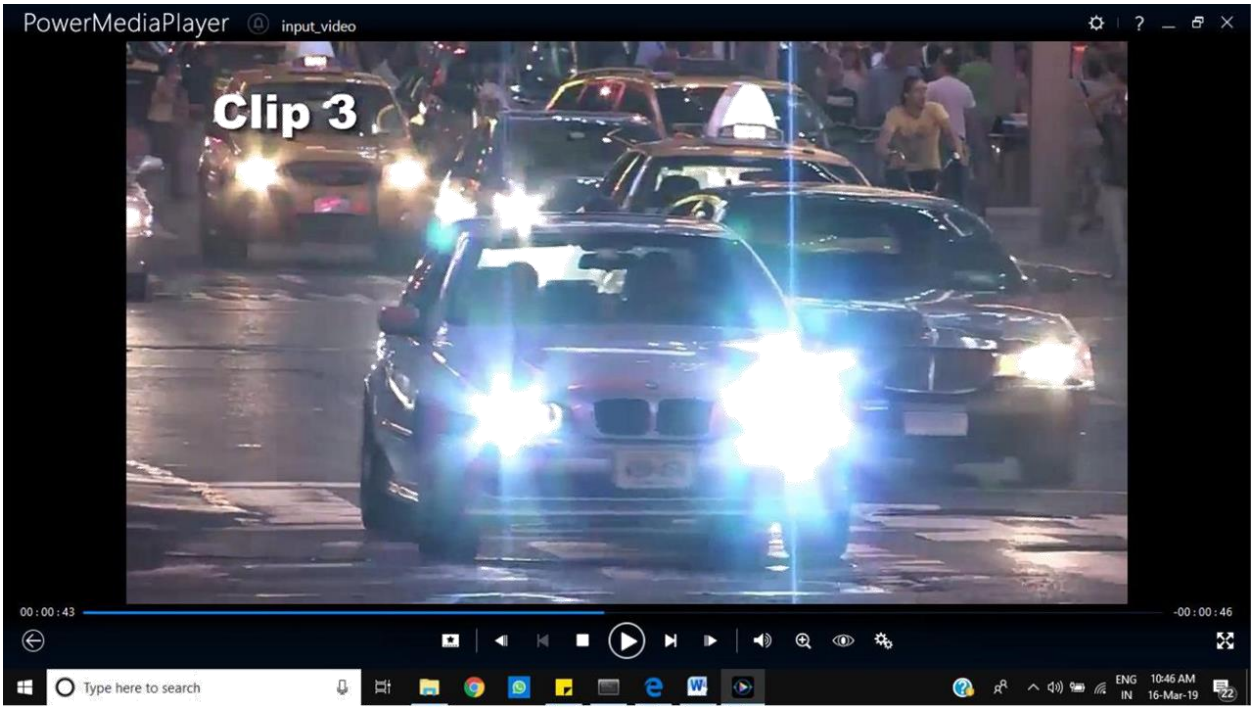

Project/object_detection/
webcam_final
object_detection
localhost:8888/notebooks/Project/object_detection/object_detection.ipynb
jupyter object_detection Last Checkpoint: 01/11/2019 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Connected Trusted Python 3
Run Code
200
400
600
800
1000
In [ ]:
Type here to search
ENG IN 9:56 PM 15-Mar-19

