|  |
| --- |
| AUTOMATED FOOD IMAGE CLASSIFICATION USING DEEP LEARNING A PROJECT REPORT SUBMITTED TO  JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR  In partial fulfillment of the requirement for the award of degree of  MASTER OF COMPUTER APPLICATIONS    SUBMITTED BY    DERANGULA VEERANJANEYULU    Roll No: 21P11F0033  Under the Esteemed Guidance of    Dr.K.SAILAJA, MCA, M.Tech, M.Phil, Ph.D  Professor & HOD  Department of Computer Application      DEPARTMENT OF COMPUTER APPLICATIONS  CHADALAWADA RAMANAMMA ENGINEERING COLLEGE  (AUTONOMOUS)  Accredited by NAAC with ‘A’ Grade, Approved by A.I.C.T.E, New Delhi,  Affiliated to JNTUA, Anantapur CHADALAWADA NAGAR,RENIGUNTA ROAD, TIRUPATHI-517506, CHITTOR (Disc.), A.P. INDIA 2022-2023. |
| **CHADALAWADA RAMANAMMA ENGINEERING COLLEGE**  **(AUTONOMUS)**  **DEPARTMENT OF COMPUTER APPLICATIONS** 2021-2023     **CERTIFICATE**    This is to certify that the project work entitled "**AUTOMATED FOOD IMAGE CLASSIFICATION USING DEEP LEARNING**", is a bonafide work done by DERANGULA VEERANJANEYULU (21P11F0033), in the Department of "MASTER OF COMPUTER APPLICATIONS", and submitted to **Chadalawada Ramanamma Engineering College (Autonomous), Tirupati** in partial fulfilment of the requirements for the award of the Master of Computer Applications and the project work carried out by them under my guidance during the academic year **2022-2023.**               |  |  |  | | --- | --- | --- | | **PROJECT GUIDE** | **HEAD OF THE DEPARTMENT** | **PRINCIPAL & DIRECTOR** | | **Dr. K. SAILAJA** | **Dr. K. SAILAJA** | **Dr. BHASKAR PATEL** | | Professor & HOD | Professor & HOD | CREC(A) & Krishna Teja |   Department of Computer Applications Department of Computer Applications Educational Institution        Submitted for the project viva-voce examination held on    INTERNAL EXAMINER: EXTERNAL EXAMINER: |
|  |
| ACKNOWLEDGEMENT    I hereby convey my sincere thanks to my guide **Dr.K.SAILAJA, MCA, M.Tech, M.Phil, Ph.D PROFESSOR & HOD** for her guidance and cooperation during my work.  I hereby wish to express my deep sense of gratitude to **Dr. K. SAILAJA,**  **PROFESSOR & HOD, DEPARTMENT OF COMPUTER APPLICATIONS** for all provisions made and for her constant encouragement throughout the work.  I thank our principal **Dr. BHASKER PATEL** for providing all the facilities and support in completing my project work successfully.  I sincerely thank our Chairperson **Dr. C. KRISHNA MURTHY** for providing necessary infrastructure and resources for the accomplishment of the Seminar report at **Chadalawada Ramanamma Engineering College (Autonomous)**, Tirupati.  I am very much grateful to all the faculty members of Department of Computer Applications for their value-based import of theory and practical subjects, which I have put to use in my project work. I also thank the non-teaching staff for their cooperation and timely help.  I would like to take this opportunity and thank my parents for their kind and moral support last, but not least, I would like to thank all my friends who extended their support either directly or indirectly during the submission of during the Seminar report work.            DERANGULA VEERANJANEYULU  (ROLLNO: 21P11F0033) |

# ABSTRACT

Food image classification is an emerging research field due to its increasing benefits in the health and medical sectors. For sure, in the future automated food recognition tools will help in developing diet monitoring systems, calorie estimation and so on. In this paper, automated methods of food classification using deep learning approaches are presented. SqueezeNet and VGG-16 Convolutional Neural Networks are used for food image classification. It is demonstrated that using data augmentation and by fine-tuning the hyperparameters, these networks exhibited much better performance, making these networks suitable for practical applications in health and medical fields. SqueezeNet being a lightweight network, is easier to deploy and often more desirable. Even with fewer parameters, VGG-16 is able to achieve quite a good accuracy. Higher accuracy of food image classification is further achieved by extracting complex features of food images. The performance of automatic food image classification is further improved by the proposed VGG-16 network. Due to increased network depth, proposed SqueezeNet has achieved significant improvement in accuracy. In Food image classification SqueezeNet is get good classification results compare to VGG-16. The classify food item name with images approximately recognition the item name.

**Keywords** - Tensor flow, SSD, Yolo, Yolo\_v3, GTTS, and Deep Learning.

|  |  |  |
| --- | --- | --- |
| **S.NO** | **DESCRIPTION** | **PAGE NO** |
| **1** | **1.INTRODUCTION** | **1-2** |
| **2** | **2.LITERATURE SURVEY**  2.1 DIFFERENT AUTHORE DISCUSSION | **3-6** |
| **3** | **3.EXISTING SYSTEM**  3.1 EXISTING SYSTEM  3.2 DISADVANTAGES OF EXISTING SYSTEM | **7** |
| **4** | **4.PROPOSED SYSTEM**  4.1 PROPOSED SYSTEM  4.2 ADVANTAGES OF PROPOSED SYSTEM | **8** |
| **5** | **5.MODULE DESCRIPTION**  5.1 MODULE EXPLANATION | **9** |
| **6** | **6.PROJRCT DESIGN**  6.1 SYSTEM ARCHITECTURE  6.2 UML DIAGRAMS  6.3 PROJECT REQUIRMENT | **10-16** |
| **7** | **7.PROJECT IMPLEMENTATION**  7.1 TECHNOLOGIES USED  7.2 SAMPLE CODE  7.3 SYSTEM TESTIG | **17**-**45** |
| 8 | **8.SCREEN SHOTS** | **46-48** |
| 9 | **9.CONCLUSION**  **9.1FUTURE ENHANCMENT** | **49** |
| 10 | **10.REFERENCE** | **50-51** |

# 1. INTRODUCTION

Automatic food recognition is an emerging research topic not only for the social network domain aspect. Indeed, researchers are focusing on this area because of its increasing benefits for medical point of view. Automatic food recognition tools will help in facilitating the decision-making process of calories estimation, quality detection of food, build diet monitoring systems to combat obesity and so on.

