JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

REAL-TIME HUMAN DETECTION & COUNTING

[1]Mrs.M.Angel Shalini, [2]H.Linda Evangelin,

[1]Assistant professor, [2]Research Scholar.
[1]PG And Research Department of Computer Applications,,
[1]Hindusthan College of Arts And Science, Coimbatore, Tamilnadu, India.

ABSTRACT

Nowadays population growth increased. The linear growth of population has resulted in a large number of people comes to public places. This system will thus provide the count of the person in particular area malls, supermarket, etc. some business is only depending upon on the customer schedule and its timing. Hence, our work satisfies the problem and provides a solution by designing a people counting system. So, the single shot detector (SSD) mobile net is proposed along with a centroid tracker. This model replaces the VGG16 base network with a Mobile Net network for better extracting features, and after the base network, it connects with six convolutional layers for classification. The centroid tracking algorithm obtains bounding box coordinates from an object detector SSD and applies them to calculate the center of a bounding box. After the centroid is calculated, it will assign an id to each person and this model operates the dataset with training and testing the data. The maximum true positive rate (TPR) is 95.03% and FPR (false positive rate) is 0.08%. A total of 2416 positive images samples and 1218 negative samples. After that, the accuracy of the model is obtained and also 96.64% accuracy rate is obtained by testing the given data set. Sometimes method do not produce high accuracy due to the high people counting, clothes of person, shadow issue etc

INTRODUCTION

1.1. Overview of the project

There are many ways for people detection and counting in a scene. Manual counting is not a good approach as most of the public places like malls and shopping centres have dense crowd. People counting systems mostly used today in public places are based on thermal sensors and tally counters. These systems can count only people passing through a passage and cannot be extended to wider area. Now-a-days these systems are also becoming unreliable due to occlusion. When two people walk through the tally counter at the same time, they produce inaccurate results. Thus research has focus towards vision based counting as it is relatively inexpensive and provides more information, does not have a direct contact with the people who move. The main idea behind video based people counting is detection and tracking. Detection involves locating the people in the scene and tracking is done after detection to determine the location of people along the video sequence. The existing methods on video based people counting are background modelling, body part detection and body shape detection. In background modelling, the foreground is obtained by subtracting the background from the particular frame, but this method fails due to occlusion. Body part detection uses the method of dividing the body of the person into separate parts as head, arms, upper body and legs. Each part is detected separately and using reconstruction it is determined whether the detection is a person or not.

1.2. Module Description

- An application which allows to detect human or people's in any image, video or even through Camera.
- Also along with detecting, it also counts the maximum no. of people appeared.
- It also visualize the data of no. of human detected and the avg. accuracy with which humans got detected w.r.t. time, using graph plot

SYSTEM STUDY

2.1. Existing System

Accurately counting objects instances in a given image or video frame is a hard problem to solve in machine learning. A number of solutions have been developed to count people, cars and other objects and none of them is perfect. Of course, we are talking about image processing here, so a neural network seems to be a good tool for the job. Below you can find a description of different approaches, common problems, challenges and latest solutions in the Neural Networks object counting field. As a proof of concept, existing model for Faster R-CNN network will be used to count objects on the street with video examples given at the end of the post. But it is slow and less accurate.

2.1.1. Disadvantages of existing system

1. Highly coding difficult.

2. Not useful in airports, railway station, bus stands.

3. Cannot operate from other places.

4. Needs more memory to execute and communicate

2.2. Proposed System

The proposed system involves in SSD with Deep neural networks. The direct development of populace has brought about countless individuals comes to public spots. This framework will consequently give the include of the individual specifically region shopping centers, general store, and so forth some business is just relying upon the client plan and its planning. Henceforth, our work fulfills the issue and gives an answer by planning a group counting framework. Thus, the single shot identifier (SSD) portable net is proposed alongside a centroid tracker. This model replaces the VGG16 base organization with a Mobile Net organization for better removing highlights, and after the base organization, it associates with six convolutional layers for order. The centroid following calculation acquires bouncing box facilitates from an item identifier SSD and applies them to compute the focal point of a jumping box. After the centroid is determined, it will allot an id to every individual and this model works the dataset with preparing and testing the information.

2.2.1. Advantages of proposed system

1. Makes implementation super easy

2.Provides continuous surveillance.

3. It is a very efficient way of delivering alerts to admin.

4.As you have access to the net 24*7,you can train system anytime and from anywhere a

SYSTEM SPECIFICATIONS

3.1 Hardware Specifications:

PROCESSOR	: I3
RAM	: 2GB RAM
MONITOR	: 15” COLOR
HARD DISK	: 120 GB
CDDRIVE	: LG 52X
KEYBOARD	: STANDARD 102 KEYS
MOUSE	: 3 BUTTONS

3.2 Software Specifications:

OPERATING SYSTEM	: Windows 7
FRONT END	: Python

SYSTEM DESIGN AND DEVELOPMENT

4.1 Input Design

Input design is the process of converting user originated inputs to a computer based format. input design is one of the most expensive phases of the operation of computerized system and is often the major problem of a system.

In the project, the input design is made in various web forms with various methods. For example, in the user creation form, the empty username and password is not allowed. The username if exists in the database, the input is considered to be invalid and is not accepted.

4.2 Output Design

Output design generally refers to be the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis is on which they evaluate the usefulness of the application. In this project, the mail details, the greetings details, and the other details are the web forms in which the output is available.

SYSTEM DEVELOPEMENT

A Systems Development Life Cycle (SDLC) adheres to important phases that are essential for developers, such as planning, analysis, design, and implementation, and are explained in the section below. A number of system development life cycle (SDLC) models have been created: waterfall, fountain, and spiral, build and fix, rapid prototyping, incremental, and synchronize and stabilize. The oldest of these, and the best known, is the waterfall model: a sequence of stages in which the output of each stage becomes the input for the next.

