**Multi-Task Cascaded Convolutional Network based Intelligent Fruit Detection for Designing Automated Robot**

**Multi-Task Cascaded Convolutional Network based Intelligent Fruit Detection for Designing Automated Robot**

A Project Report submitted in partial fulfillment of the degree of the   
Bachelor of Technology in Computer Science and Engineering

By

**T.Srividhya 19C41A05E9(srividyasrividya303@gmail.com)**

**A.Manohitha 19C41A05G6(manohithaanna@gmail)**

**P.Nandini 19C41A05F1(penukulanandini@gmail.com)**

**K.Vamshidhar 19C41A05F5(vamshikasarala143@gmail.com)**

Under the Guidance of

**Mr R.Bharath Kumar**



**Department of Computer Science and Engineering**

**Jayamukhi institute of technological Sciences**

**Narsampet, warangal-506 332**

**(Affiliated to JNTUH, Accredited by NAAC ‘A’ Grade)**

**2022-2023**

**Jayamukhi institute of technological Sciences**

**Narsampet, warangal – 506 332**

**(Affiliated to JNTUH, Accredited by NAAC ‘A’ Grade)**

**CERTIFICATE**



This is to certify that the Project Report entitled “**Multi-Task Cascaded Convolutional Network based Intelligent Fruit Detection for Designing Automated Robot”** is a bonafide work of the students by **T.Srividya (19C41A05E9), A.Manohitha(19C41A05G6), P.Nandini(19C41A05F1), K.Vamshidhar(19C41A05F5)** submitted in partial fulfillment of the requirements for the award of the degree of Bachelor Technology in Computer Science & Engineering during the academic year 2022-2023.

**Guide Head of the Department**

**Principal**

**ACKNOWLEDGEMENT**

It gives us immense pleasure in expressing sincere and deep sense of gratitude to **Dr.V.Janaki Principal.** Jayamukhi Institute of Technological sciences for the facility made available for the progress and completion of my main project.

We are extremely thankful to **Dr.P.Srinivas Rao, Head of the Department** Computer Science and Engineering for the permission and encouragement accorded to carry out this major Project work.

We are extremely thankful to **Mr.R.Bharath Kumar** Guide Computer Science and Engineering for his enthusiastic and innovative guidance and the permission, encouragement accorded to carry out this project work.

**T.Srividya(19C41A05E9)**

**A.Manohitha(19C41A05G6)**

**P.Nandini(19C41A05F1)**

**K.Vamshidhar(19C41A05F5)**

**ABSTRACT**

Effective and efficient fruit detection is considered crucial for designing automated robot (AuRo) for yield estimation, disease control, harvesting, sorting and grading. Several fruit detection schemes for designing AuRo have been developed during the last decades. However, conventional fruit detection methods are deficient in real-time response, accuracy and extensibility. This paper proposes an improved multi-task cascaded convolutional network (MTCNN) based intelligent fruit detection (InFD) method. This method has the capability to make the AuRo work in real-time and with high accuracy. Moreover, based on the relationship between the diversity samples of dataset and the parameters of neural networks evolution, this work presents an improved augmented method. A procedure that is based on image fusion to improve the detector performance. The experiment results demonstrated that the proposed detector performed immaculately, both in terms of accuracy and time-cost. Furthermore, the extensive experiment also demonstrated that the proposed technique has the capacity and a good portability to work with other akin objects conveniently.

**CONTENTS**

**1.INTRODUCTION**

**2.LITERATURE SURVEY**

**3.SYSTEM DESIGN**

3.1 SYSTEM ARCHITECTURE

3.2 UML DIAGRAMS

3.3 USE - CASE DIAGRAMS

3.4 CLASS DIAGRAMS

3.5 SEQUENCE DIAGRAMS

**4. MODULES USED IN PROJECT**

5**. SYSTEM REQUIREMENTS**

5.1 HARDWARE REQUIREMENTS

5.2 SOFTWARE REQUIREMENTS

**6. SYSTEM STUDY**

6.1 FEASIBILITY STUDY

6.2 ECONOMIC FEASIBILITY

6.3 TECHNICAL FEASIBILITY

6.4 SOCIAL FEASIBILITY

**7. SOFTWARE ENVIRONMENT**

**8. SYSTEM TEST**

**9. RESULTS**

**10. CONCLUSION**

**11.REFERENCES**

**Introduction**

Fruit detection for yield estimation, grade sorting, disease control and other applications in agricultural field have achieve intensive popularity over the past few decades [1]–[5]. Several systems have been deployed for automated harvesting robots, which have led to considerable improvement in the industry [6], [7]. Particularly, recognizing and classifying fruits according to their quality has been one of the most popular research fields attracting most of the farm enterprises. Fruit detection is undoubtedly the first and foremost parameter to be considered in order carry out more in-depth studies on the subject. Therefore, many researchers have made efforts for years to develop robust algorithms for fruit detection [8]–[10]. Although, the performance of fruit detection systems has been improved remarkably, they are still far from practical application. The basic difficulties in developing such fruit detection system are the uncertain and unrestrained environments of orchards. These include numerous challenging tasks, such as insufficient or over illumination, indistinguishable backgrounds, heavy occlusion by neighborhood fruits or foliage, low-resolutions, variation of pose and so on.

Fruit detection can be considered a special type of object detection that has many similarities with face detection task [11]–[13]. Due to the advantage of high precision, cascaded convolutional networks (CCN) based face detection has acquired a remarkable breakthrough [14], [15]. Among these state-of-the-art methods, multi-task cascaded convolutional network (MTCNN) [16] is the most popular one due to its outstanding performance in accuracy and time-consumption. Although MTCNN has achieved great progress in face detection task, deploying this method directly for fruit detection task is not suitable. It is due to the design of MTCNN, that its architecture includes many specificity functions for face detection, which are not suitable for the task of fruit detection. Thus, there is a need to improve this MTCNN framework by removing customized functionality

The absence of a unified benchmark is another great challenge for fruit detection. A sufficient amount of sample images plays an important role in deep learning based model training. In this research, we collected images from apple orchard by digital camera. Then we selected the suitable ones and labeled them to create a dataset. Creating a dataset manually is a tedious and time-consuming task. So we devised a new augmented method based on fusion algorithm. The motivation for this fusion method came from the principle that the generated new samples should be close to authentic images. Supplementary samples were created for diversity by adding fusion images that would help improve the final result of this detector. In order to evaluate the structure whether it could be applied to other kinds of objects conveniently, we trained the detector on two other fruits species (strawberry and orange) as well.

To summarize, our contributions are as follows:

1. We proposed a new architecture for fruit detection called Fruit-MTCNN (F-MTCNN) by improving the baseline model of MTCNN. And this detector has the attributes of high accuracy and less time-consumption.
2. We proposed a novel augmented method called fusion augmentation (FA). We generate artificial images samples by adding negative patches from samples of dataset by random cropping that supplement the samples diversity.
3. The proposed approach can be deployed to other kinds of objects conveniently with a small amount of training samples.

The organization of the rest paper is arranged as follows. In Section II, we review prior related work in fruit detection. Section III, IV we introduce method used in this study. Our experiments in this research are shows in Section V. In Section VI, we analyze and discuss our results and present the conclusion of this work.

**LITERATURE SURVEY**

L[1]: Deep learning implementation using convolutional neural network in mangosteen surface defect detection.

Authors: [Laila Ma'rifatul Azizah](https://ieeexplore.ieee.org/author/37086284454), Sitti Fadillah Umayah, Slamet Riyadi, Cahya Damarjat, Nafi Ananda Utama

Description: Mangosteen is one of the fruits that has an enormous export potential in Indonesia. However, not all mangosteen is the defect free fruit. The quality assurance in mangosteen export is done manually by sorting expert. Therefore, this can lead unstandardized and inaccurate results. The result happens because of human error. It needs an image processing technology to help the sorting process which one is the defect and non-defect. In this study, we use one of deep learning architecture that is Convolutional Neural Network (CNN). Therefore, we use CCN as a detection of mangosteen. CNN proved to be very efficient regarding classifying images. This CNN method is implemented using 4-folds Validation Cross to validate data accuracy. In the preparation of the CNN architecture model, initializing the parameter configuration accelerates the network training process. The results of the experiments using CNN algorithm showed the performance of defect detection on the mangosteen fruit of 97%.

L[2] :Grading of ripening stages of red banana using dielectric properties changes and image processing approach

Authors: Asutosh Mohapatra , S. Shanmugasundaram,  R. Malmathanraj

Description : In this research work, the dielectric properties of red banana fruit are studied at different ripening temperatures for developing a rapid and non-destructive assessment method to measure the ripening stages of red banana. A 5 volt sine wave AC power supply and a rectangular parallel plate capacitor circuit are used to measure the difference in dielectric properties caused by the introduction of a red banana in between the plates. The values of properties like capacitance and relative permittivity are increased continuously whereas impedance and admittance are decreased gradually with increase in ripening stages of red banana. In image processing approach, Noise Reductant Local Binary Pattern (NRLBP), Local Binary Pattern (LBP), Completed Local Binary Pattern (CLBP) based techniques are used for red banana’s ripening grade determination. The processing stages involved are enhancement, Binary Pattern generation and classification. The variant Binary patterns are tested on noisy as well as noiseless condition and the results are compared. A novel enhancement technique for banana ripening grade determination is proposed based on segmentation using Tsallis entropy. Also novel idea on the automation of q parameter involved in Tsallis Entropy is implemented. The threshold parameter of the Noise Reductant Local Binary Pattern (NRLBP) varied and its effect on classification rate is studied. A new modification is proposed and implemented on NRLBP to accommodate uniform background and areas with the image. Classification is done using Chi-Square distance/nearest neighbor and Fuzzy C means (FCM) clustering. The results are compared and superiority of FCM method for banana ripening grade determination is noted.

L[3]: An in-field automatic wheat disease diagnosis system .