On the other hand, food is inherently deformable and shows high divergence in appearance. Since food images have high intraclass variance and low inter-class variance due to which classic approaches do not recognize complex features. This makes food recognition a difficult task for which complex features are not recognized by classic approaches. CNNs can easily identify these features automatically, thus increasing classification accuracy. Therefore, this paper attempts to use CNNs for food image classification.

Image classification has become less complicated with deep learning and availability of larger datasets and computational assets. The Convolution neural network is the most popular and extensively used image classification technique in the latest days. Image classification is performed on diverse food dataset using various transfer learning techniques. The food plays a vital role in human’s life as it provides us different nutrients and consequently it is necessary for every individual to maintain a watch on their eating habits. Therefore, food classification is a quintessential thing for a healthier lifestyle.

Unlike the traditional methods of building a model from the scratch, pre trained models are used in this project which saves the computation time and cost and also has given better results. The food dataset of many classes with many images in each class is used for training and validating. Using these pre-trained models, the given food will be recognized, and the nutrient content will be predicted based on the colour in the image. Statistics show that 95% of the people no longer follow any dietary plan as these restrict the people from consuming their day-to-day food.

So, the primary cause for obesity is imbalance of the amount of food intake and

energy consumed by the individual, and a healthy meal is necessary. Thus, maintaining a healthy diet is an important goal for many people. The process of tracking the number of calories consumed can be very tedious as it requires the user to keep a food journal and perform little messy calculations to estimate the number of calories consumed in every food item. Through this research we try to classify Indian food images into their respective classes.

Traditional image analysis approaches have achieved low classification accuracy in the past, whereas deep learning approaches enabled the identification of food types and their ingredients. The contents of food dishes are typically deformable objects, usually including complex semantics, which makes the task of defining their structure very difficult. Deep learning methods have already shown very promising results in such challenges, so this presentation focuses on some popular approaches and techniques applied in image-based food recognition.

# 2. LITERATURE SURVEY

**[1].Zhou, L., Zhang, C., Liu, F., Qiu, Z., & He, Y.** Deep learning has been proved to be an advanced technology for big data analysis with a large number of successful cases in image processing, speech recognition, object detection, and so on. Recently, it has also been introduced in food science and engineering. To our knowledge, this review is the first in the food domain. In this paper, we provided a brief introduction of deep learning and detailed described the structure of some popular architectures of deep neural networks and the approaches for training a model. We surveyed dozens of articles that used deep learning as the data analysis tool to solve the problems and challenges in food domain, including food recognition, calories estimation, quality detection of fruits, vegetables, meat and aquatic products, food supply chain, and food contamination. The specific problems, the datasets, the preprocessing methods, the networks and frameworks used, the performance achieved, and the comparison with other popular solutions of each research were investigated. We also analyzed the potential of deep learning to be used as an advanced data mining tool in food sensory and consume researches. The result of our survey indicates that deep learning outperforms other methods such as manual feature extractors, conventional machine learning algorithms, and deep learning as a promising tool in food quality and safety inspection. The encouraging results in classification and regression problems achieved by deep learning will attract more research efforts to apply deep learning into the field of food in the future.

**Summary:** we investigated a large number of latest articles related to the APP of deep learning in food, described in detail the proposed structure, training methods, and the final evaluation result of DNNs used to process food image, spectrum, text, and other information in each surveyed article. In the aspect of performance, we compared the deep learning with other existing popular methods, and found that the deep learning method achieves better results than other methods in these reviewed studies. We concluded the advantages and disadvantages of deep learning methods and made a detailed discussion of the challenges and future perspective of deep learning in food

domain. To authors’ knowledge, it is the first survey on the APPs of deep learning in the food domain. The purpose of this review is to encourage researchers and workers in this field to perform more experiments on food with deep learning methods, to present precise solutions for classification or regression problems and put them into practice for the benefits of food quality and safety inspection for human dietary health. At last, we recommend that the combination of deep learning and multisource data fusion including RGB images, spectra, smell, taste, and so on, would be considered to make a more comprehensive assessment of food, the development of full-automatic information acquisition equipment/systems with stable signal output for food and global food data sharing platforms should be studied in the future, since it is still very hard to obtain big data related to food due to the usage of semiautomatic or even manual information acquisition tools and incomplete data management and sharing platforms, the potential of deep learning technology in data mining can be evaluated in food related areas rarely explored such as food sensory and consume, food supply chain, and so on, and successful cases of deep learning such as in food (such as food image recognition, intelligent recipe recommendation APP, and fruit quality evaluation system) can be further transformed into practical products.

**[2].Farinella, G. M., Moltisanti, M., &Battiato, S.**The classification of food images is an interesting and challenging problem since the high variability of the image content which makes the task difficult for current state-of the-art classification methods. The image representation to be employed in the classification engine plays an important role. We believe that texture features have been not properly considered in this application domain. This paper points out, through a set of experiments, that textures are fundamental to properly recognize different food items. For this purpose, the bag of visual words model (BoW) is employed. Images are processed with a bank of rotation and scale invariant filters and then a small codebook of Textons is built for each food class. The learned class-based Textons are hence collected in a single visual dictionary. The food images are represented as visual words distributions (Bag of Textons) and a Support Vector Machine is used for the classification stage. The experiments demonstrate that the image representation based on Bag of Textons is more accurate than existing (and more complex) approaches in classifying the 61 classes of the Pittsburgh Fast-Food Image Dataset. Automatic food classification is an emerging research topic, not only to recognize food images for the web and social networks application domain

**Summary:** Evaluates the class-based Bag of Textons representation in the context of food classification. The MRS4 filter banks are used to build class-based Textons vocabularies. The image representation is coupled with a Support Vector Machine for classification purpose. This representation is compared with respect to other state-of- the-art methods on the public available Pittsburgh Fast-Food Image Dataset (PFID). The class-based Bag of Textons representation obtained better results with respect to all the other methods. Future works could be devoted to the exploitation of Textons (and/or other types of texture-like feature, such as CLBP) in joint with other kind of features as well as in encoding spatial information between local Textons (e.g., through correlograms of textons) to better discriminate food items. Moreover, could be important to test the Textons based representation (both Global and Class-Based) on bigger food image datasets for both classification and retrieval purposes.