The waterfall model is a popular version of the systems development life cycle model for software engineering. Often considered the classic approach to the systems development life cycle, the waterfall model describes a development method that is linear and sequential. Waterfall development has distinct goals for each phase of development. Imagine a waterfall on the cliff of a steep mountain. Once the water has flowed over the edge of the cliff and has begun its journey down the side of the mountain, it cannot turn back. It is the same with waterfall development. Once a phase of development is completed, the development proceeds to the next phase and there is no turning back.

The advantage of waterfall development is that it allows for departmentalization and managerial control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process like a car in a carwash, and theoretically, be delivered on time. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order, without any overlapping.

SYSTEM TESTING AND IMPLEMENTATION

System Testing

The most important phase in system development life cycle is system testing. The number and nature of errors in a newly designed system depend on the system specifications and the time frame was given for the design.

A newly designed system should have all the subsystems working together, but in reality, each subsystem work independently. During this phase, all the subsystems are gathered into one poo an tested to determine whether it meets the user requirements.

Testing is done at two-level -Testing of individual modules and testing the entire system. During the system testing, the system is used experimentally to ensure that the software will run according to the specifications and in the way the user expects. Each test case is designed with the intent of finding errors in the way the system will process it.

Testing plays a very critical role in determining the reliability and efficiency of software and hence is a very important stage in software development. Software testing is done at different levels. They are the unit testing and system testing which comprises of integration testing and acceptance testing.

TYPES OF TESTING

Unit Testing

This is the first level of testing. The different modules are tested against the specifications produced during the integration. This is done to test the internal logic of each module. Those resulting from the interaction between modules are initially avoided. The input received and output generated is also tested to see whether it falls in the expected range of values. Unit testing is performed from the bottom up, starting with the smallest and lowest modules and proceeding one at a time.

The units in a system are the modules and routines that are assembled and integrated to perform a specific function. The programs are tested for correctness of logic applied and detection of errors in coding. Each of the modules was tested and errors are rectified. They were then found to function properly.

Integration Testing

In integration testing, the tested modules are combined into sub-systems, which are then tested. The goal of integration testing to check whether the modules can be integrated properly emphasizing on the interfaces between modules. The different modules were linked together and integration testing done on them.

Validation Testing

The objective of the validation test is to tell the user about the validity and reliability of the system. It verifies whether the system operates as specified and the integrity of important data is maintained. User motivation is very important for the successful performance of the system.

All the modules were tested individually using both test data and live data. After each module was ascertained that it was working correctly and it had been "integrated" with the system. Again the system was tested as a whole. We hold the system tested with different types of users. The System Design, Data Flow Diagrams, procedures etc. were well documented so that the system can be easily maintained and upgraded by any computer professional at a later

System Testing

The integration of each module in the system is checked during this level of testing. The objective of system testing is to check if the software meets its requirements. System testing is done to uncover errors that were not found in earlier tests. This includes forced system failures and validation of total system as the user in the operational environment implements it. Under this testing, low volumes of transactions are generally based on live data. This volume is increased until the maximum level for each transactions type is reached. The total system is also tested for recovery after various major failures to ensure that no data are lost during the breakdown.

SYSTEM IMPLEMENTATION

Implementation is the process that actually yields the lowest-level system elements in the system hierarchy (system breakdown structure). The system elements are made, bought, or reused. Production involves the hardware fabrication processes of forming, removing, joining, and finishing; or the software realization processes of coding and testing; or the operational procedures development processes for operators' roles. If implementation involves a production process, a manufacturing system which uses the established technical and management processes may be required.

The purpose of the implementation process is to design and create (or fabricate) a system element conforming to that element's design properties and/or requirements. The element is constructed employing appropriate technologies and industry practices. This process bridges the system definition processes and the integration process.

System Implementation is the stage in the project where the theoretical design is turned into a working system. The most critical stage is achieving a successful system and in giving confidence on the new system for the user that it will work efficiently and effectively. The existing system was long time process.

The proposed system was developed using Python. The existing system caused long time transmission process but the system developed now has a very good user-friendly tool, which has a menu-based interface, graphical interface for the end user. After coding and testing, the project is to be installed on the necessary system. The executable file is to be created and loaded in the system. Again the code is tested in the installed system. Installing the developed code in system in the form of executable file is implementation.

SYSTEM ENVIRONMENT

LANGUAGE FEATURES

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Statements and control flow

The assignment statement (token '=', the equals sign). This operates differently than in traditional imperative programming languages, and this fundamental mechanism (including the nature of Python's version of variables) illuminates many other features of the language. Assignment in C, e.g., x = 2, translates to "typed variable name x receives a copy of numeric value 2". The (right-hand) value is copied into an allocated storage location for which the (left-hand) variable name is the symbolic address. The memory allocated to the variable is large enough (potentially quite large) for the declared type. In the simplest case of Python assignment, using the same example, x = 2, translates to "(generic) name x receives a reference to a separate, dynamically allocated object of numeric (int) type of value 2." This is termed binding the name to the object. Since the name's storage location doesn't contain the indicated value, it is improper to call it a variable. Names may be subsequently rebound at any time to objects of greatly varying types, including strings, procedures, complex objects with data and methods, etc. Successive assignments of a common value to multiple names, e.g., x = 2; y = 2; z = 2 result in allocating storage to (at most) three names and one numeric object, to which all three names are bound. Since a name is a generic reference holder it is unreasonable to associate a fixed data type with it. However at a given time a name will be bound to some object, which will have a type; thus there is dynamic typing.

The if statement, which conditionally executes a block of code, along with else and elif (a contraction of else-if). The for statement, which iterates over an iterable object, capturing each element to a local variable for use by the attached block. The while statement, which executes a block of code as long as its condition is true.

The try statement, which allows exceptions raised in its attached code block to be caught and handled by except clauses; it also ensures that clean-up code in a finally block will always be run regardless of how the block exits. The class statement, which executes a block of code and attaches its local namespace to a class, for use in object-oriented programming. The def statement, which defines a function or method. The with statement (from Python 2.5), which encloses a code block within a context manager (for example, acquiring a lock before the block of code is run and releasing the lock afterwards, or opening a file and then closing it), allowing Resource Acquisition Is Initialization (RAII)-like behavior. The pass statement, which serves as a NOP. It is syntactically needed to create an empty code block. The assert statement, used during debugging to check for conditions that ought to apply. The yield statement, which returns a value from a generator function. From Python 2.5, yield is also an operator. This form is used to implement coroutines.