Authors: Jiang Lu, Jie Hu, Guannan Zhao, Fenghua Mei, Changshui Zhang

Description : Crop diseases are responsible for the major production reduction and economic losses in agricultural industry worldwide. Monitoring for health status of crops is critical to control the spread of diseases and implement effective management. This paper presents an in-field automatic wheat disease diagnosis system based on a weakly supervised deep learning framework, i.e. deep multiple instance learning, which achieves an integration of identification for wheat diseases and localization for disease areas with only image-level annotation for training images in wild conditions. Furthermore, a new in-field image dataset for wheat disease, Wheat Disease Database 2017 (WDD2017), is collected to verify the effectiveness of our system. Under two different architectures, i.e. VGG-FCN-VD16 and VGG-FCN-S, our system achieves the mean recognition accuracies of 97.95% and 95.12% respectively over 5-fold cross-validation on WDD2017, exceeding the results of 93.27% and 73.00% by two conventional CNN frameworks, i.e. VGG-CNN-VD16 and VGG-CNN-S. Experimental results demonstrate that the proposed system outperforms conventional CNN architectures on recognition accuracy under the same amount of parameters, meanwhile maintaining accurate localization for corresponding disease areas. Moreover, the proposed system has been packed into a real-time mobile app to provide support for agricultural disease diagnosis.

L[4]: A segmentation method for greenhouse vegetable foliar disease spots images using color information and region growing.

Authors: Juncheng Ma, Keming Du, Lingxian Zhang, Feixiang Zheng, Jinxiang Chu, Zhongfu Sun

Description: This paper presents a novel image processing method using color information and region growing for segmenting greenhouse vegetable foliar disease spots images captured under real field conditions. Disease images captured under real field conditions are suffering from uneven illumination and complicated background, which is a big challenge to achieve robust disease spots segmentation. A disease spots segmentation method consisting of two pipelined procedures is proposed in this paper. Firstly a comprehensive color feature and its detection method are presented. The comprehensive color feature (CCF) consists of three color components, Excess Red Index (ExR), H component of HSV color space and b∗ component of L∗a∗b∗ color space, which implements powerful discrimination of disease spots and clutter background. Then an interactive region growing method based on the CCF map is used to achieve disease spots segmentation from clutter background. To evaluate the robustness and accuracy, the proposed segmentation method is assessed by cucumber downy mildew images. Results show that the proposed method can achieve accurate and robust segmentation under real field conditions.

L[5]: A robust algorithm based on color features for grape cluster segmentation.

Authors: Nasser Behroozi-Khazaei, Mohammad Reza Maleki.

Description: Image processing has been widely used for automation purposes in modern agriculture. The algorithm development for the image segmentation is the most controversial and challenging issue in orchard environment which researchers encounter. This paper describes a robust algorithm based on artificial neural network (ANN) and genetic algorithm (GA) for segmenting grape clusters from leaves and background using color features near to harvest. GA was employed for optimizing of ANN structure and selecting supreme color features simultaneously. The results showed that GA specifies the 8 color features as supreme features and define 8–15-35–3 as the best structure of the ANN. The overall accuracy of the algorithm was 99.40%. The promising results in algorithm development described in this study lead to introduce it as a practical sensing tool in precision agriculture as well as those industrial facilities dealing with image analysis.

L[6]: Apple Location Method for the Apple Harvesting Robot

Authors: Wenhua Mao, Baoping Ji, Jicheng Zhan, Xiaochao Zhang, Xiaoan Hu.

Description: The apple harvesting robot should have two eyes to sensor the location of picking apples. The binocular machine vision system used two Canon digital cameras was built, instead of digital video used as usual. Thus, the digital camera vision system had higher resolution and superior performance than a digital video vision system, which could capture a Jpeg image with 3456\*2592 pixels, its field of view included the whole apple tree. For the Fuji apple tree, it is an obvious color difference with ripe apples and their surroundings of leaves and braches. Therefore, the Drg-Drb color index was used to segment apples from their surroundings. Then the mistaken classified background regions was deleted by the area filter, and he picking apples were similarly chosen by area parameter. After that, the conglutinated apples were segmented by the bidirectional scanning line algorithm, which were scanned from the horizontal and vertical direction. Finally, all of picking apples were positioned by their circum-diameter matching algorithm. The experimental result showed that the correct classification rate of picking apple fruit achieved 90%.

L[7]: Real-time segmentation of strawberry flesh and calyx from images of singulated strawberries during postharvest processing.

Authors: A. Durand-Petiteville, S. Vougioukas, D.C. Slaughter

Description: This paper presents an image processing algorithm that automatically extracts the flesh and calyx areas from strawberry images. Images are captured by a camera included in a strawberry de-capping machine. Lighting is controlled and the background is known, conditions that are typical of postharvest processing. The goal is to extract as many flesh and calyx pixels as possible while rejecting any pixels belonging to the background. The proposed approach relies on image color segmentation in a two-dimensional color space, followed by a blob detection and selection stage. A set of 250 images is used to analyze the sensitivity of the algorithm with respect to user-defined parameters, and evaluate the performance of the approach. The algorithm appears to be easy to tune and allows accurate extraction of the areas of interest despite natural variation in strawberry shape and visual appearance. More than 98% of the flesh area was successfully extracted by the algorithm with less than 1% of the background pixels falsely included. Moreover, up to 79% of the calyx area could be extracted with less than 0.25% erroneous background pixels. Finally, the algorithm has been implemented using the C++ and Cuda languages and can be executed in real-time.

L[8]: Immature citrus fruit detection based on local binary pattern feature and hierarchical contour analysis.

Authors: Jun Lu, Won Suk Lee, Hao Gan, Xiuwen Hu

Description: Detecting immature fruit in groves provides a promising benefit for growers to plan application of nutrients and estimate their yield and profit prior to harvesting. The goal of this study was to develop a robust algorithm to detect and count immature citrus fruit in images of the tree canopy. Images were all taken in low natural light conditions with a flashlight, and the green component of the colour images was used for further analysis. Local intensity maxima were detected and local binary pattern (LBP) features around them were extracted as an input of an ensemble classifier-RUSBoost. The positive predictions were considered as candidates and the hierarchical contour maps around them were extracted and fitted with Circular Hough Transform. The fitted circles were predicted as fruit targets if its radius were in a predetermined range.

The algorithm was evaluated with a test set of 25 images, achieved 80.4% true positive rate and 82.3% precision rate, and F-measure was 81.3%. The good performance of occlusion tolerance of the proposed method was mainly coming from the robust LBP texture descriptor and hierarchical contour analysis (HCA) which used the pattern of light intensity distribution on fruit surface. This study proposed an innovative method to detect green fruit in images of trees only by using texture and intensity distribution.

L[9]: Improved Cross-Label Suppression Dictionary Learning for Face Recognition

Authors: Tian Zhou, Sujuan Yang, Lei Wang, Jiming Yao, Guan Gui

Description: Cross-label suppression dictionary learning is an effective approach to preserve the label property for signal representation in face recognition. This paper presents a proposed improved dictionary learning algorithm, considering the tradeoffs between the operating time and the signal reconstruction residuals for the face recognition problem that combines an optimal loss function and the cross-label suppression supervised dictionary learning approach. Based on the relationship of the cost time of the dictionary learning algorithm and the residuals of the sparse representations, this paper attempts to select an optimal sparse coding dimension for the original signal to reduce the computational cost. Experiments on face recognition confirm that our proposed algorithm is able to achieve a desired classification results as well as obtain a considerably faster dictionary learning process.

**Existing System**

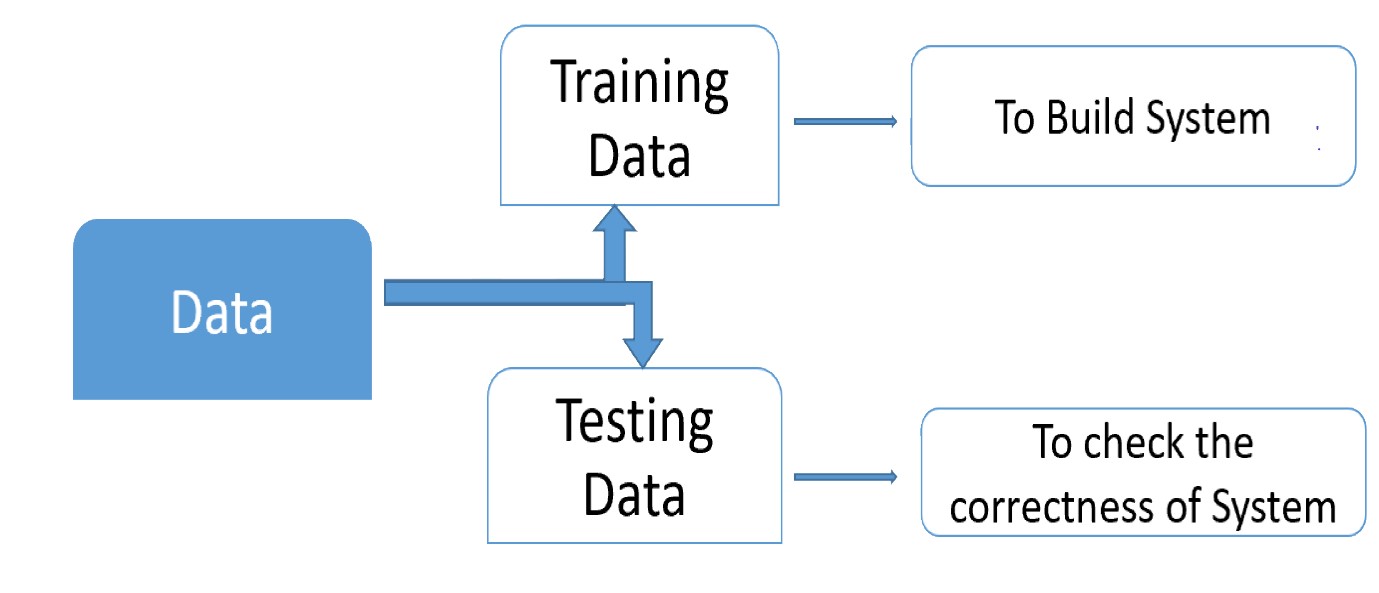
Automated harvesting robot is a potential solution for many challenges in agriculture such as the explosively increasing global old-age population, labor cost increase, increasing demand for of produce and so on. Identify and obtaining precise positions of fruits are the most important parts of the visual system for a harvesting robot. Due to thisreason, fruits identification and detection has been extensively studied for years. Generally, these methods can be divided into three types by the technologies they employ.