**[3]. Wang, M., Wan, Y., Ye, Z., & Lai, X.** Support vector machine (SVM) is one of the most successful classifiers for remote sensing image classification. However, the performance of SVM is mainly dependent on its parameters; in addition, for remote sensing images with high-dimensional features, feature redundancy will have a major influence on the classification efficiency and accuracy. Feature selection and parameter optimization are two important factors for improving the performance of SVM and are traditionally solved separately. In fact, these two issues are affected by each other, so to obtain the better classification performance, selection of the optimal feature subset and tuning of SVM parameters should be considered simultaneously, as they both belong to the combinatorial optimization problem and could be handled with evolutionary algorithms and swarm intelligence algorithms. In this paper, a remote sensing image classification technique based on the optimal SVM is proposed, in which the parameters of SVM and feature selection are handled integrally by a modified coded ant colony optimization algorithm combined with genetic algorithm. The results are compared with other evolutionary algorithms and swarm intelligence algorithms, such as genetic algorithm (GA), binary-coded particle swarm optimization (BPSO) algorithm, binary-coded ant colony optimization (BACO) algorithm, binarycoded

differential evolution (BDE) algorithm, and binary-coded cuckoo search (BCS) algorithm. It is demonstrated that the proposed method is robust, adaptive and exhibits the better performance than the other methods involved in the paper in terms of fitness values, so could be suitable for some practical applications.

**[4] Xia, J., Ghamisi, P., Yokoya, N., & Iwasaki, A.** Classification techniques for hyperspectral images based on random forest (RF) ensembles and extended multi extinction profiles (EMEPs) are proposed as a means of improving performance. To this end, five strategies—bagging, boosting, random subspace, rotation-based, and boosted rotation-based—are used to construct the RF ensembles. EPs, which are based on an extrema-oriented connected filtering technique, are applied to the images associated with the first informative components extracted by independent component analysis, leading to a set of EMEPs. The effectiveness of the proposed method is investigated on two benchmark hyperspectral images: the University of Pavia and Indian Pines. Comparative experimental evaluations reveal the superior performance of the proposed methods, especially those employing rotation-based and boosted rotation- based approaches. An additional advantage is that the CPU processing time is acceptable. Hyperspectral imaging sensors generate tens or hundreds of narrow bands with very fine spectral resolution. Therefore, they can provide excellent discriminability of different materials. With the development of hyperspectral technology, the sensors can also make further improvements on spatial resolution, which allows us to describe the spatial structures in the scene with clarity.

**Summary:** we developed a framework to classify hyperspectral images. Our framework uses RF ensembles for the classification with spatial information modeled by EMEPs. The proposed methods were tested on two benchmark hyperspectral data sets: Indian Pines AVIRIS and University of Pavia ROSIS images. Different strategies were used to construct RF ensembles, and the results compared in terms of classification accuracies and CPU time. Experimental results indicate good generalization performance of RF ensembles on EMEPs features. Particular attention should be paid to the BRoRF classifier, which shows the best results due to its property of generating the ensemble with high accuracy of member classifiers as well as diversity.

**3. EXISTING SYSTEM**

**3.1 EXISTING SYSTEM**

This model emphasizes an existing method that which is designed using the some of the algorithms of deep learning. Here the process is performed using the ResNet51, which is one of the transfer learning methods, but this could not get the high accuracy.

**3.2 DRAWBACKS OF EXISTING SYSTEM:**

* + - * **Limited accuracy:**

While deep learning models such as ResNet have achieved impressive results in automated food image classification, the accuracy of these models can be limited by factors such as lighting, image quality, and variability in food appearance.

* + - * **Dataset bias:**

The accuracy of automated food image classification can be affected by bias in the training dataset, which may not be representative of the full range of foods or cultural variations in food preparation and presentation.

* + - * **Difficulty in identifying specific ingredients:**

While automated food image classification can be used to identify general categories of food, such as fruits or vegetables, it can be difficult to identify specific ingredients or nutritional information.

**4.PROPOSED SYSTEM**

**4.1 PROPOSED SYSTEM**

In purposed method we are performing the classification of either the Plant nutrient deficiency identification using Convolution Neural Network (CNN) of deep learning along with the Transfer learning methods. As image analysis based approaches for nutrient deficiency detection. Hence, proper classification is important for the proper nutrition that which will be possible by using our proposed method. Block diagram of proposed method is shown below.

**4.2 ADVANTAGES IN PROPOSED SYSTEM**

## Accurate classification:

By using deep learning techniques, the proposed system can achieve accurate and reliable classification of nutrient deficiencies in plants, which can help farmers and growers take targeted actions to correct the deficiency and optimize crop yields.

## High performance:

By leveraging the power of CNNs and Transfer Learning methods, the proposed system can achieve high performance and efficiency, which can help farmers and growers save time and resources.

## Easy identification:

The proposed system can be easily integrated into existing agricultural processes and can be used by farmers and growers with little to no technical expertise, making it accessible and user-friendly.

## 5. MODULES DESCRIPTION

**5.1 USER**

**Create Data:**

User Once Run The Code And Opens Webcam.

**Object Detection:**

The System Can Detect The Object Using Yolo.

**Take Snapshots:**

User Click The Take Snapshot Button, Hence The System Takes Image.

**Audio Feedback:**

Once The Object Is Detected Using Yolo, It Produces An Audio Feedback Of The Objects.

**View Results:**

User View’s The Generated Results from the Model.

**5.2 SYSTEM**

**Model Checking:**

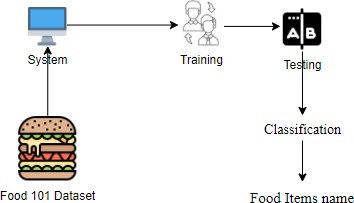
System Can Be Detecting The Object With Yolo Model.

**Generate Results:**

It Generates The Result For User.