The import statement, which is used to import modules whose functions or variables can be used in the current program. There are four ways of using import: import <module name> or from <module name> import * or import numpy as np or from numpy import pi as Pie. The print statement was changed to the print() function in Python 3. Python does not support tail call optimization or first-class continuations, and, according to Guido van Rossum, it never will. However, better support for coroutine-like functionality is provided in 2.5, by extending Python's generators. Before 2.5, generators were lazy iterators; information was passed unidirectionally out of the generator. From Python 2.5, it is possible to pass information back into a generator function, and from Python 3.3, the information can be passed through multiple stack levels.

Tkinter

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit,[1] and is Python's de facto standard GUI.[2] Tkinter is included with the standard Microsoft Windows and Mac OS X install of Python.

The name Tkinter comes from Tk interface. Tkinter was written by Fredrik Lundh.

As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application.

Python 2.7 and Python 3.1 incorporate the "themed Tk" ("ttk") functionality of Tk 8.5. This allows Tk widgets to be easily themed to look like the native desktop environment in which the application is running, thereby addressing a long-standing criticism of Tk (and hence of Tkinter).

tkinter provides us with a variety of common GUI elements which we can use to build our interface – such as buttons, menus and various kinds of entry fields and display areas. We call these elements widgets. We are going to construct a tree of widgets for our GUI – each widget will have a parent widget, all the way up to the root window of our application. For example, a button or a text field needs to be inside some kind of containing window.

The widget classes provide us with a lot of default functionality. They have methods for configuring the GUI’s appearance – for example, arranging the elements according to some kind of layout – and for handling various kinds of user-driven events. Once we have constructed the backbone of our GUI, we will need to customise it by integrating it with our internal application class.

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit,[1] and is Python's de facto standard GUI.[2] Tkinter is included with the standard Microsoft Windows and Mac OS X install of Python.

The name Tkinter comes from Tk interface. Tkinter was written by Fredrik Lundh.

As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application.

Python 2.7 and Python 3.1 incorporate the "themed Tk" ("ttk") functionality of Tk 8.5. This allows Tk widgets to be easily themed to look like the native desktop environment in which the application is running, thereby addressing a long-standing criticism of Tk (and hence of Tkinter).

tkinter provides us with a variety of common GUI elements which we can use to build our interface – such as buttons, menus and various kinds of entry fields and display areas. We call these elements widgets. We are going to construct a tree of widgets for our GUI – each widget will have a parent widget, all the way up to the root window of our application. For example, a button or a text field needs to be inside some kind of containing window.

The widget classes provide us with a lot of default functionality. They have methods for configuring the GUI's appearance – for example, arranging the elements according to some kind of layout – and for handling various kinds of user-driven events. Once we have constructed the backbone of our GUI, we will need to customise it by integrating it with our internal application class.

Section 1.01 What is Python?

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to www browsers to games.

Section 1.02 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python's features include −

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.

- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.
- Apart from the above-mentioned features, Python has a big list of good features, few are listed below –
- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Article II. My-SQL

Article III. MySQL is a simple, yet powerful Open Source Software relational database management system that uses SQL. MySQL is a true multi-user, multithreaded SQL database server. MySQL is fast and flexible enough to store logs and pictures in it. Its main goals are speed, robustness, and ease of use. Most likely MySQL and PHP combination is encountered today and probably for the years to come.

This basic MySQL explains some of the basic SQL statements. If this is the first time you have used a relational database management system, this tutorial gives you everything you need to know to work with MySQL such as querying data, updating data, managing databases, and creating tables. If you are already familiar with other relational database management systems such as PostgreSQL, Oracle, and Microsoft SQL Server. you can use this tutorial to refresh your knowledge and understand how SQL dialect of MySQL is different from other systems.

Strengths of MySQL

- MySQL has the biggest market share of any open source database. Almost any web-hosting company can provide.
- After setting database and access to it, managing the database is straightforward. Initial access needs to be configured by a database administrator. Tools such as MySQL Administrator or phpMyAdmin help to manage database.
- SQL has always been relatively fast, much due to its simplicity. In the last few years, MySQL has gained foothold in the enterprise market due to new “enterprise class” features and general maturity without compromising performance for simple usage.
- MySQL is the most popular Open Source Relational SQL Database Management System. MySQL is one of the best RDBMS being used for developing various web-based software applications. MySQL

is developed, marketed and supported by MySQL AB, which is a Swedish company. This tutorial will give you a quick start to MySQL and make you comfortable with MySQL programming.

Article IV. Flask

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

What is Web Framework?

Web Application Framework or simply Web Framework represents a collection of libraries and modules that enables a web application developer to write applications without having to bother about low-level details such as protocols, thread management etc.

What is Flask?

Flask is a web application framework written in Python. It is developed by Armin Ronacher, who leads an international group of Python enthusiasts named Pocco. Flask is based on the Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

A **Flask** application is started by calling the **run()** method. However, while the application is under development, it should be restarted manually for each change in the code. To avoid this inconvenience, enable **debug support**. The server will then reload itself if the code changes. It will also provide a useful debugger to track the errors if any, in the application.