**Proposed System**

This proposed multi-modal approach provides better performance compare to prior work. However, using Faster RCNN architecture for fruit detection directly is inadequate. This is because the Faster-RCNN designed detection task for many categories of objects with large scale change. Whereas, the visual system in agriculture needs to detect one or only a few kinds of fruit in general, and usually the fruit size does not change significantly. Thus, the application of FasterRCNN model for fruit detection task is complicated and time-consuming. Furthermore, providing a large amount of data is necessary to prevent over-fitting problems, because the structure of Faster-RCNN is of a deeper architecture that contains thirteen convolution layers. During the recent years, due to the rapid development of security, intelligent equipment and other applications, the detection accuracy has been highly improved

**3. SYSTEM DESIGN**

**3.1 SYSTEM ARCHITECTURE:**

****

**3.2 UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

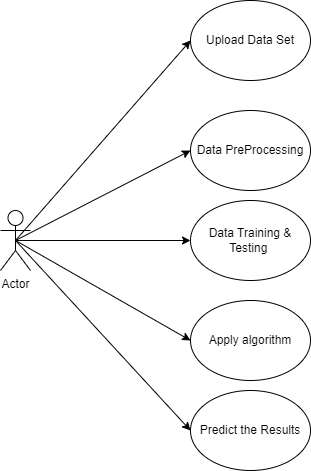
**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

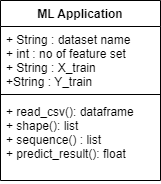
**3.3 USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted



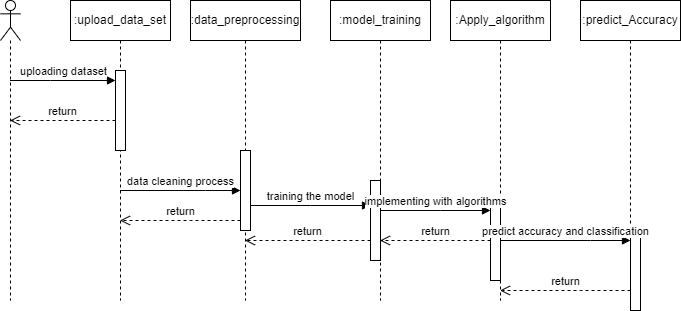
**3.4 CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



**3.5 SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



1. **Modules Used in Project**

**Tensorflow** TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software) and [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software library for dataflow and differentiable programming](https://en.wikipedia.org/wiki/Library_(computing)) across a range of tasks. It is a symbolic math library, and is also used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications such as [neural networks](https://en.wikipedia.org/wiki/Neural_networks). It is used for both research and production at [Google](https://en.wikipedia.org/wiki/Google).‍

TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain) team for internal Google use. It was released under the [Apache 2.0](https://en.wikipedia.org/wiki/Apache_License) [open-source license](https://en.wikipedia.org/wiki/Open-source_license) on November 9, 2015.

**Numpy**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

**Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots](https://matplotlib.org/tutorials/introductory/sample_plots.html) and [thumbnail gallery](https://matplotlib.org/gallery/index.html).

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

**Scikit – learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. Python is an interpreted high-level programming language for general-purpose programming. Created by uido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* 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 also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**5. SYSTEM REQUIREMENTS:**

**5.1 HARDWARE REQUIREMENTS:**

* System : Pentium IV 2.4 GHz.
* Hard Disk : 40 GB.
* Floppy Drive : 1.44 Mb.
* Monitor : 15 VGA Colour.
* Mouse : Logitech.
* Ram : 512 Mb.

**5.2 SOFTWARE REQUIREMENTS:**

* **Operating System:** Windows
* **Coding Language**: Python 3.7

**6. SYSTEM STUDY**

**6.1 FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**6.2 ECONOMICAL** **FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### 6.3 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**6.4 SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system

1. **SOFTWARE ENVIRONMENT**

# What is Python :-

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* + [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
  + GUI Applications (like Kivy, Tkinter, PyQt etc. )
  + Web frameworks like Django (used by YouTube, Instagram, Dropbox)
  + Image processing (like Opencv, Pillow)
  + Web scraping (like Scrapy, BeautifulSoup, Selenium)
  + Test frameworks
  + Multimedia

### Advantages of Python :-

Let’s see how Python dominates over other languages.

#### 1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

#### 2. Extensible

#### As we have seen earlier, Python can be**extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

#### 3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities**to our code in the other language.

#### 4. Improved Productivity

The language’s simplicity and extensive libraries render programmers**more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

#### 5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

#### 6. Simple and Easy

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and**code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

#### 7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

#### 8. Object-Oriented

This language supports both the **procedural and object-oriented**programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

#### 9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you[**download Python**](https://data-flair.training/blogs/install-python-windows/) for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

#### 10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to**code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

#### 11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

### Advantages of Python Over Other Languages

#### 1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

#### 2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

**The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.**

**3. Python is for Everyone**

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and [**machine learning**](https://data-flair.training/blogs/machine-learning-tutorials-home/), automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

### Disadvantages of Python

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

#### 1. Speed Limitations

We have seen that Python code is executed line by line. But since [Python](https://www.python.org/) is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

#### 2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

#### 3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can**raise run-time errors**.

#### 4. Underdeveloped Database Access Layers

#### Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

#### 5. Simple

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

**History of Python : -**What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time iproject at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

**What is Machine Learning : -**

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

**Categories Of Machine Leaning :-**

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

## Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

## Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

**Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

**Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

**Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

**No clear objective for formulating business problems** − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

**Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

## Applications of Machines Learning :-

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

* Emotion analysis
* Sentiment analysis
* Error detection and prevention
* Weather forecasting and prediction
* Stock market analysis and forecasting
* Speech synthesis
* Speech recognition
* Customer segmentation
* Object recognition
* Fraud detection
* Fraud prevention
* Recommendation of products to customer in online shopping

# How to Start Learning Machine Learning?

Arthur Samuel coined the term **“Machine Learning”** in 1959 and defined it as a **“Field of study that gives computers the capability to learn without being explicitly programmed”.**

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to [Indeed](http://blog.indeed.com/2019/03/14/best-jobs-2019/), Machine Learning Engineer Is The Best Job of 2019 with a 344% growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

### How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

### Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

#### (a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

#### (b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!  
Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

#### (c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is [Python](https://www.geeksforgeeks.org/python-programming-language/)! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as [Keras](https://keras.io/), [TensorFlow](https://www.tensorflow.org/), [Scikit-learn](https://scikit-learn.org/stable/), etc.

So if you want to learn ML, it’s best if you learn Python! You can do that using various online resources and courses such as [**Fork Python**](https://practice.geeksforgeeks.org/courses/fork-python) available Free on GeeksforGeeks.

### Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It’s best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

#### (a) Terminologies of Machine Learning

* **Model –**A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
* **Feature –**A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
* **Target (Label) –**A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
* **Training –**The idea is to give a set of inputs(features) and it’s expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
* **Prediction –**Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

#### (b) Types of Machine Learning

* **Supervised Learning –**This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
* **Unsupervised Learning –**This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
* **Semi-supervised Learning –**This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
* **Reinforcement Learning –**This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

### Advantages of Machine learning :-

#### 1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

#### 2. No human intervention needed (automation)

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

#### 3. Continuous Improvement

As [**ML algorithms**](https://data-flair.training/blogs/machine-learning-algorithms/) gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

#### 4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

#### 5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

### Disadvantages of Machine Learning :-

#### 1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

#### 2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

#### 3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

#### 4. High error-susceptibility

[MachineLearning](https://en.wikipedia.org/wiki/Machine_learning) is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

**Python Development Steps : -**

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list , dict, str and others. It was also object oriented and had a module system.  
Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

* Print is now a function
* Views and iterators instead of lists
* The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
* There is only one integer type left, i.e. int. long is int as well.
* The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
* Text Vs. Data Instead Of Unicode Vs. 8-bit

**Purpose :-** We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

**Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* 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 also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Install Python Step-by-Step in Windows and Mac :**

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

## How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

**Note:** The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. [Download the Python Cheatsheet here.](https://myelearninghub.com/python-cheat-sheet/)The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

### Download the Correct version into the system

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: [**https://www.python.org**](https://www.python.org)



Now, check for the latest and the correct version for your operating system.

**Step 2:** Click on the Download Tab.

****

**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

****

**Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.



• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

### Installation of Python

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.



**Step 2:** Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH



**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes.

### Verify the Python Installation

**Step 1:** Click on Start

**Step 2:** In the Windows Run Command, type “cmd”.



**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python –V** and press Enter.



**Step 5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

### Check how the Python IDLE works

**Step 1:** Click on Start

**Step 2:** In the Windows Run command, type “python idle”.



**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**Step 4:** To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



**Step 5:** Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

**Step 6:** Now for e.g. **enter print.**

1. **SYSTEM TEST**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### TYPES OF TESTS

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# **Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

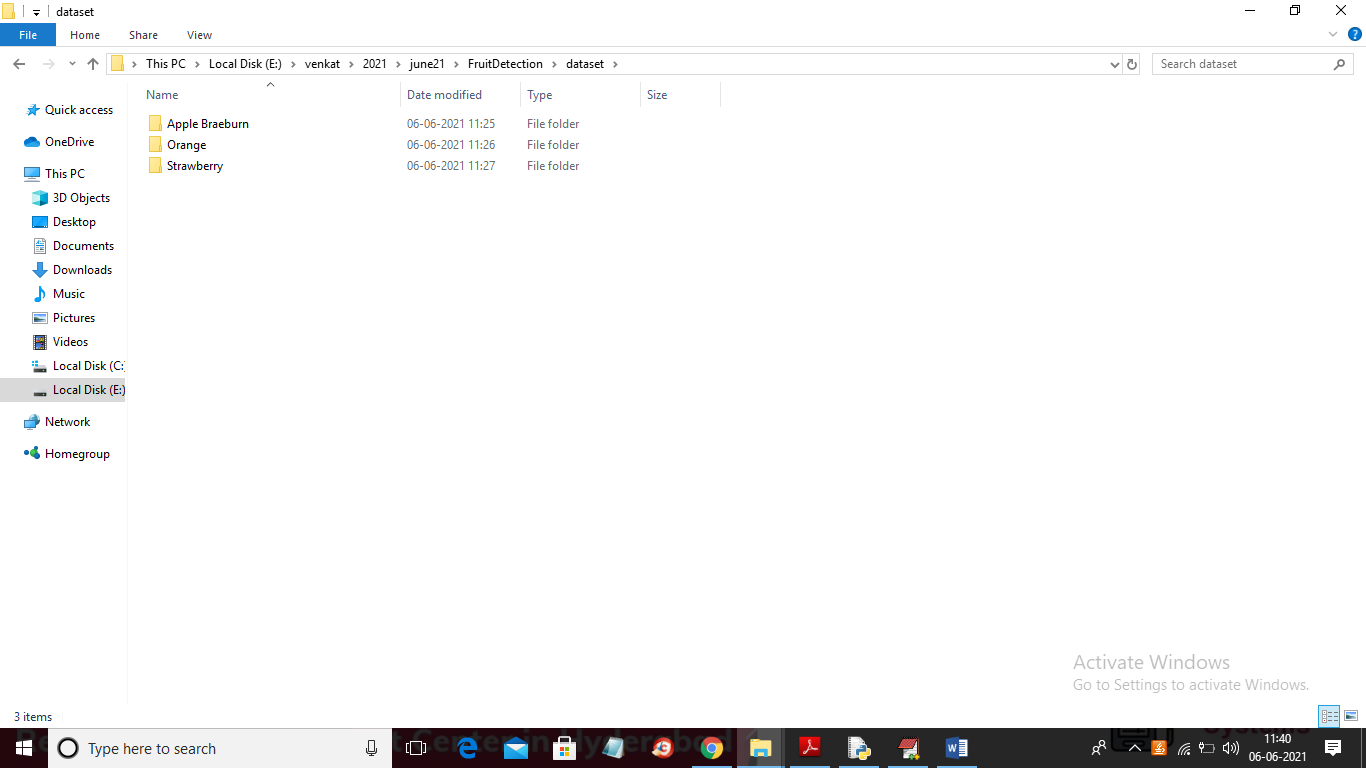
**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**9.RESULTS**