# 6. SYSTEM DESIGN

System Design Is The Process Of Defining The Architecture, Components, Modules, Interfaces, And Data For A System To Satisfy Specified Requirements. One Could See It As The Application Of Systems Theory To Product Development. There Is Some Overlap With The Disciplines Of Systems Analysis, Systems Architecture And Systems Engineering. If The Broader Topic Of Product Development "Blends The Perspective Of Marketing, Design, And Manufacturing Into A Single Approach To Product Development, Then Design Is The Act Of Taking The Marketing Information And Creating The Design Of The Product To Be Manufactured. Systems Design Is Therefore The Process Of Defining And Developing Systems To Satisfy Specified Requirements Of The User.

**6.1 SYSTEM ARCHITECTURE** 

***Fig 5.1: Architecture for*** *Automated Food Inage Classification Using Deep Learning*

**6.2 UML DIAGRAMS**

The Unified Modeling Language (UML) Is A Standard Language For Specifying,

Visualizing, Constructing, And Documenting The Artifacts Of Software Systems, As Well As For Business Modelling And Other Non-Software Systems.

The UML Represents A Collection Of Best Engineering Practices That Have Proven Successful In The Modelling Of Large And Complex Systems. The UML Is A Very Important Part Of Developing Objects Oriented Software And The Software Development Process. The UML Uses Mostly Graphical Notations To Express The Design Of Software Projects. Using The UML Helps Project Teams Communicate, Explore Potential Designs, And Validate The Architectural Design Of The Software.

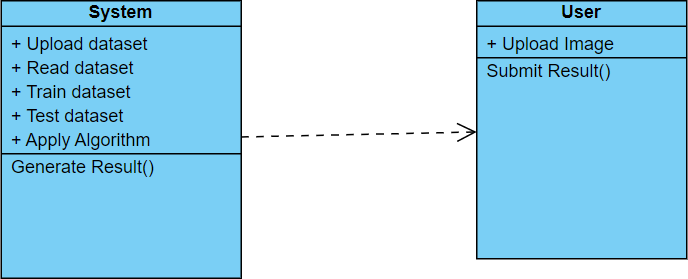
In this project seven basic UML diagrams have been explained

* Class Diagram
* Use Case Diagram
* Collaboration Diagram
* Sequence Diagram
* Activity Diagram
* Component Diagram
* Deployment Diagram

**6.2.1 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

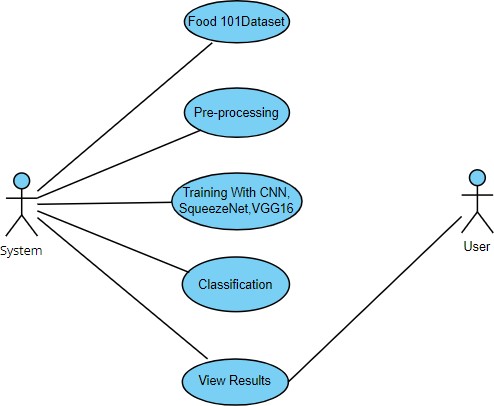
By Showing The System’s Classes, Their Attributes, Operations (Or Methods), And The Relationships Among The Classes. It Explains Which Class Contains Information Accordingly.



**Fig. 6.2.1: *Class Diagram for Automated Food Inage Classification Using Deep Learning***

**6.2.2 USE CASE DIAGRAM**

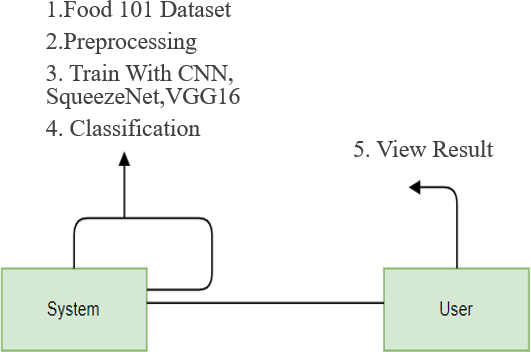
A Use Case Diagram In The Unified Modelling Language (UML) Is A Type Of Behavioral Diagram Defined By And Created From A Use-Case Analysis. Its Purpose Is To Present A Graphical Overview Of The Functionality Provided By A System In Terms Of Actors, Their Goals (Represented As Use Cases), And Any Dependencies 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.



***Fig. 6.2.2: Use Case Diagram for Automated Food Inage Classification Using Deep Learning***

**6.2.3 COLLABORATION DIAGRAM**

In Collaboration Diagram The Method Call Sequence Is Indicated By Some Numbering Technique As Shown Below. The Number Indicates How The Methods Are Called One After Another. We Have Taken The Same Order Management System To Describe The Collaboration Diagram. The Method Calls Are Similar To That Of A Sequence Diagram. But The Difference Is That The Sequence Diagram Does Not Describe The Object Organization Whereas The Collaboration Diagram Shows The Object Organization.



### Fig. 6.2.3: Collaboration Diagram for Automated Food Inage Classification Using Deep Learning

**6.2.4 SEQUENCE DIAGRAM**

A Sequence Diagram Is A Type Of Interaction Diagram Because It Describes How

And In What Order A Group Of Objects Works Together. These Diagrams Are Used By

Software Developers And Business Professionals To Understand Requirements For A

New System Or To Document An Existing Process. Sequence Diagrams Are Sometimes

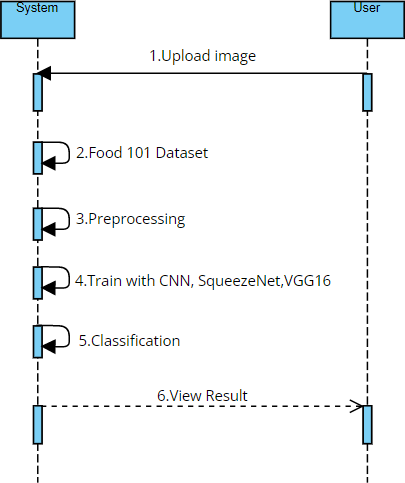
Known As Event Diagrams Or Event Scenarios. A Sequence Diagram Is The Most

Commonly Used Interaction Diagram. Interaction Diagram – An Interaction Diagram Is Used

To Show The Interactive Behaviour Of A System. Since Visualizing The Interactions In

A System Can Be A Cumbersome Task, We Use Different Types Of Interaction Diagrams

To Capture Various Features And Aspects Of Interaction In A System.