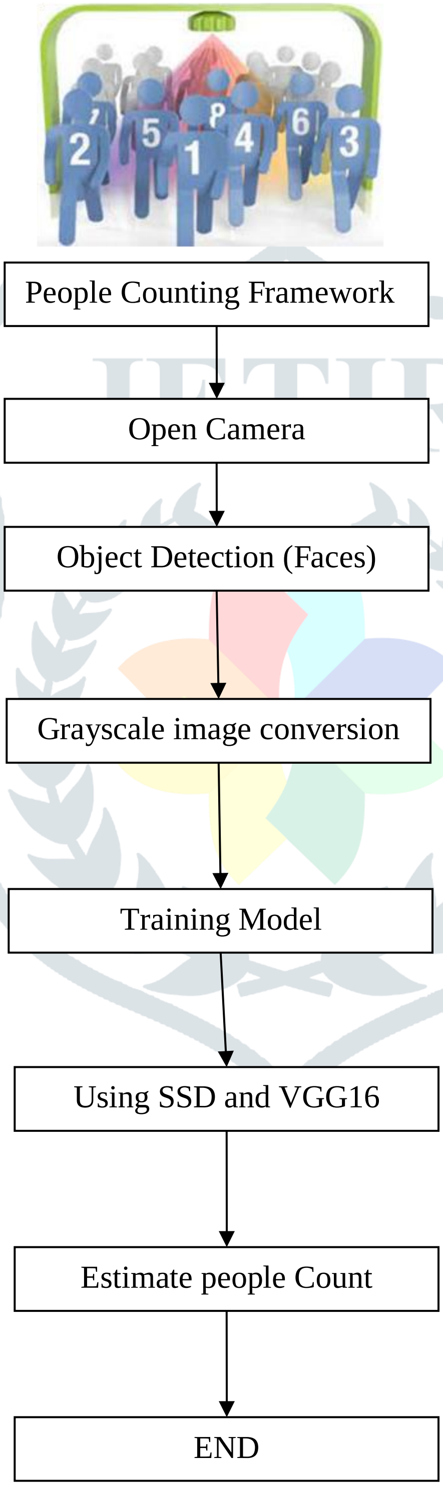
The **Debug** mode is enabled by setting the **debug** property of the **application** object to **True** before running or passing the debug parameter to the **run()** method.

```
app.debug = True

app.run()

app.run(debug = True)
```

Flow chart diagram



CONCLUSION

This research aims to focus on the People Counting system which is explored by the deep learning SSD model. The single shot detector (SSD) was proposed to produce better results as compare to YOLO model. The YOLO model has low recall and more localization error when compared to SSD. In this model, VGG16 convolutional neural network was replaced. The main motive is to replace the VGG16 convolutional neural network model which is thirty time larger in size as compare to mobile net SSD. The SSD is not compactable for our model requirement for better extracting features and non-maximum suppression selects. This model is obtained for experimental results and the model gives good accuracy TPR of 95.06% with FPR of 0.08%. The results of output can be further improved, by training the data set which is different and in different place. Sometimes this method do not produce high accuracy due to high traffic counting, clothes of person, shadow issue, and so on.

In the future, a different kind of detecting algorithm along with top-down view dataset and the same algorithm can be implemented on crowd detection, traffic counting and vehicle detection, etc. The major limitation of SSD has little depth layer in a neural network that do not produce high level prediction for small object. Therefore, SSD algorithm is not compatible for small object detection. In the future work, our model can be trained with complex dataset augmentation or a large amount of data for training.

SCOPE OF FUTURE DEVELOPMENT

In future with this technology, we can easily identify the person’s gender with machine learning and deep learning concept, it will be more useful to identify the crossing people in places . Detecting human beings accurately in a visual surveillance system is crucial for diverse application areas including abnormal event detection human gait characterization congestion analysis, person identification, gender classification and fall elderly people.

BIBILOGRAPHY

- Python Essential Reference: Developer's Library Paperback – 9 Jul 2009 by David Beazley (Author)

- Python Programming: An Introduction to Computer Science 2nd Edition by John Zelle (Author)

- Automate the Boring Stuff with Python: Practical Programming for Total Beginners 1st Edition by Al Sweigart (Author)

- Python Programming for the Absolute Beginner, 3rd Edition 3rd Edition by Michael Dawson (Author)

- Python Programming for Beginners: An Introduction to the Python Computer Language and Computer Programming 1st Edition by Jason Cannon (Author)

- Python: Python Programming For Beginners - The Comprehensive Guide To Python Programming: Computer Programming, Computer Language, Computer Science Paperback – August 9, 2016 by Adam Stark (Author)

WEBSITES

https://www.w3schools.com/python/

https://en.wikipedia.org/wiki/Python_(programming_language)

https://www.codecademy.com/learn/learn-python

https://www.learnpython.org/

REFERENCES

[1] Paresh M Tank, Hitesh A Patel. Survey on Human Detection Techniquesin Real Time Video. IJCSI International Journal of Computer Science Issues Vol. 7, Issue 5, May 2018.