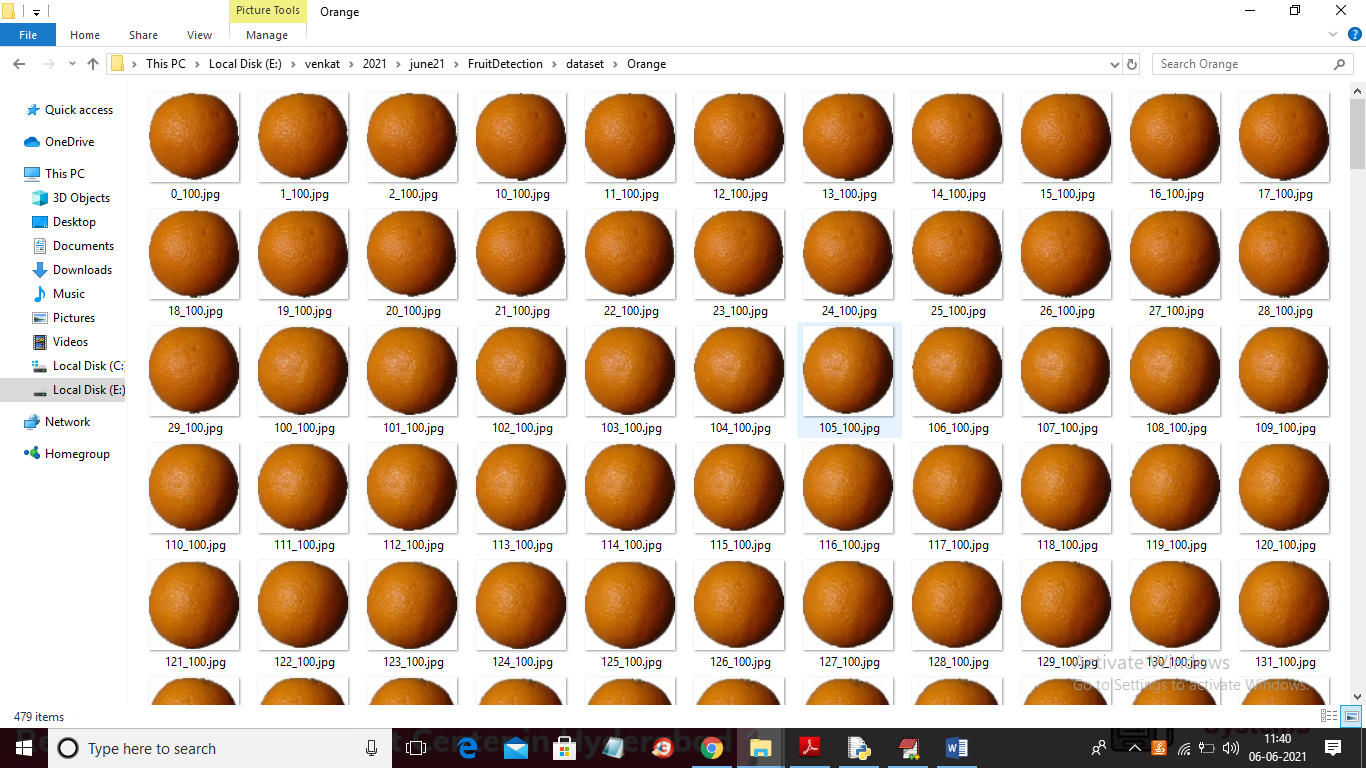
Snapshots of Multi-Task Cascaded Convolutional Networks based Intelligent Fruit Detection for Designing Automated Robot

In this paper author is designing Multi-Task Cascaded Convolution Neural Network to build fruit detection model as this network is good at face detection so author applying same MTCNN model to build fruit detection model. This model will accept tree images as input and then detect 3 different types of fruit such as Apple, Strawberry and Oranges. The author has used own fruit dataset which he has capture with his digital camera and he has not publish this dataset on internet so to build MTCNN model we have 360 degree fruit dataset from KAGGLE.

Below is the dataset use to build MTCNN model



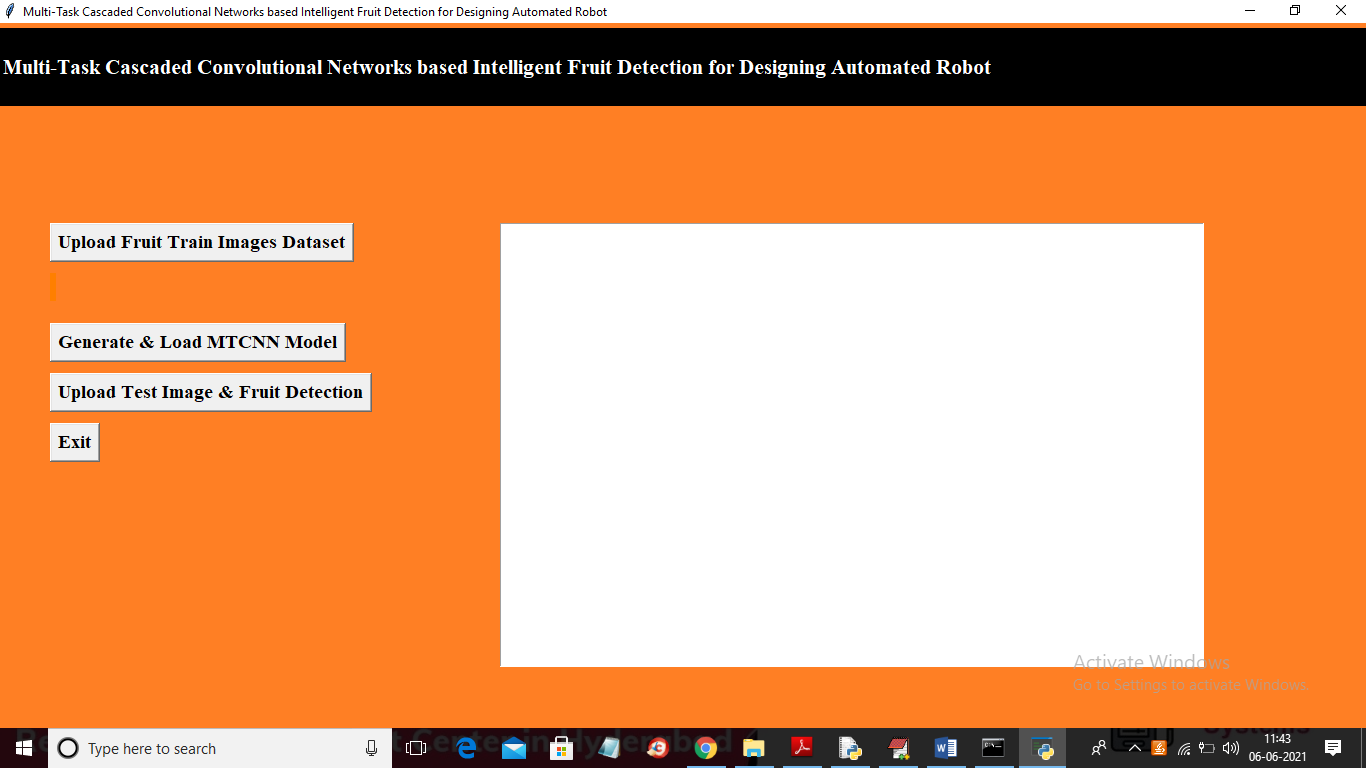
Go inside any fruit folder to see images of that fruit