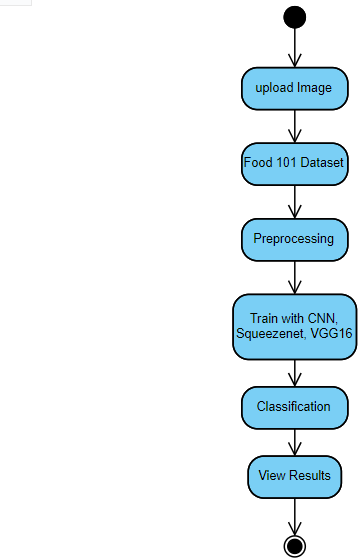


***Fig. 6.2.4: Sequence Diagram for Automated Food Inage Classification Using Deep Learning***

**6.2.5 ACTIVITY DIAGRAM**

Activity Diagram Is Another Important Diagram In UML To Describe Dynamic Aspects Of The System. Activity Diagram Is Basically A Flow Chart To Represent The Flow Form One Activity To Another Activity. The Activity Can Be Described As An Operation Of The System. So, The Control Flow Is Drawn From One Operation To Another. This Flow Can Be Sequential, Branched Or Concurrent. Activity Diagrams Deals With All Type Of Flow Control By Using Different Elements Like Fork, Join Etc. An Activity Diagram Is A Behavioural Diagram I.E. It Depicts The Behaviour Of A System. An Activity Diagram Portrays The Control Flow From A Start Point To A Finish Point Showing The Various Decision Paths That Exist While The Activity Is Being Executed. We Can Depict Both Sequential Processing And Concurrent Processing Of Activities Using An Activity Diagram. They Are Used In Business And Process

Modelling Where Their Primary Use Into Depict The Dynamic Aspects Of A System. An Activity Diagram Is Very Similar To A Flowchart.



### Fig. 6.2.5: Activity Diagram for Automated Food Inage Classification Using Deep Learning

**6.2.6 DEPLOYMENT DIAGRAM**

Deployment Diagram Are Used To Visualize The Topology Of The Physical Components Of A System,

Where The Software Components Are Deployed. Deployment Diagrams Are Used To Describe The

Static Deployment View Of A System. Deployment Diagrams Consist Of Nodes And Their Relationships.



***Fig. 6.2.6: Deployment Diagram for Automated Food Inage Classification Using Deep Learning***

**6.3 SYSTEM REQUIREMENTS**

**6.3.1 HARDWARE REQUIREMENTS:**

* + - **Processor :** I3/Intel Processor
    - **RAM :** 4GB (min)
    - **Hard Disk :** 160 GB
    - **Key Board :** Normal and Multimedia
    - **Mouse :** Compatible Mouse

**6.3.2 SOFTWARE REQUIREMENTS:**

* + - **Operating System :** Windows 7
    - **Server-side Script :** Python
    - **Language :** Python
    - **Browser :** Any latest browser

# 7.PROJECT IMPLEMENTATION

**7.1 SOFTWARE DESCRIPTION**

**Python:**

## Python

**What Is a Script?**

Up to this point, I have concentrated on the interactive programming capability of Python. This is a very useful capability that allows you to type in a program and to have it executed immediately in an interactive mode

**Scripts are reusable**

Basically, a script is a text file containing the statements that comprise a Python program. Once you have created the script, you can execute it over and over without having to retype it each time.

**Scripts are editable**

Perhaps, more importantly, you can make different versions of the script by modifying the statements from one file to the next using a text editor. Then you can execute each of the individual versions. In this way, it is easy to create different programs with a minimum amount of typing.

You will need a text editor

Just about any text editor will suffice for creating Python script files.

You can use *Microsoft Notepad, Microsoft WordPad, Microsoft Word,* or just about any word processor if you want to.

## Difference between a script and a program

**Script:**

Scripts are distinct from the core code of the application, which is usually written in a different language, and are often created or at least modified by the end-user. Scripts are often interpreted from source code or byte code, whereas the applications they control are traditionally compiled to native machine code.

**Program:**

The program has an executable form that the computer can use directly to execute the instructions.

The same program in its human-readable source code form, from which executable programs are derived (e.g., compiled)

## Python

What is Python? Chances you are asking yourself this. You may have found this book because you want to learn to program but don’t know anything about programming languages. Or you may have heard of programming languages like C, C++, C#, or Java and want to know what Python is and how it compares to “big name” languages. Hopefully I can explain it for you.

**Python concepts**

If you’re not interested in the how’s and whys of Python, feel free to skip to the next chapter. In this chapter I will try to explain to the reader why I think Python is one of the best languages available and why it’s a great one to start programming with.

* + Open-source general-purpose language.
  + Object Oriented, Procedural, Functional
  + Easy to interface with C/Obj/Java/Fortran
  + Easy-is to interface with C++ (via SWIG)
  + Great interactive environment
  + Great interactive environment

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.

## History of Python:

Python was developed by Guido van Possum 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 Possum still holds a vital role in directing its progress.

## Python Features:

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-maintained.
* 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.

The Python Programming Language can be characterized by the following buzzwords:

* + Easy to code and easy to read
  + Expressive
  + Free and Open source
  + High level language
  + Portable
  + Interpreted
  + Extensible
  + Embeddable
  + Object oriented
  + Has large Standard library
  + GUI Programming
  + Dynamically typed

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. Inobject-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simple way of doing object- oriented programming, especially, when compared to languages like C++ or Java. Python is a general purpose and high-level programming language. You can use Python for developing desktop GUI applications, websites and web applications Also.

Python, as a high-level programming language, allows you to focus on core functionality of the application by taking care of common programming tasks. An important goal of Python's developers is keeping it fun to use.

This is reflected in the language's name—a tribute to the British comedy group Monty Python and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs instead of the standard foo and bar. Users and admirers of Python, especially those considered knowledgeable or experienced, are often referred to as Pythonistas. The simple syntax rules of the programming language further make it easier for you to keep the code base readable and application maintainable. There are also a number of reasons why you should prefer Python to other programming languages.