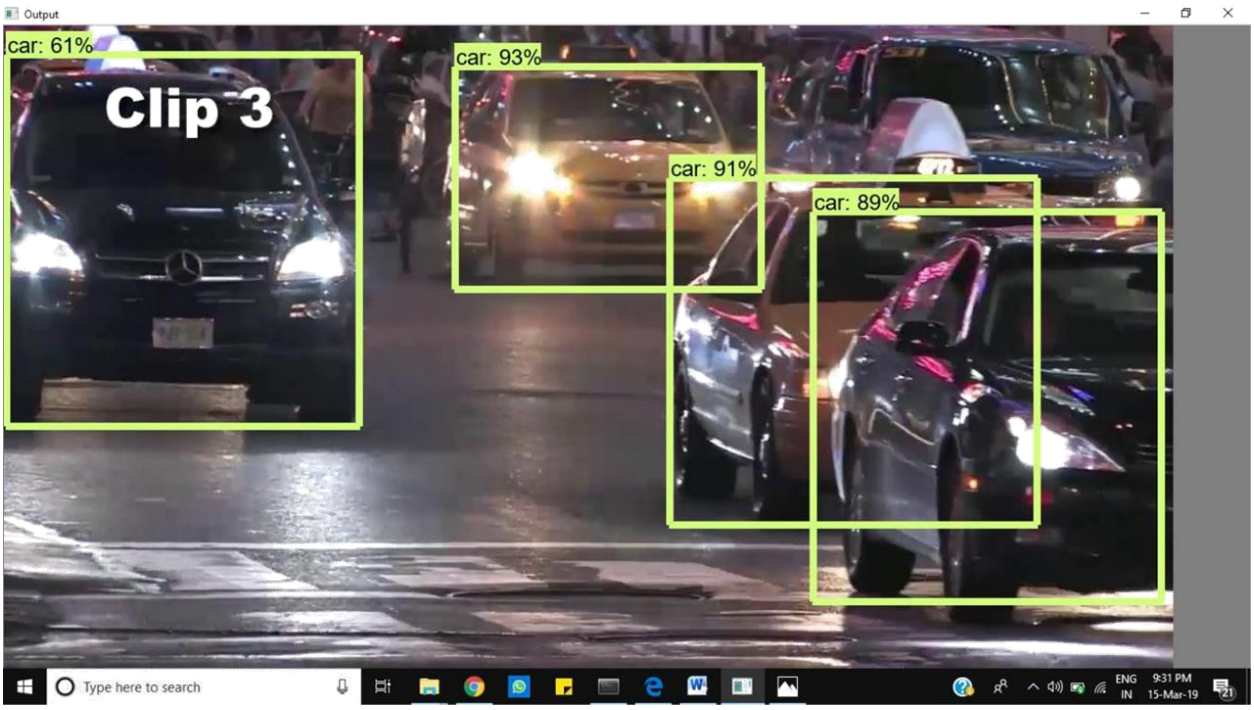

Project/object_detection/
webcam_final
object_detection
localhost:8888/notebooks/Project/object_detection/object_detection.ipynb
jupyter object_detection Last Checkpoint: 01/11/2019 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Connected
Trusted
Python 3
Run
Code
200
400
600
800
1000
In [ ]:
Type here to search
ENG IN
9:57 PM
15-Mar-19