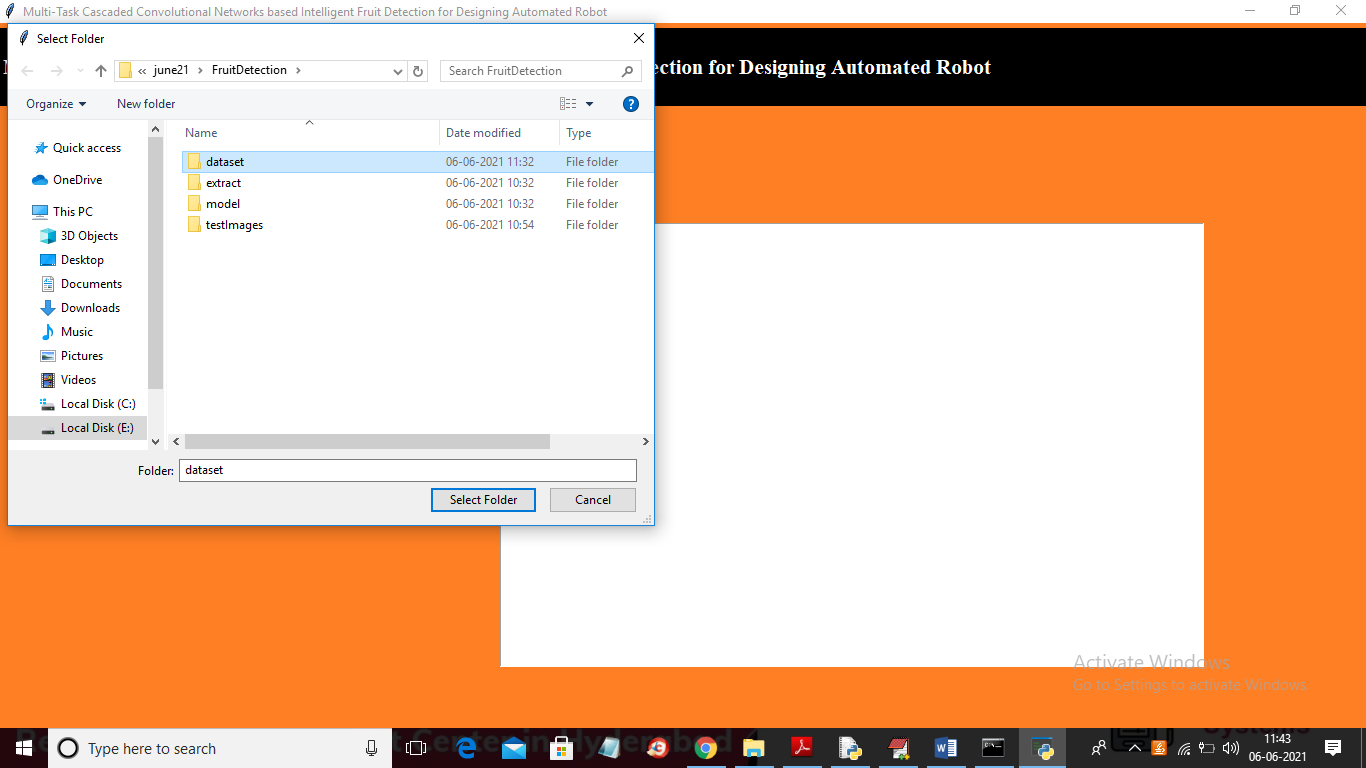
To build MTCNN model we have used image generator and augmentation technique from KERAS.

SCREEN SHOTS

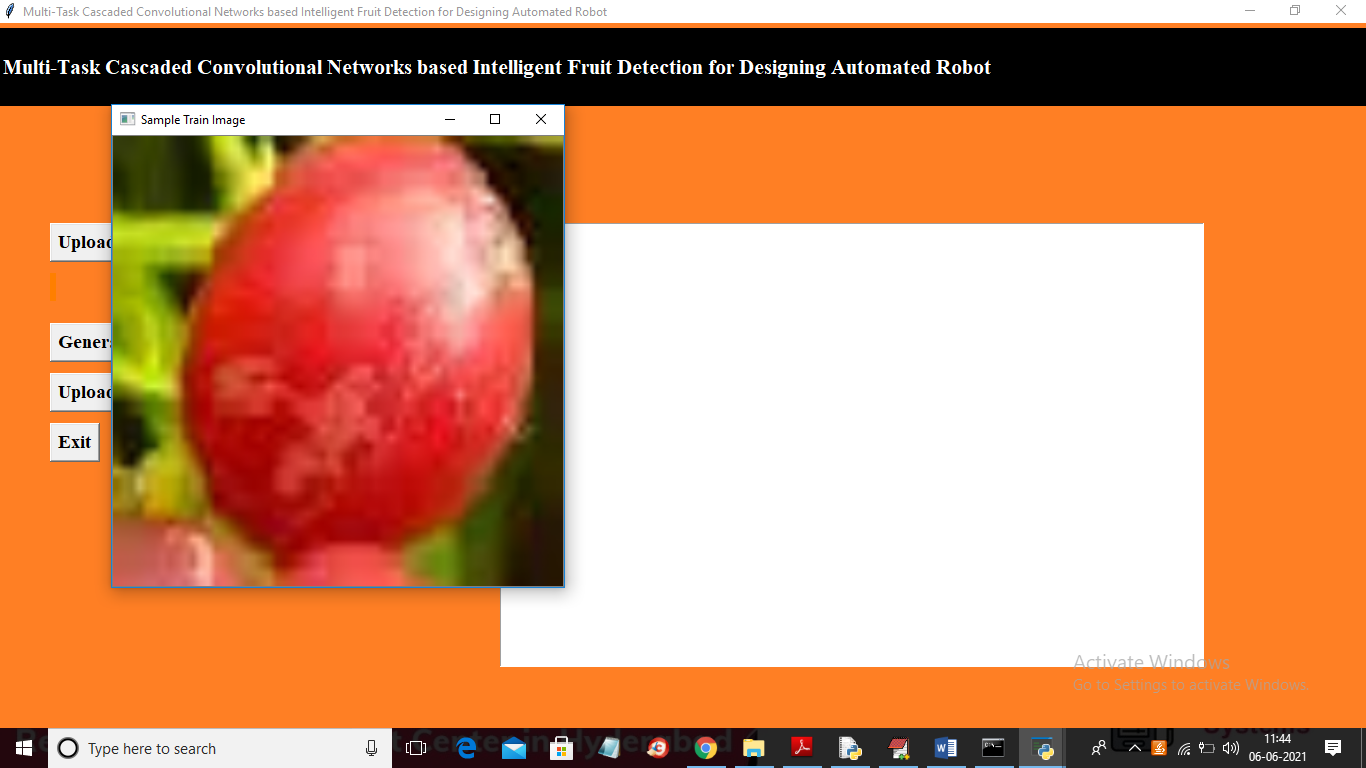
To run project double click on ‘run.bat’ file to get below screen



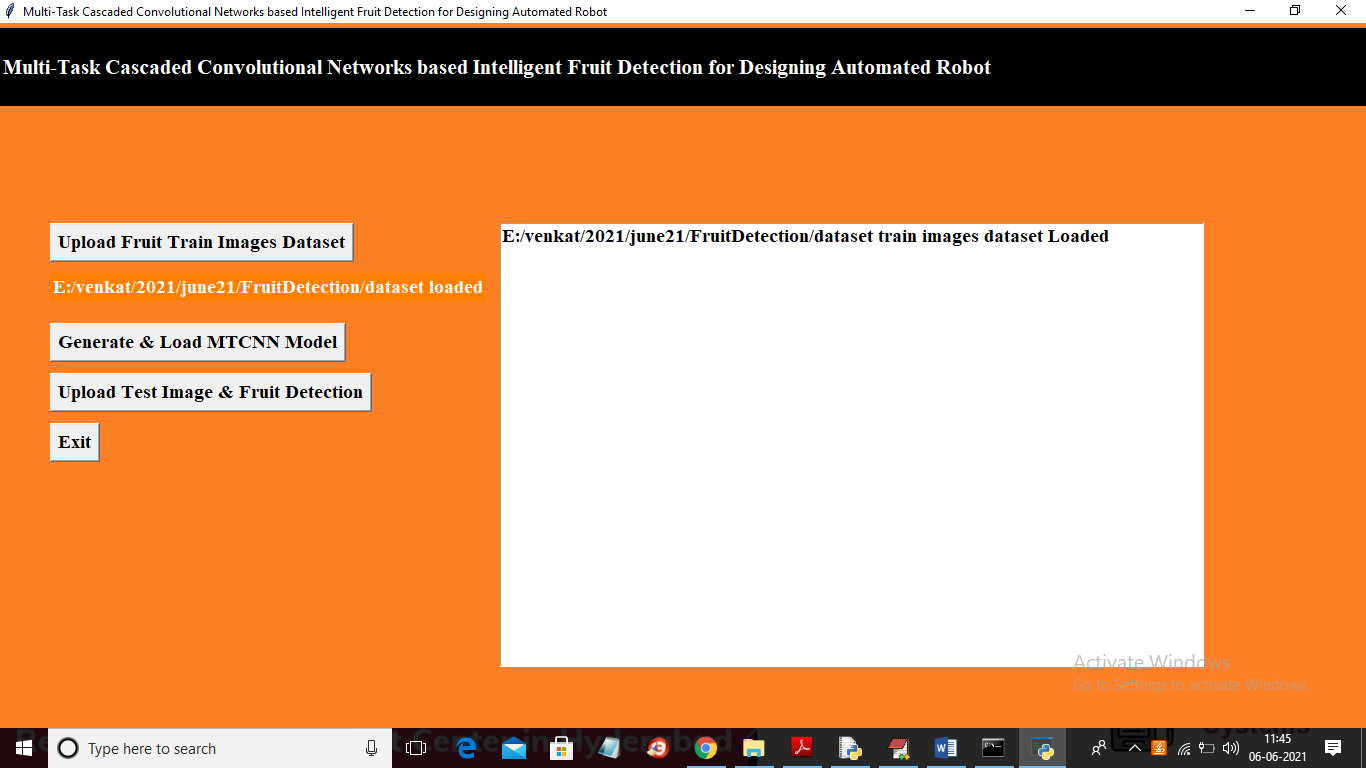
In above screen click on ‘Upload Fruit Train Images Dataset’ button to load dataset