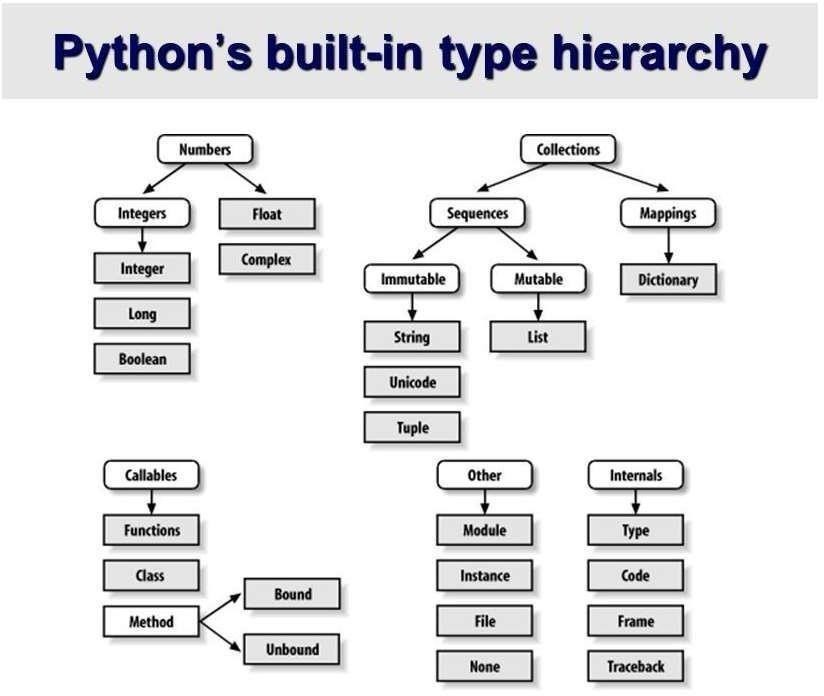
**Python is Expressive**

Suppose we have two languages A and B, and all programs that can be made in A can be made in B using local transformations. However, there are some programs that can be made in B, but not in A, using local transformations. Then, Bis said to be more expressive than A. Python provides us with a myriad of constructs that help us focus on the solution rather than on the syntax.

**Python is a High- Level Language**

It is a high-level language. This means that as programmers, we don’t need to remember the system architecture. Nor do we need to manage the memory. This makes it more programmer-friendly and is one of the key python features**.**

**Python Is Portable**

You can take one code and run it on any machine, there is no need to write different code for different machines. This makes Python a portable language. Let’s assume you’ve written a Python code for your Windows machine. Now, if you want to run it on a Mac, you don’t need to make changes to it for the same. Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution. Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications.

## Variables

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

## Standard Data Types

The data stored in memory can be of many types. For example, a person's age is stored as a numeric value and his or her address is stored as alphanumeric characters. Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.

Python has five standard data types −

* Numbers
* String
* List
* Tuple
* Dictionary

## Python Numbers

Number data types store numeric values. Number objects are created when you assign a value to them

## Python Strings

Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes. Subsets of strings can be taken using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end

## Python Lists

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type.

The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The plus (+) sign is the list concatenation operator, and the asterisk (\*) is the repetition operator.

## Python Tuples

A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.

The main differences between lists and tuples are: Lists are enclosed in brackets ([ ]) and their elements and size can be changed, while tuples are enclosed in parentheses ((

)) and cannot be updated. Tuples can be thought of as read-only lists.

## Python Dictionary

Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).

## Python libraries

1. Requests. The most famous http library written by Kenneth remits. It’s a must have for every python developer.
2. Scrappy. If you are involved in web scraping then this is a must have library for you. After using this library, you won’t use any other.
3. Python. A guy toolkit for python. I have primarily used it in place of tinder. You will really love it.
4. Pillow. A friendly fork of PIL (Python Imaging Library). It is more user friendly than PIL and is a must have for anyone who works with images.
5. SQL Alchemy. A database library. Many love it and many hate it. The choice is yours.
6. Beautiful Soup. I know it’s slow but this xml and html parsing library is very useful for beginners.
7. Twisted. The most important tool for any network application developer. It has a very beautiful ape and is used by a lot of famous python developers.
8. Numbly. How can we leave this very important library? It provides some advance math functionalities to python.
9. Skippy. When we talk about numbly then we have to talk about spicy. It is a library of algorithms and mathematical tools for python and has caused many scientists to switch from ruby to python.
10. Matplotlib. A numerical plotting library. It is very useful for any data scientist or any data analyzer.
11. Pygmy. Which developer does not like to play games and develop them? This library will help you achieve your goal of 2d game development.
12. Piglet. A 3d animation and game creation engine. This is the engine in which the famous [python port](https://github.com/fogleman/Minecraft) of mine craft was made
13. Pit. A GUI toolkit for python. It is my second choice after python for developing GUI’s for my python scripts.
14. Pit. Another python GUI library. It is the same library in which the famous Bit torrent client is created.
15. Scaly. A packet sniffer and analyzer for python made in python.
16. Pywin32. A python library which provides some useful methods and classes for interacting with windows.
17. Notch. Natural Language Toolkit – I realize most people won’t be using this one, but it’s generic enough. It is a very useful library if you want to manipulate strings. But its capacity is beyond that. Do check it out.
18. Nose. A testing framework for python. It is used by millions of python developers. It is a must have if you do test driven development.
19. Simply. Simply can-do algebraic evaluation, differentiation, expansion, complex numbers, etc. It is contained in a pure Python distribution.
20. I Python. I just can’t stress enough how useful this tool is. It is a python prompt on steroids. It has completion, history, shell capabilities, and a lot more. Make sure that you take a look at it.

## NumPy

Humpy’s main object is the homogeneous multidimensional array. It is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In numbly dimensions are called *axes*. The number of axes is *rank*.

* Offers Matlab-ish capabilities within Python
* Fast array operations
* 2D arrays, multi-D arrays, linear algebra etc.

## Matplotlib

* High quality plotting library.

## Python class and objects

These are the building blocks of OOP. Class creates a new object. This object can be anything, whether an abstract data concept or a model of a physical object, e.g. a chair. Each class has individual characteristics unique to that class, including variables and methods. Classes are very powerful and currently “the big thing” in most programming languages. Hence, there are several chapters dedicated to OOP later in the book.

The class is the most basic component of object-oriented programming. Previously, you learned how to use functions to make your program do something.

Now will move into the big, scary world of Object-Oriented Programming (OOP). To be honest, it took me several months to get a handle on objects.