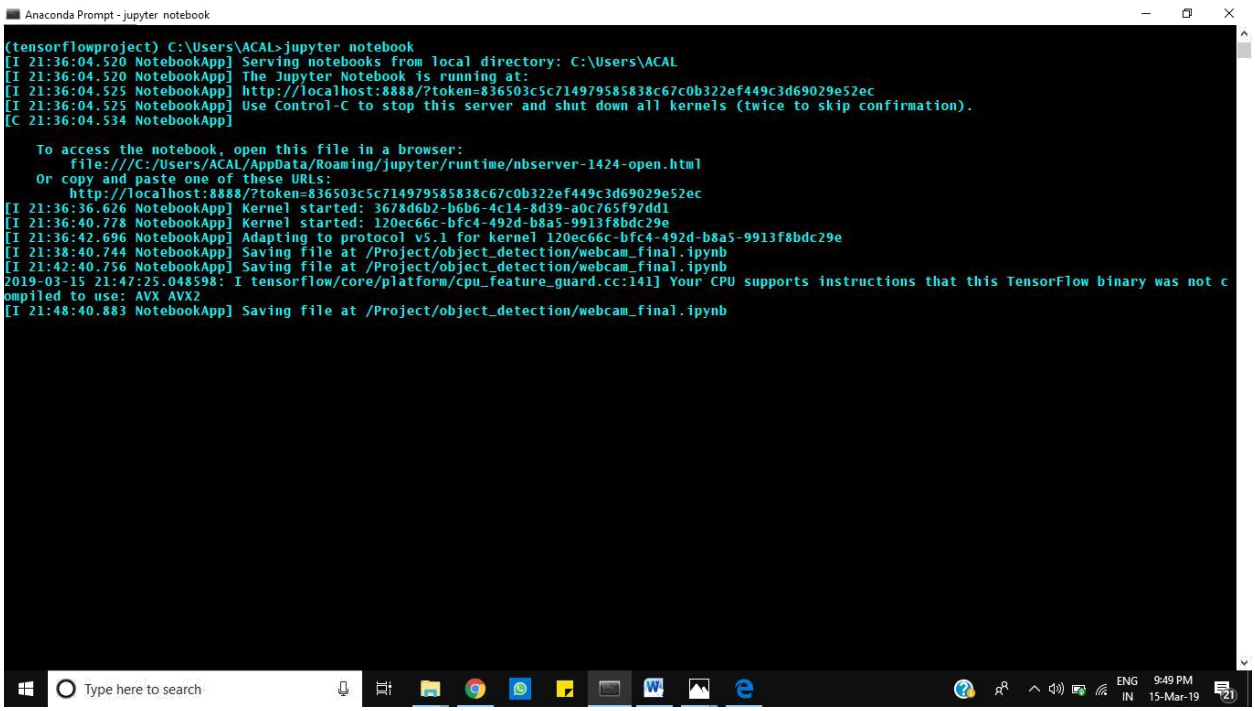

Anaconda Prompt - jupyter notebook
(base) C:\Users\ACAL>activate tensorflowproject
(tensorflowproject) C:\Users\ACAL>jupyter notebook
[I 21:14:53.603 NotebookApp] Serving notebooks from local directory: C:\Users\ACAL
[I 21:14:53.603 NotebookApp] The Jupyter Notebook is running at:
[I 21:14:53.607 NotebookApp] http://localhost:8888/?token=5e1593da1628bbc2426dd73e974a504cce0ffee71ce30ba4
[I 21:14:53.607 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 21:14:53.616 NotebookApp]
To access the notebook, open this file in a browser:
file:///C:/Users/ACAL/AppData/Roaming/jupyter/runtime/nbserver-13176-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=5e1593da1628bbc2426dd73e974a504cce0ffee71ce30ba4
[I 21:15:23.513 NotebookApp] Kernel started: 4f6b2636-1a53-452e-8cf1-6cd94f627bc5
[I 21:15:25.355 NotebookApp] Adapting to protocol v5.1 for kernel 4f6b2636-1a53-452e-8cf1-6cd94f627bc5
OpenCV: FFMPEG: tag 0xffffffff/'yyyy' is not found (format 'avi / AVI (Audio Video Interleaved)')'
[I 21:17:23.447 NotebookApp] Saving file at /Project/object_detection/video_object_detection.ipynb
2019-03-15 21:19:23.535227: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not c
ompiled to use: AVX AVX2
[I 21:32:15.240 NotebookApp] Starting buffering for 4f6b2636-1a53-452e-8cf1-6cd94f627bc5:ed8a8d6d7aea476991ae5947cdcd0428
Type here to search
ENG IN
9:32 PM
15-Mar-19


Project/object_detection/
video_object_detection
localhost:8888/notebooks/Project/object_detection/video_object_detection.ipynb
jupyter video_object_detection Last Checkpoint: 02/19/2019 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [*]:
sys.path.append("..")
cap = cv2.VideoCapture("input_video.mp4")
fps = 30
capSize = (1280, 720)
out = cv2.VideoWriter()
success = out.open('output_video.avi',-1,fps,capSize,True)
In [*]:
MODEL_NAME = 'ssd_mobilenet_v1_co
MODEL_FILE = MODEL_NAME + '.tar.g
DOWNLOAD_BASE = 'http://download.
PATH_TO_CKPT = MODEL_NAME + '/fro
PATH_TO_LABELS = os.path.join('da
NUM_CLASSES = 90
Video Compression
Compressor:
Intel IYUV codec
Intel IYUV codec
Intel IYUV codec
Microsoft RLE
Microsoft Video 1
Full Frames (Uncompressed)
OK
Cancel
Configure...
About...
In [*]:
opener = urllib.request.URLopener()
opener.retrieve(DOWNLOAD_BASE + MODEL_FILE, MODEL_FILE)
tar_file = tarfile.open(MODEL_FILE)
for file in tar_file.getmembers():
file_name = os.path.basename(file.name)
if 'frozen_inference_graph.pb' in file_name:
tar_file.extract(file, os.getcwd())
detection_graph = tf.Graph()
with detection_graph.as_default():
od_graph_def = tf.GraphDef()
with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
serialized_graph = fid.read()
Type here to search
ENG
IN
9:16 PM
15-Mar-19


Project/object_detection/
video_object_detection
localhost:8888/notebooks/Project/object_detection/video_object_detection.ipynb
jupyter video_object_detection Last Checkpoint: 02/19/2019 (unsaved changes)
Logout
Trusted
Python 3
Code
Video Compression
Compressor:
Intel IYUV codec
Compression Quality:
OK
Cancel
Configure...
About...
success = out.open('output_video.avi',-1,fps,capSize,True)
In [*]:
MODEL_NAME = 'ssd_mobilenet_v1_coco_11_06_2017'
MODEL_FILE = MODEL_NAME + '.tar.gz'
DOWNLOAD_BASE = 'http://download.tensorflow.org/models/object_detection/'
PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'
PATH_TO_LABELS = os.path.join('data', 'mscoco_label_map.pbtxt')
NUM_CLASSES = 90
In [*]:
opener = urllib.request.URLopener()
opener.retrieve(DOWNLOAD_BASE + MODEL_FILE, MODEL_FILE)
tar_file = tarfile.open(MODEL_FILE)
for file in tar_file.getmembers():
file_name = os.path.basename(file.name)
if 'frozen_inference_graph.pb' in file_name:
tar_file.extract(file, os.getcwd())
detection_graph = tf.Graph()
with detection_graph.as_default():
od_graph_def = tf.GraphDef()
with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
serialized_graph = fid.read()
Type here to search
ENG
IN
9:16 PM
15-Mar-19