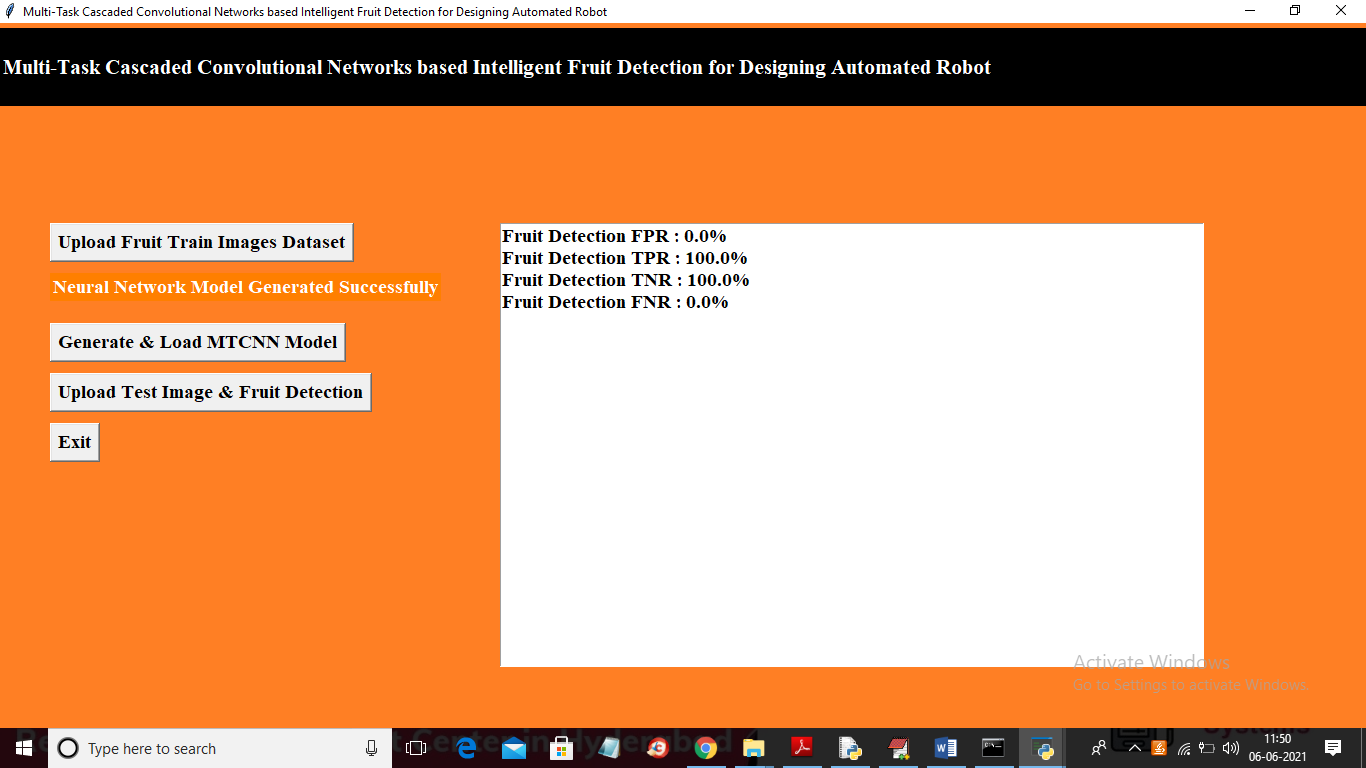
In above screen selecting and uploading ‘dataset’ folder and then click on ‘Select Folder’ to upload dataset images and to get below screen



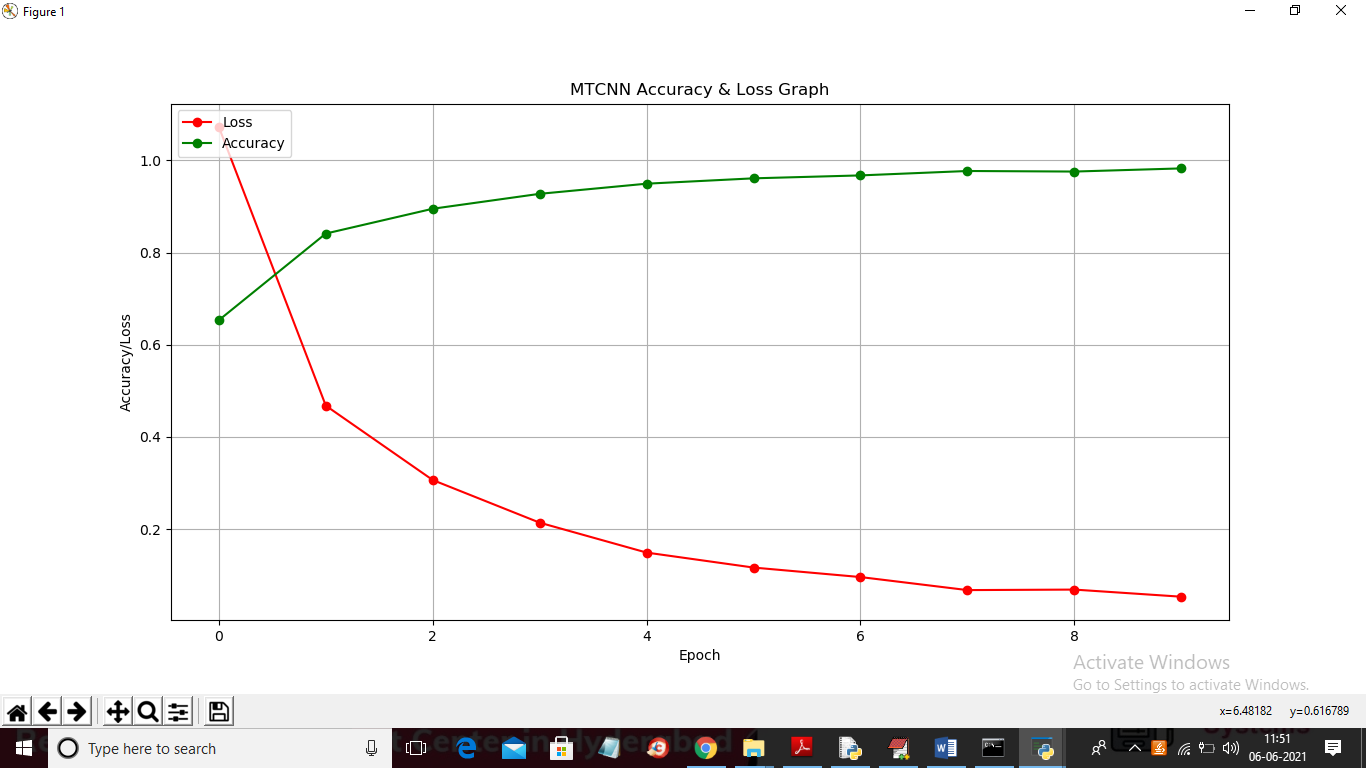
In above screen dataset images loaded and to check image process successfully I am displaying one sample processed image from dataset and now close above image to get below screen



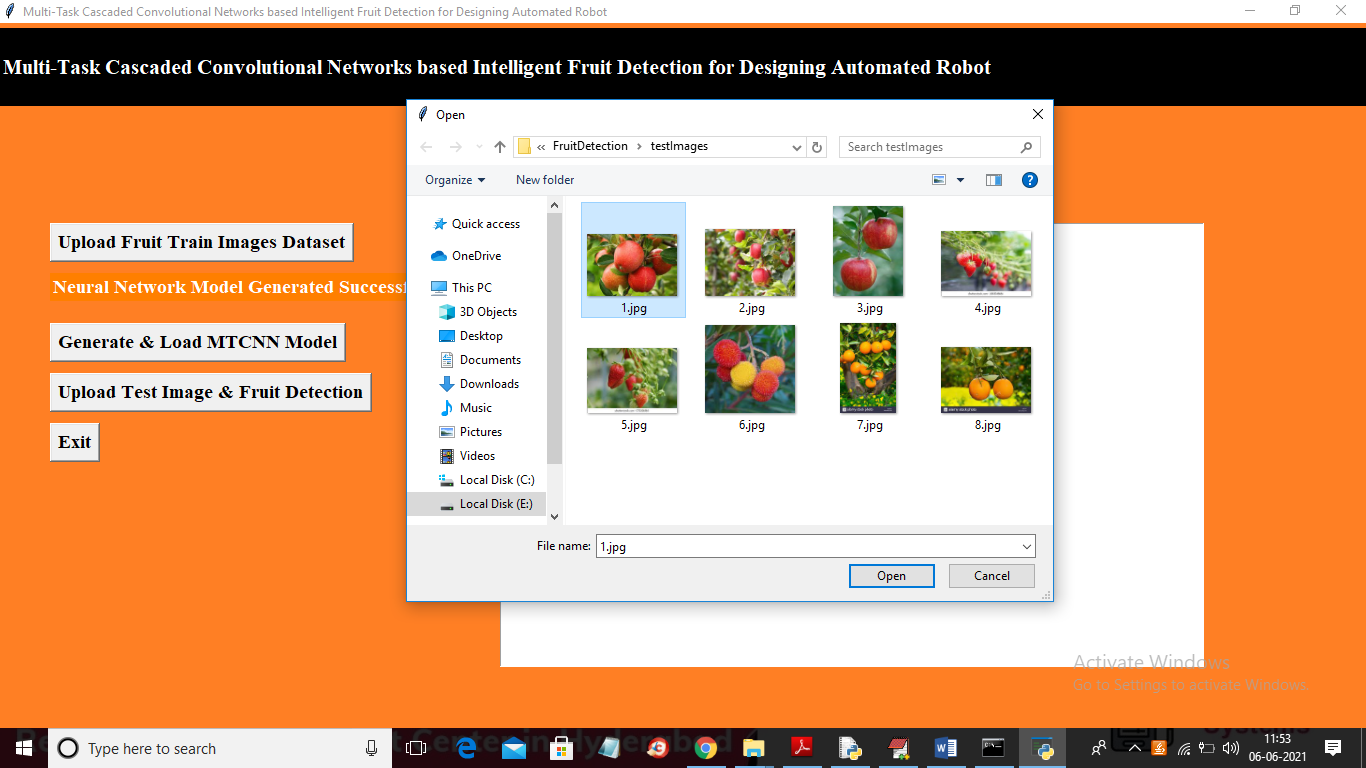
In above screen dataset loaded and now click on ‘Generate & Load MTCNN Model’ button to generate and load model