When I first learned C and C++, I did great; functions just made sense for me.

Having messed around with BASIC in the early ’90s, I realized functions were just like subroutines so there wasn’t much new to learn.

However, when my C++ course started talking about objects, classes, and all the new features of OOP, my grades definitely suffered.

Objects are an encapsulation of variables and functions into a single entity. Objects get their variables and functions from classes. Classes are essentially a template to create your objects.

Here’s a brief list of Python OOP ideas:

* The class statement creates a class object and gives it a name. This creates a new namespace.
* Assignments within the class create class attributes. These attributes are accessed by qualifying the name using dot syntax: ClassName.Attribute.
* Class attributes export the state of an object and its associated behavior. These attributes are shared by all instances of a class.
* Calling a class (just like a function) creates a new instance of the class. This is where the multiple copy’s part comes in.
* Each instance gets ("inherits") the default class attributes and gets its own namespace. This prevents instance objects from overlapping and confusing the program.
* Using the term self identifies a particular instance, allowing for per-instance attributes. This allows items such as variables to be associated with a particular instance.

## Exceptions

I’ve talked about exceptions before but now I will talk about them in depth. Essentially, exceptions are events that modify program’s flow, either intentionally or due to errors. They are special events that can occur due to an error, e.g. trying to open a file that doesn’t exist, or when the program reaches a marker, such as the completion of a loop. Exceptions, by definition, don’t occur very often; hence, they are the "exception to the rule" and a special class has been created for them. Exceptions are everywhere in Python. Virtually every module in the standard Python library uses them, and Python itself will raise them in a lot of different circumstances.

Here are just a few examples:

* Accessing a non−existent dictionary key will raise a Key Error exception.
* Searching a list for a non−existent value will raise a Value Error exception
* Calling a non−existent method will raise an Attribute Error exception.
* Referencing a non−existent variable will raise a Name Error exception.
* Mixing data types without coercion will raise a Type Error exception.

One use of exceptions is to catch a fault and allow the program to continue working; we have seen this before when we talked about files.

This is the most common way to use exceptions. When programming with the Python command line interpreter, you don’t need to worry about catching exceptions.

Your program is usually short enough to not be hurt too much if an exception occurs. Plus, having the exception occur at the command line is a quick and easy way to tell if your code logic has a problem.

However, if the same error occurred in your real program, it will fail and stop working. Exceptions can be created manually in the code by raising an exception.

It operates exactly as a system-caused exceptions, except that the programmer is doing it on purpose. This can be for a number of reasons. One of the benefits of using exceptions is that, by their nature, they don’t put any overhead on the code processing. Because exceptions aren’t supposed to happen very often, they aren’t processed until they occur.

Exceptions can be thought of as a special form of the if/elf statements. You can realistically do the same thing with if blocks as you can with exceptions.

However, as already mentioned, exceptions aren’t processed until they occur; if blocks are processed all the time.

Proper use of exceptions can help the performance of your program.

The more infrequent the error might occur, the better off you are to use exceptions; using if blocks requires Python to always test extra conditions before continuing.

Exceptions also make code management easier: if your programming logic is mixed in with error-handling if statements, it can be difficult to read, modify, and debug your program.

## User-Defined Exceptions

Making your own exceptions involves object-oriented programming, which will be covered in the next chapter. To make a custom exception, the programmer determines which base exception to use as the class to inherit from, e.g. making an exception for negative numbers or one for imaginary numbers would probably fall under the Arithmetic Error exception class. To make a custom exception, simply inherit the base exception and define what it will do.

## Python modules

Python allows us to store our code in files (also called modules). This is very useful for more serious programming, where we do not want to retype a long function definition from the very beginning just to change one mistake. In doing this, we are essentially defining our own modules, just like the modules defined already in the Python library. To support this, Python has a way to put definitions in a file and use them in a script or in an interactive instance of the interpreter. Such a file is called a module; definitions from a module can be *imported* into other modules or into the *main* module.

## Testing code

As indicated above, code is usually developed in a file using an editor. To test the code, import it into a Python session and try to run it.

Usually there is an error, so you go back to the file, make a correction, and test again. This process is repeated until you are satisfied that the code works. T

His entire process is known as the development cycle.

There are two types of errors that you will encounter. Syntax errors occur when the form of some command is invalid.

This happens when you make typing errors such as misspellings, or call something by the wrong name, and for many other reasons. Python will always give an error message for a syntax error.

## Functions in Python

It is possible, and very useful, to define our own functions in Python. Generally speaking, if you need to do a calculation only once, then use the interpreter. But when you or others have need to perform a certain type of calculation many times, then define a function.

You use functions in programming to bundle a set of instructions that you want to use repeatedly or that, because of their complexity, are better self-contained in a sub-program and called when needed. That means that a function is a piece of code written to carry out a specified task.

### To carry out that specific task, the function might or might not need multiple inputs.

### When the task is carved out, the function can or cannot return one or more values.

### There are three types of functions in python Help (), min (), print ().

Namespaces in Python are implemented as Python dictionaries, this means it is a mapping from names (keys) to objects (values). The user doesn't have to know this to write a Python program and when using namespaces.

Some namespaces in Python:

* global names of a module
* local names in a function or method invocation
* built-in names: this namespace contains built-in functions (e.g. abs(), camp(), ...) and built-in exception names

## Garbage Collection

Garbage Collector exposes the underlying memory management mechanism of Python, the automatic garbage collector. The module includes functions for controlling how the collector operates and to examine the objects known to the system, either pending collection or stuck in reference cycles and unable to be freed.

## Python XML Parser

XML is a portable, open-source language that allows programmers to develop applications that can be read by other applications, regardless of operating system and/or developmental language.

What is XML? The Extensible Markup Language XML is a markup language much like HTML or SGML.

This is recommended by the World Wide Web Consortium and available as an open standard.

XML is extremely useful for keeping track of small to medium amounts of data without requiring a SQL-based backbone.

XML Parser Architectures and APIs the Python standard library provides a minimal but useful set of interfaces to work with XML.

The two most basic and broadly used APIs to XML data are the SAX and DOM

interfaces.

Simple API for XML SAX: Here, you register call-backs for events of interest and then let the parser proceed through the document.