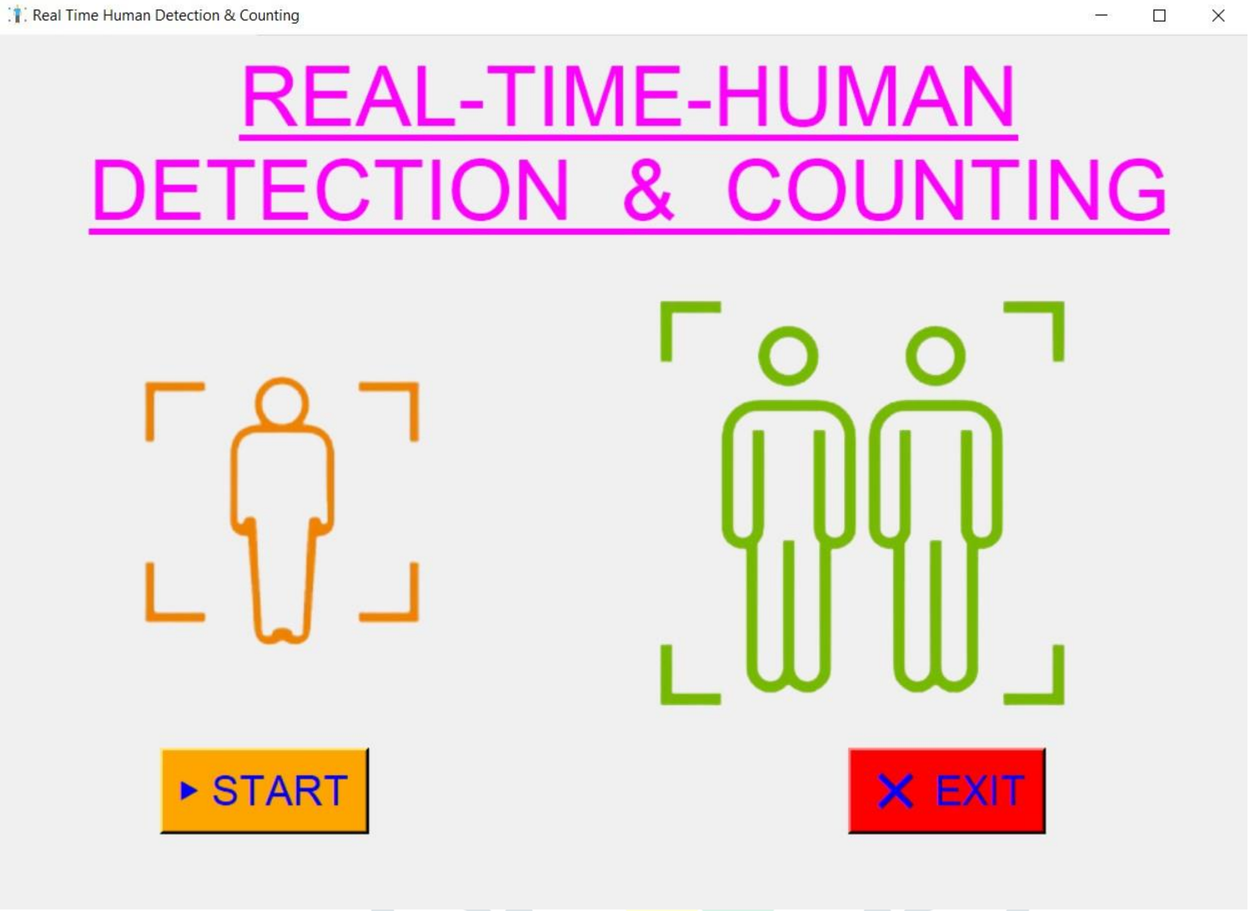
[2] N. Dalal and B. Triggs, “Human detection using oriented histograms of flow and appearance,” in European Conference on Computer Vision (ECCV), 2006, pp. 428–441 .

[3] Mohamed E Hussein, W. Abd-Almageed, Yang Ran, Larry S. Davis. Real-Time Human Detection, Tracking, and Verification in Uncontrolled Camera Motion Environments, 10.1109/ICVS.2006.52 Computer Vision Systems, 2006 ICVS '06. IEEE International Conference

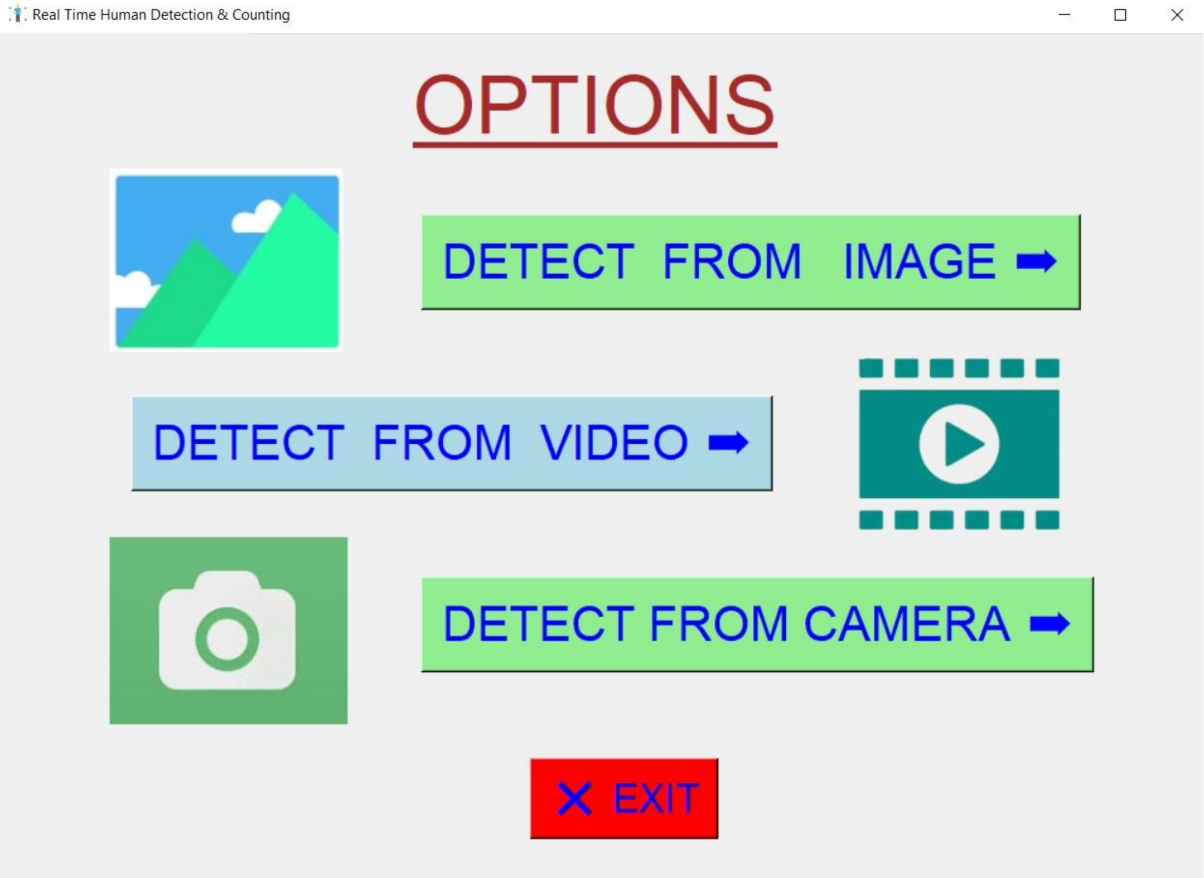
[4] Manoranjan Paul, Shah M E Haque, and Subrata Chakraborty 2013 Human detection in surveillance videos and its applications, EURASIP Journal on Advances in Signal Processing 2013(1):25 .DOI: 10.1186/1687-6180-2013-176