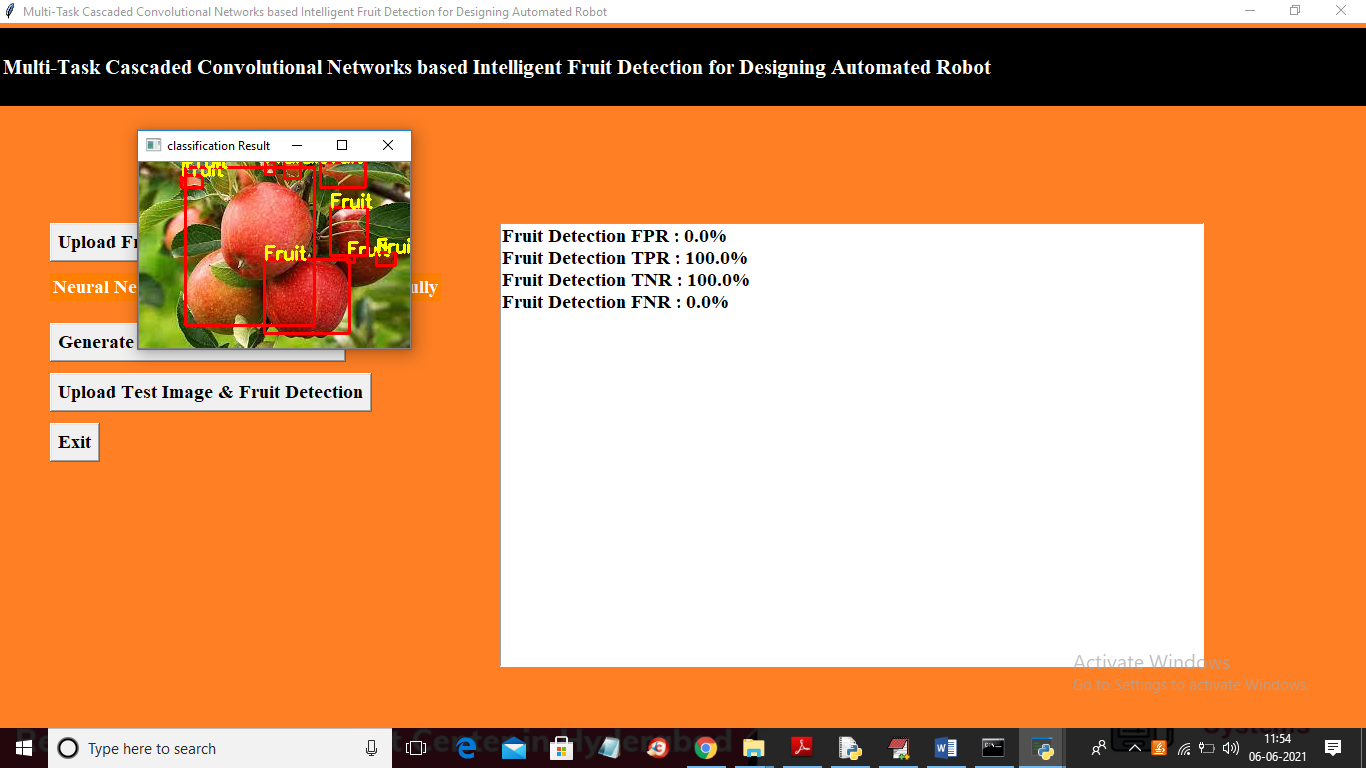
In above screen MTCNN model generated and we got TPR and TNR which means MTCNN prediction on test data is 100% and false prediction rate is 0% and in below graph we can see MTCNN accuracy and loss



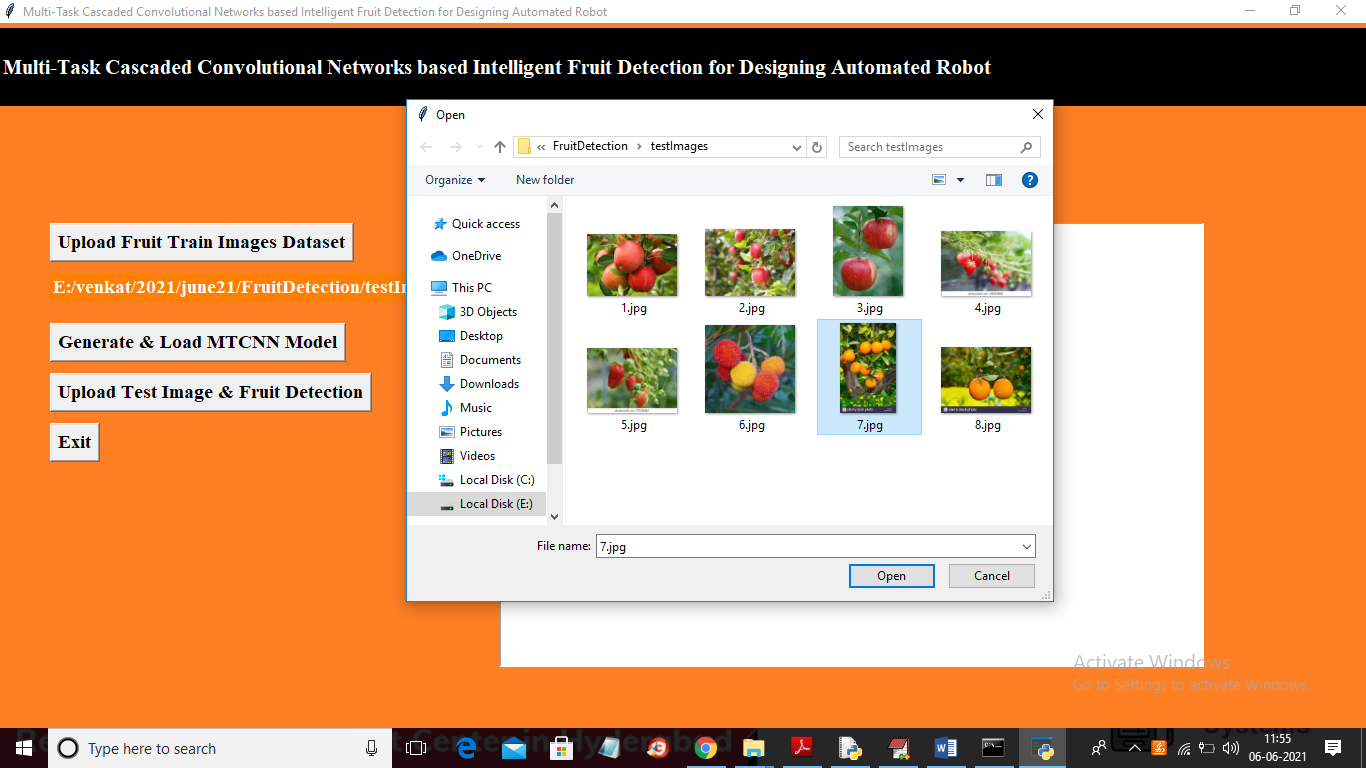
In above graph x-axis represents MTCNN epoch and y-axis represents accuracy and loss value and in above graph red line represents loss and green line represents accuracy and with each increasing epoch we can see loss value decrease and accuracy get increase closer to 100%. Now close above graph and then click on ‘Upload Test Image & Fruit Detection’ button to upload test image and then get fruit detection output



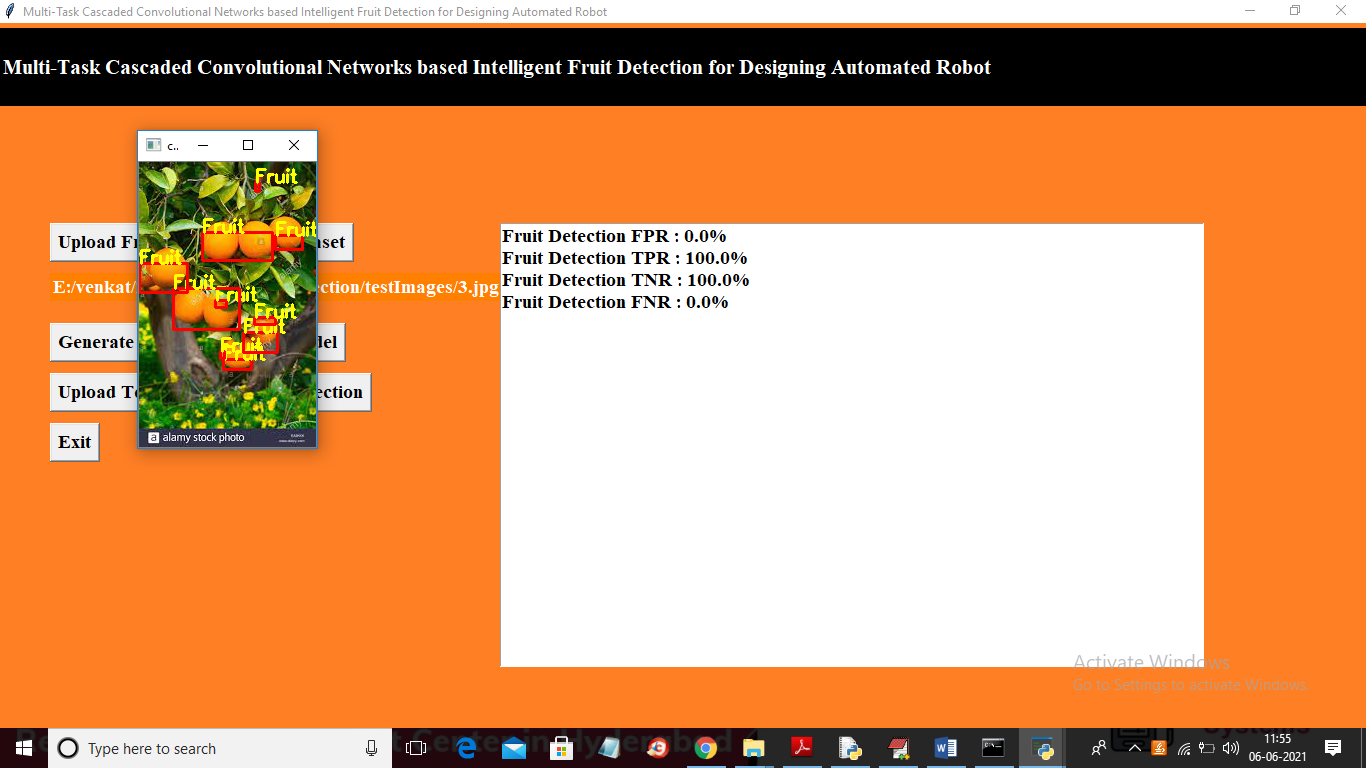
In above screen selecting and uploading ‘1.jpg’ image and then click on ‘Open’ button to get below result