PowerMediaPlayer
input_video
Clip 3
00 : 00 : 43
-00 : 00 : 46
Type here to search
ENG
IN
10:46 AM
16-Mar-19


Output
car: 61%
Clip 3
car: 93%
car: 91%
car: 89%
Type here to search
ENG
IN
9:31 PM
15-Mar-19


Anaconda Prompt - jupyter notebook
(tensorflowproject) C:\Users\ACAL>jupyter notebook
[I 21:36:04.520 NotebookApp] Serving notebooks from local directory: C:\Users\ACAL
[I 21:36:04.520 NotebookApp] The Jupyter Notebook is running at:
[I 21:36:04.525 NotebookApp] http://localhost:8888/?token=836503c5c714979585838c67c0b322ef449c3d69029e52ec
[I 21:36:04.525 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 21:36:04.534 NotebookApp]
To access the notebook, open this file in a browser:
file:///C:/Users/ACAL/AppData/Roaming/jupyter/runtime/nbserver-1424-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=836503c5c714979585838c67c0b322ef449c3d69029e52ec
[I 21:36:36.626 NotebookApp] Kernel started: 3678d6b2-b6b6-4c14-8d39-a0c765f97dd1
[I 21:36:40.778 NotebookApp] Kernel started: 120ec66c-bfc4-492d-b8a5-9913f8bdc29e
[I 21:36:42.696 NotebookApp] Adapting to protocol v5.1 for kernel 120ec66c-bfc4-492d-b8a5-9913f8bdc29e
[I 21:38:40.744 NotebookApp] Saving file at /Project/object_detection/webcam_final.ipynb
[I 21:42:40.756 NotebookApp] Saving file at /Project/object_detection/webcam_final.ipynb
2019-03-15 21:47:25.048598: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not c
ompiled to use: AVX AVX2
[I 21:48:40.883 NotebookApp] Saving file at /Project/object_detection/webcam_final.ipynb
Type here to search
ENG
IN
9:49 PM
15-Mar-19

REFERENCES

[1] [Online] : scholar.google.fr/scholar?hl=fr&q=Object+detection+using+Haar−cascade

[2] [Online] : lab.cntl.kyutech.ac.jp/kobalab/nishida/opencv/OpenCVObjectDetectionHowTo.pdf

[3] [Online] : cs.colby.edu/maxwell/courses/cs397−vision/F07/papers/viola−Faces−cvpr01.pdf

[4] [Online] : cbcl.mit.edu/publications/ps/heisele−x3hei.lo.pdf

[5] [Online] : arxiv.org/pdf/1502.05461v1.pdf

[6] [Online] : www.researchgate.net/profile/VitorSantos6/publication/267863282Automatic DetectionofCarsinRealRoadsusingHaar−likeFeatures/links/552c0bae0cf2e089a3ac3bc3.pdf

[7] [Online] : www.svcl.ucsd.edu/projects/traffic/

[8] [Online] : github.com/andrewssobral/bgslibrary#bgslibrary

[9] [Online] : github.com/andrewssobral/simplevehiclecounting

[10] [Online] : github.com/andrewssobral/bgslibrary

[11] C.P.Papageorgiou, M.Oren, T.Poggio, A general framework for object detection, in: ICCV ’98: Proceedings of the International Conference on Computer Vision, Washington, DC, USA, 1998, pp. 555-562.

[12] S.Z.Li, L. Zhu, Z.Zhang, A.Blake, H.Zhang, H.Shum, Statistical learning of multiview face detection, in: ECCV ’02: Proceedings of the European Conference on Computer Vision, Lecture Notes in Computer Sciences, vol. 2353, London, UK, 2002, pp. 67-81.

[13] R.Lienhart, J.Maydt, An extended set of Haar-like features for rapid object detection, in: ICIP ’02: Proceedings of the International Conference on Image Processing, 2002, pp. 900903

[14] P.Viola, M.J.Jones, Rapid object detection using a boosted cascade of simple features, in: CVPR ’01: Proceedings of the Conference on Computer Vision and Pattern Recognition, Los Alamitos, CA, USA, 2001, pp. 511-518.

[15] M.Jones, P.Viola, Fast multi-view face detection, Technical Report MERLTR2003-96, Mitsubishi Electric Research Laboratories, July 2003.

[16] Viola,P.; Jones,M., ”Rapid object detection using a boosted cascade of simple features,” Computer Vision and Pattern Recognition (CVPR), 2001 IEEE Conference on, June 2001.

[17] [Online] : ir.library.louisville.edu/cgi/ viewcontent.cgi?article=2731&context=etd