Screenshot:

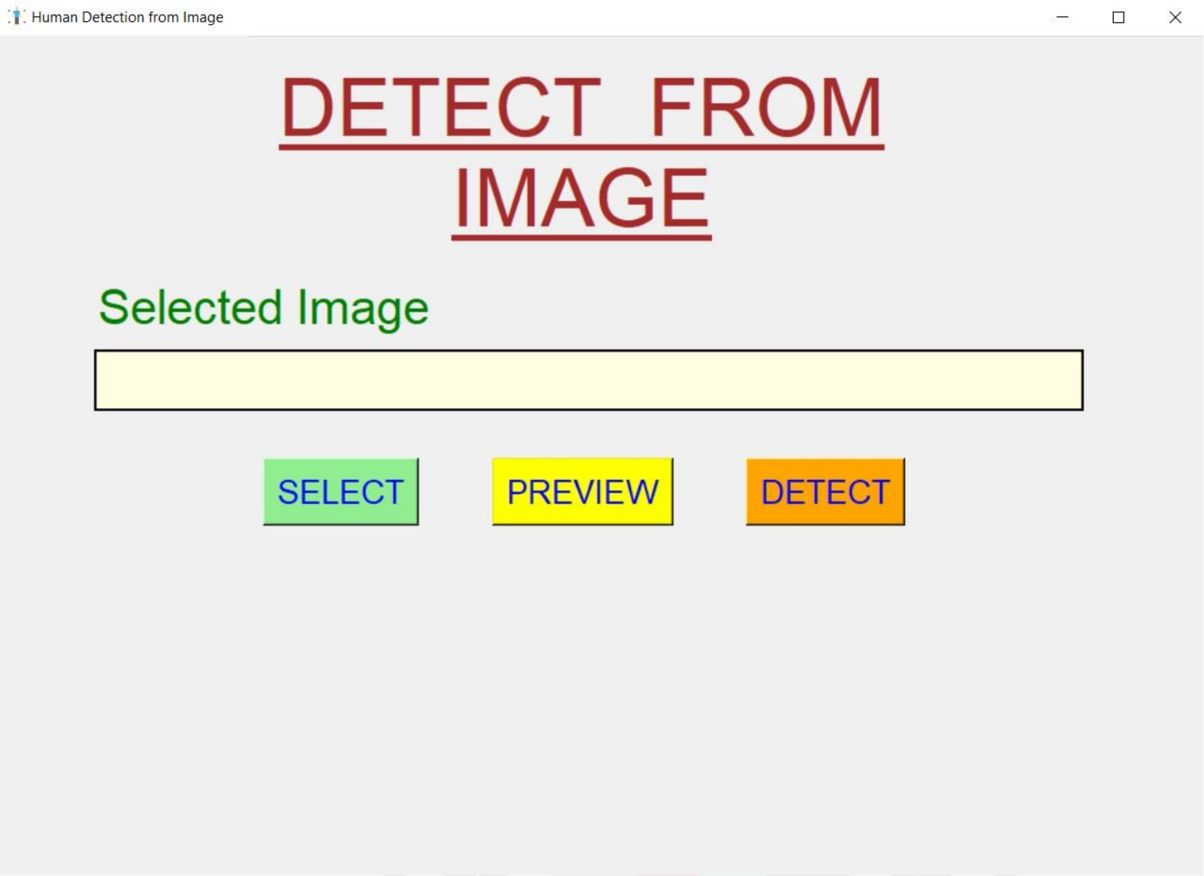
DETECTION PROCESS:



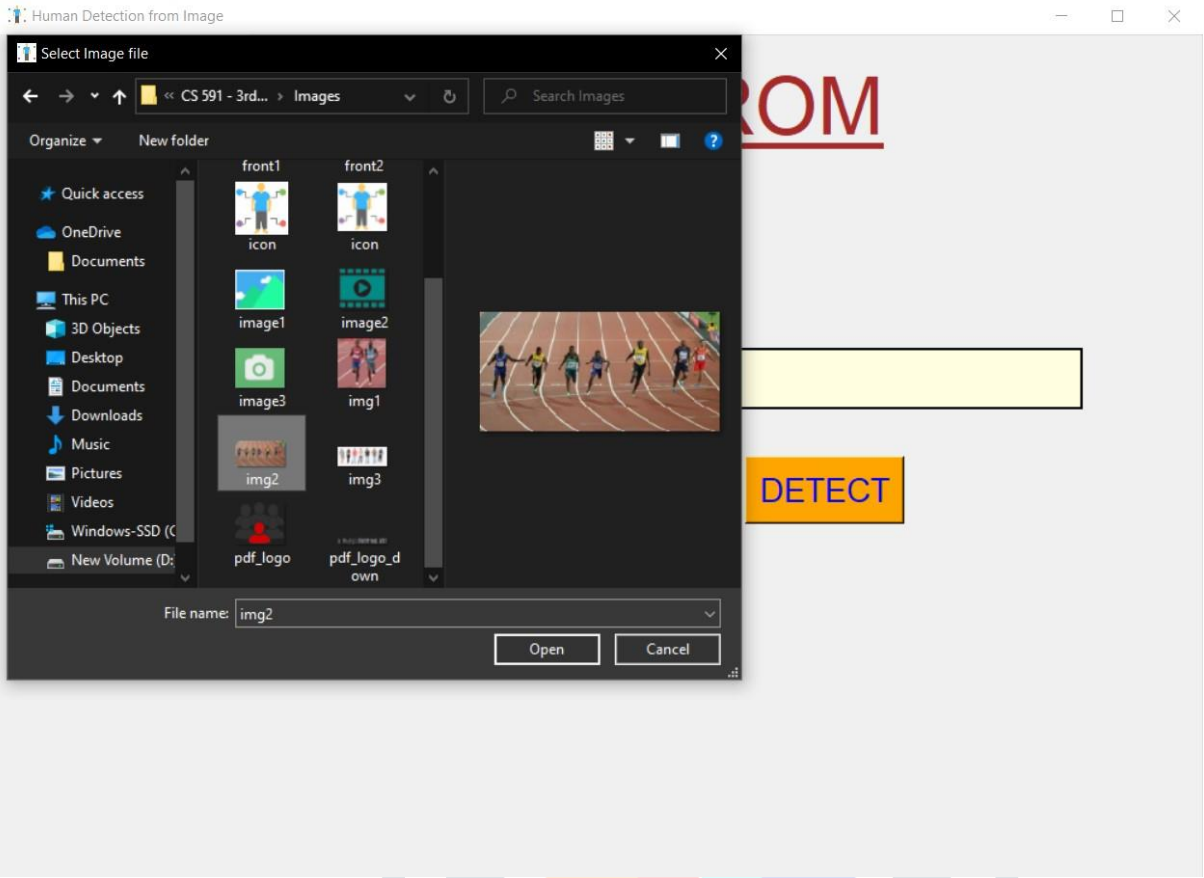
OPTIONS


Real Time Human Detection & Counting
OPTIONS
DETECT FROM IMAGE ➡
DETECT FROM VIDEO ➡
DETECT FROM CAMERA ➡
✕ EXIT

SELECTING PROCESS:

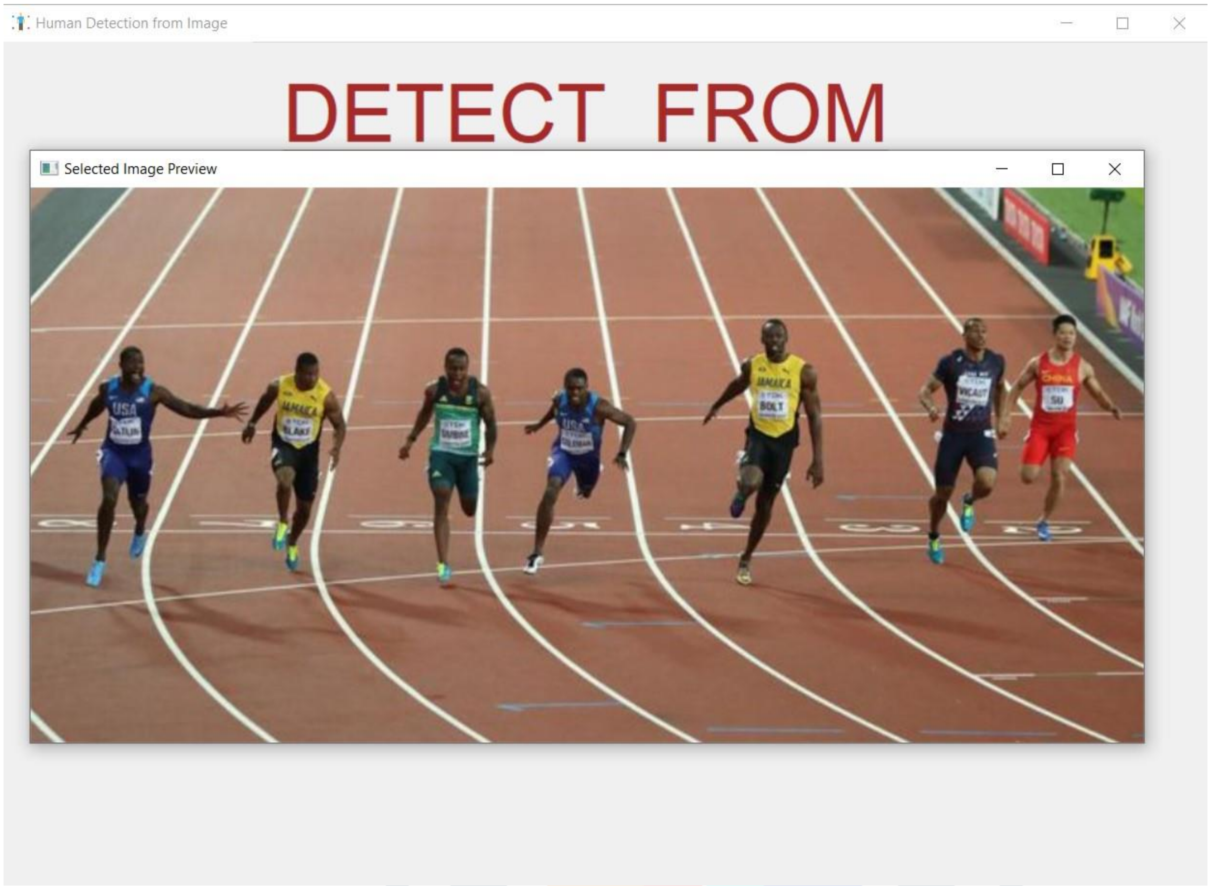


SELECT IMAGE FILE:

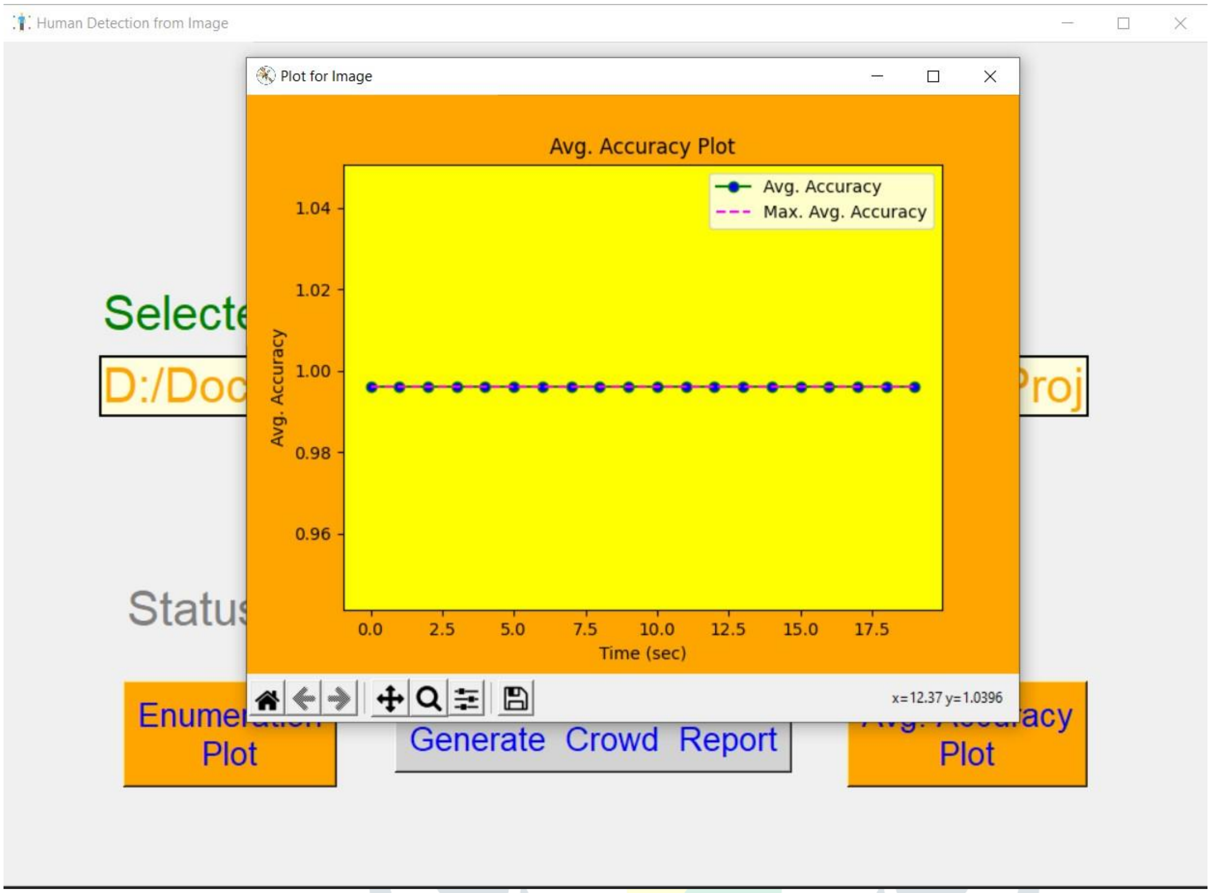


File name: img2

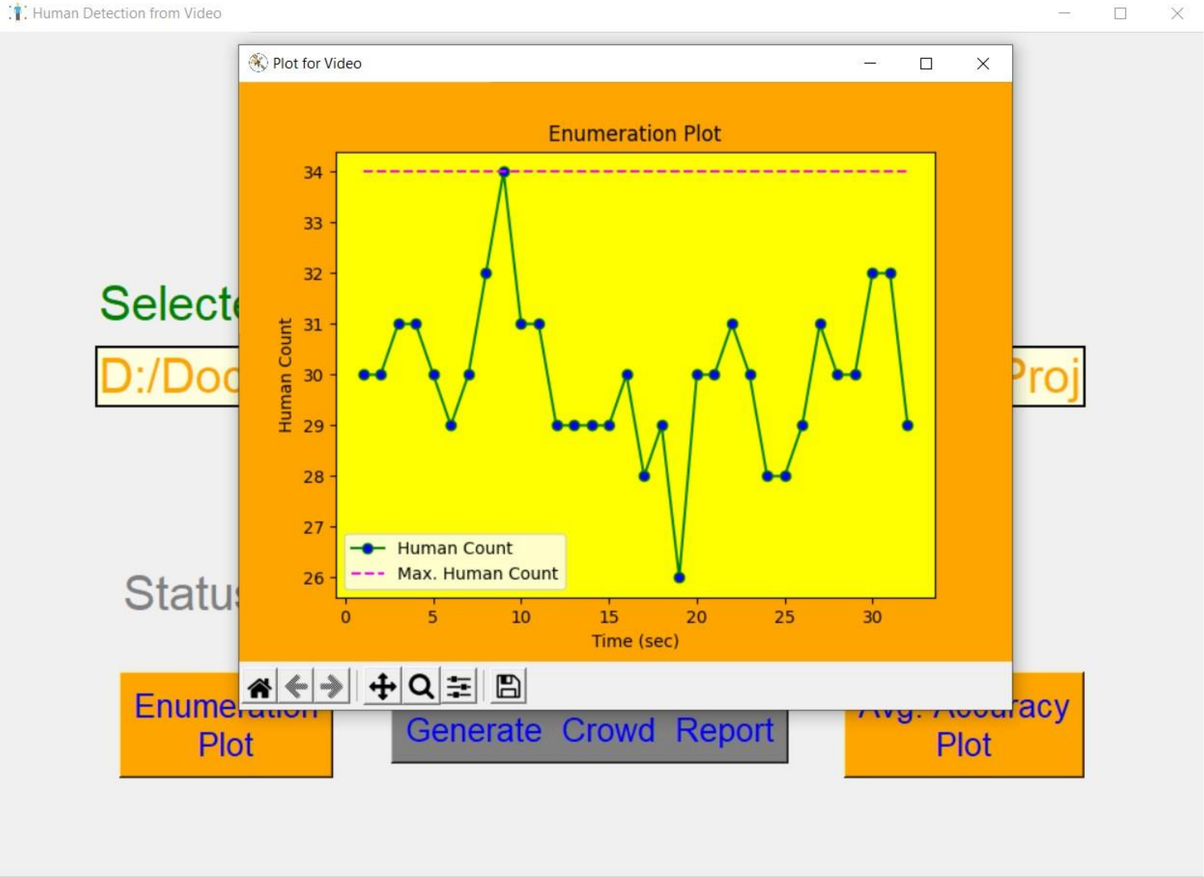
DETECTION:


Human Detection from Image
DETECT FROM
Selected Image Preview

ACCURACY PLOT:



ENUMERATION PLOT:


Human Detection from Video
Plot for Video
Enumeration Plot
Human Count
Time (sec)
34
33
32
31
30
29
28
27
26
0
5
10
15
20
25
30
Human Count
Max. Human Count
Selecte
D:/Doc
Proj
Status
Enumeration Plot
Generate Crowd Report
Avg. Accuracy Plot