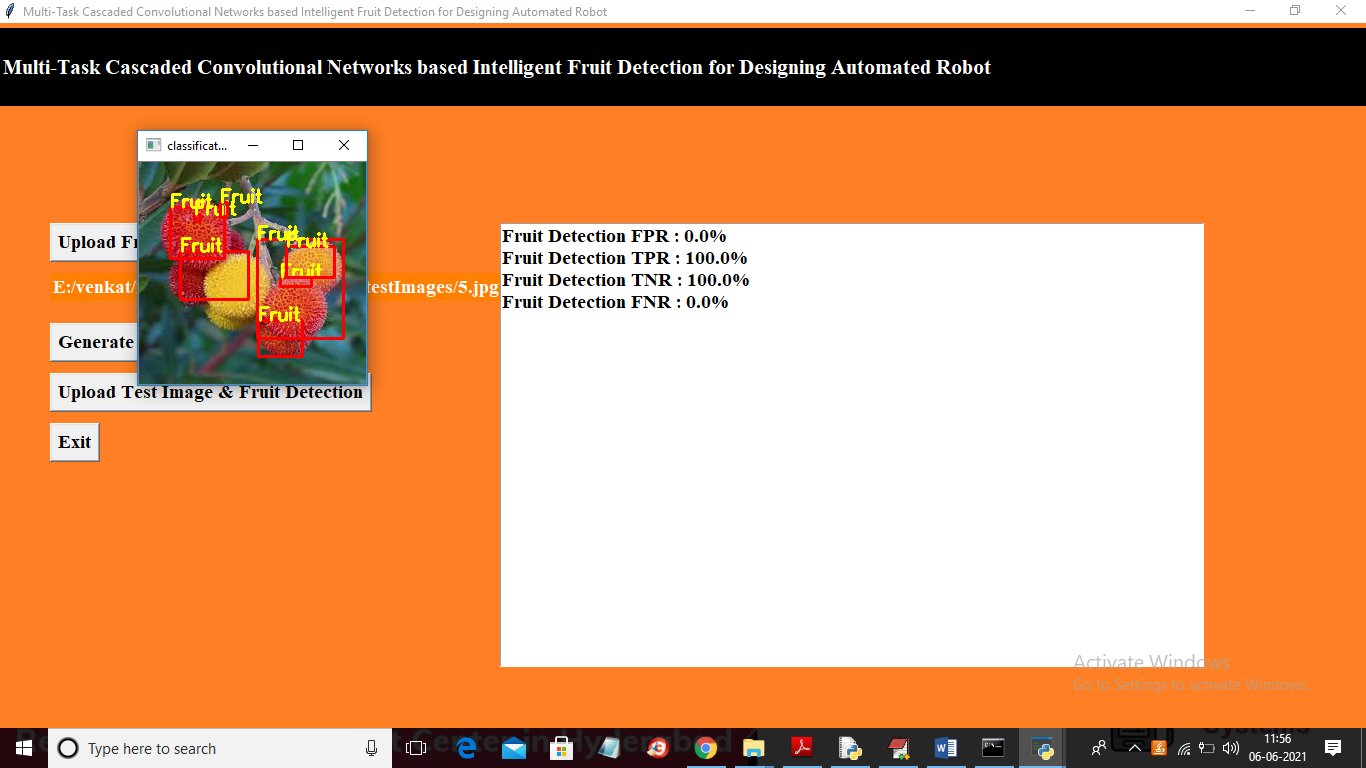


In above screen we can see MTCNN detected fruit and surround them with bounding boxes. Test other image



In above screen uploading ‘7.jpg’ below is the result





Similarly you can test with other images also

**10. CONCLUSION**

In this study, we exploited a multi-task cascaded convolutional networks based detector for fruit detection. We chose apple for our study and collected more than one thousands of images from apple orchards and labeled them. Alongside this, we also added an appropriate amount of supplementary images from internet and ImageNet dataset to create a dataset. Furthermore, we proposed a novel augmented method called fusion augmentation. The comparative experiment results demonstrated that this augmented method can improve the final result. To verify whether the detector could be applied to other kinds of fruits as well, we selected strawberry and orange as two other test fruits. The dataset for training was obtained from ImageNet dataset, which contains hundreds of images. Our results showed that the detector can conveniently adapt to other kinds of fruit as well. Finally, we tested the detector on twelve groups of images with different resolutions. Each group had one hundred images. The average time cost of the detector was less than 80 seconds per one hundred images, which is very close to real-time response.

**11. REFERENCES**

[1] L. M. Azizah, S. F. Umayah, S. Riyadi, C. Damarjati, and N. A. Utama, ‘‘Deep learning implementation using convolutional neural network in mangosteen surface defect detection,’’ in Proc. 7th IEEE Int. Conf. Control Syst., Comput. Eng. (ICCSCE), Nov. 2017, pp. 242–246.

[2] A. Mohapatra, S. Shanmugasundaram, and R. Malmathanraj, ‘‘Grading of ripening stages of red banana using dielectric properties changes and image processing approach,’’ Comput. Electron. Agricult., vol. 143, no. 382, pp. 100–110, 2017.

[3] J. Lu, J. Hu, G. Zhao, F. Mei, and C. Zhang, ‘‘An in-field automatic wheat disease diagnosis system,’’ Comput. Electron. Agricult., vol. 142, pp. 369–379, Nov. 2017.

[4] J. Ma, K. Du, L. Zhang, F. Zheng, J. Chu, and Z. Sun, ‘‘A segmentation method for greenhouse vegetable foliar disease spots images using color information and region growing,’’ Comput. Electron. Agricult., vol. 142, pp. 110–117, Nov. 2017.

[5] N. Behroozi-Khazaei and M. R. Maleki, ‘‘A robust algorithm based on color features for grape cluster segmentation,’’ Comput. Electron. Agricult., vol. 142, pp. 41–49, Nov. 2017. [6] W. Mao, B. Ji, J. Zhan, X. Zhang, and X. Hu, ‘‘Apple location method for the apple harvesting robot,’’ in Proc. 2nd Int. Congr. Image Signal Process., Oct. 2009, pp. 1–5.

[7] A. Durand-Petiteville, S. Vougioukas, and D. C. Slaughter, ‘‘Real-time segmentation of strawberry flesh and calyx from images of singulated strawberries during postharvest processing,’’ Comput. Electron. Agricult., vol. 142, pp. 298–313, Nov. 2017.

[8] Y. Shi, W. Huang, J. Luo, L. Huang, and X. Zhou, ‘‘Detection and discrimination of pests and diseases in winter wheat based on spectral indices and kernel discriminant analysis,’’ Comput. Electron. Agricult., vol. 141, pp. 171–180, Sep. 2017.

[9] J. Lu, W. Suk, H. Gan, and X. Hu, ‘‘Immature citrus fruit detection based on local binary pattern feature and hierarchical contour analysis,’’ Biosyst. Eng., vol. 171, pp. 78–90, Jul. 2018.

[10] A. Gongal, S. Amatya, M. Karkee, Q. Zhang, and K. Lewis, ‘‘Sensors and systems for fruit detection and localization: A review,’’ Comput. Electron. Agricult., vol. 116, pp. 8–19, Aug. 2015.

[11] T. Zhou, S. Yang, L. Wang, J. Yao, and G. Gui, ‘‘Improved cross-label suppression dictionary learning for face recognition,’’ IEEE Access, vol. 6, no. 1, pp. 48716–48725, 2018.

[12] S. Liao, A. K. Jain, and S. Z. Li, ‘‘A fast and accurate unconstrained face detector,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 2, pp. 211–223, Feb. 2016.

[13] Y. Zheng, C. Zhu, K. Luu, C. Bhagavatula, T. H. N. Le, and M. Savvides, ‘‘Towards a deep learning framework for unconstrained face detection,’’ in Proc. IEEE Int. Conf. Biometrics Theory, Appl. Syst. (BTAS), Sep. 2016, pp. 1–8.

[14] K. Zhang, Z. Zhang, H. Wang, Z. Li, Y. Qiao, and W. Liu, ‘‘Detecting faces using inside cascaded contextual CNN,’’ in Proc. IEEE Int. Conf. Comput. Vis., Oct. 2017, pp. 3190–3198.

[15] Z. Yang and R. Nevatia, ‘‘A multi-scale cascade fully convolutional network face detector,’’ in Proc. 23rd Int. Conf. Pattern Recognit. (ICPR), Dec. 2016, pp. 633–638.

[16] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, ‘‘Joint face detection and alignment using multitask cascaded convolutional networks,’’ IEEE Signal Process. Lett., vol. 23, no. 10, pp. 1499–1503, Oct. 2016.

[17] W. Ji, X. Meng, Y. Tao, B. Xu, and D. Zhao, ‘‘Fast segmentation of colour apple image under all-weather natural conditions for vision recognition of picking robots,’’ Int. J. Adv. Robotic Syst., vol. 13, no. 1, pp. 1–24, 2016.

[18] H. Dang, J. Song, and Q. Guo, ‘‘A fruit size detecting and grading system based on image processing,’’ in Proc. 2nd Int. Conf. Intell. Hum.-Mach. Syst. Cybern., vol. 2, Aug. 2010, pp. 83–86.

[19] I. B. Mustaffa, S. Fikri, and B. M. Khairul, ‘‘Identification of fruit size and maturity through fruit images using OpenCV-Python and Rasberry Pi,’’ in Proc. Int. Conf. Robot., Automat. Sci. (ICORAS), Nov. 2017, pp. 1–3.

[20] G. Moradi, M. Shamsi, M. H. Sedaghi, and M. R. Alsharif, ‘‘Fruit defect detection from color images using ACM and MFCM algorithms,’’ in Proc. Int. Conf. Electron. Devices, Syst. Appl., Apr. 2011, pp. 182–186.

[21] A. D. Aggelopoulou, D. Bochtis, S. Fountas, K. C. Swain, T. A. Gemtos, and G. D. Nanos, ‘‘Yield prediction in apple orchards based on image processing,’’ Precis. Agricult., vol. 12, no. 3, pp. 448–456, 2011.

[22] W. Ji, Z. Qian, B. Xu, Y. Tao, D. Zhao, and S. Ding, ‘‘Apple tree branch segmentation from images with small gray-level difference for agricultural harvesting robot,’’ Optik, vol. 127, pp. 11173–11182, Dec. 2016.

[23] A. Rady, N. Ekramirad, A. A. Adedeji, M. Li, and R. Alimardani, ‘‘Hyperspectral imaging for detection of codling moth infestation in GoldRush apples,’’ Postharvest Biol. Technol., vol. 129, pp. 37–44, Jul. 2017.

[24] D. M. Bulanon, T. F. Burks, and V. Alchanatis, ‘‘Image fusion of visible and thermal images for fruit detection,’’ Biosyst. Eng., vol. 103, no. 1, pp. 12–22, 2009.

[25] C. S. Nandi, B. Tudu, and C. Koley, ‘‘A machine vision-based maturity prediction system for sorting of harvested mangoes,’’ IEEE Trans. Instrum. Meas., vol. 63, no. 7, pp. 1722–1730, Jul. 2014. [26] X. Xu, D. Niu, Q. Wang, P. Wang, and D. D. Wu, ‘‘Intelligent forecasting model for regional power grid with distributed generation,’’ IEEE Syst. J., vol. 11, no. 3, pp. 1836–1845, Sep. 2017.