This is useful when your documents are large or you have memory limitations, it parses the file as it reads it from disk and the entire file is never stored in memory.

Document Object Model DOM API: This is a World Wide Web Consortium recommendation wherein the entire file is read into memory and stored in a hierarchical tree − based form to represent all the features of an XML document.

SAX obviously cannot process information as fast as DOM can when working with large files. On the other hand, using DOM exclusively can really kill your resources, especially if used on a lot of small files.

SAX is read-only, while DOM allows changes to the XML file. Since these two different APIs literally complement each other, there is no reason why you cannot use them both for large projects.

**Python Web Frameworks**

A web framework is a code library that makes a developer's life easier when building reliable, scalable and maintainable web applications.

## Why are web frameworks useful?

Web frameworks encapsulate what developers have learned over the past twenty years while programming sites and applications for the web. Frameworks make it easier to reuse code for common HTTP operations and to structure projects so other developers with knowledge of the framework can quickly build and maintain the application.

Common web framework functionality.

Frameworks provide functionality in their code or through extensions to perform common operations required to run web applications. These common operations include:

1. URL routing
2. HTML, XML, JSON, and other output format tinplating
3. Database manipulation
4. Security against Cross-site request forgery (CSRF) and other attacks
5. Session storage and retrieval

Not all web frameworks include code for all of the above functionality. Frameworks fall on the spectrum from executing a single use case to providing every known web framework feature to every developer. Some frameworks take the "batteries-included" approach where everything possible comes bundled with the framework while others have a minimal core package that is amenable to extensions provided by other packages.

## Comparing web frameworks

## There is also a repository called [compare-python-web-frameworks](https://github.com/mattmakai/compare-python-web-frameworks) where the same web application is being coded with varying Python web frameworks, tinplating engines and object.

# Web framework resources

* When you are learning how to use one or more web frameworks it's helpful to have an idea of what the code under the covers is doing.
* Frameworks is a really well done short video that explains how to choose between web frameworks. The author has some particular opinions about what should be in a framework. For the most part I agree although I've found sessions and database ORMs to be a helpful part of a framework when done well.
* What is a web framework? Is an in-depth explanation of what web frameworks are and their relation to web servers?
* Jingo vs. Flash vs. Pyramid: Choosing a Python web framework contains background information and code comparisons for similar web applications built in these three big Python frameworks.
* This fascinating blog post takes a look at the code complexity of several Python web frameworks by providing visualizations based on their code bases.
* Python’s web frameworks benchmarks is a test of the responsiveness of a framework with encoding an object to JSON and returning it as a response as well as retrieving data from the database and rendering it in a template. There were no conclusive results but the output is fun to read about nonetheless.
* What web frameworks do you use and why are they awesome? Is a language agnostic Reedit discussion on web frameworks? It's interesting to see what programmers in other languages like and dislike about their suite of web frameworks compared to the main Python frameworks.
* This user-voted question & answer site asked "What are the best general purpose Python web frameworks usable in production?” The votes aren't as important as the list of the many frameworks that are available to Python developers.

## Web frameworks learning checklist

1. Choose a major Python web framework (Jingo or Flask are recommended) and stick with it. When you're just starting it's best to learn one framework first instead of bouncing around trying to understand every framework.
2. Work through a detailed tutorial found within the resources links on the framework's page.
3. Study open source examples built with your framework of choice so you can take parts of those projects and reuse the code in your application.
4. Build the first simple iteration of your web application then go to the [deployment](https://www.fullstackpython.com/deployment.html) section to make it accessible on the web.

**7.2 SOURCE CODE**

import os

import pandas as pd import numpy as np import tensorflow as tf

from flask import Flask, request, render\_template, send\_from\_directory from tensorflow.keras.preprocessing import image

from keras.models import load\_model app = Flask( name )

Food = ['apple\_pie', 'cannoli', 'cheesecake', 'cheese\_plate', 'chicken\_wings', 'chocolate\_cake', 'deviled\_eggs', 'donuts', 'french\_fries', 'frozen\_yogurt', 'ice\_cream', 'macarons']

@app.route("/") def index():

return render\_template("index.html")

@app.route("/about") def about():

return render\_template("about.html")

@app.route("/upload/<filename>") def send\_image(filename):

return send\_from\_directory("images",filename)

@app.route("/upload",methods=["POST","GET"]) def upload():

if request.method=='POST': print("hdgkj")

m = int(request.form["alg"])

myfile = request.files['file'] fn = myfile.filename

mypath = os.path.join("images/", fn) myfile.save(mypath)

print("{} is the file name", fn) print("Accept incoming file:", fn) print("Save it to:", mypath)

if m == 1:

print("bv1")

new\_model = load\_model(r'models/CNN.h5')

test\_image = image.load\_img(mypath, target\_size=(128, 128)) test\_image = image.img\_to\_array(test\_image)

elif m == 2:

print("bv2")

new\_model = load\_model(r'models/SqueezeNet.h5')

test\_image = image.load\_img(mypath, target\_size=(128, 128)) test\_image = image.img\_to\_array(test\_image)

else:

print("bv4")

new\_model = load\_model(r'models/VGG16.h5')

test\_image = image.load\_img(mypath, target\_size=(64, 64)) test\_image = image.img\_to\_array(test\_image)

test\_image = np.expand\_dims(test\_image, axis=0) result = new\_model.predict(test\_image)

preds = Food[np.argmax(result)]

if preds == "apple\_pie":

msg = "237 Calories/100g." \

"Limit these to only special occasions and opt for whole foods and fresh fruit as your usual dessert option"

elif preds == "cannoli":

msg = "254 Calories/100g." \

"Cannoli is an indulgent sweet treat, but if you tweak the recipe, it's easy to make it diabetic-friendly"

elif preds == "cheesecake":

msg = "321 Calories/100g." \

"Cheesecake may sound like a diabetes-friendly option, but the traditional recipe can pack as much as 31.9 g of carbohydrates per slice"

elif preds == "cheese\_plate": msg = "350 Calories/100g." \

"Cheese is safe in moderation for people with diabetes. People with diabetes can safely eat cheese as part of a balanced, healthful diet. As with other foods, moderation is key, and

so a diet that includes too much cheese would be harmful to people with or without diabetes"

elif preds == "chicken\_wings": msg = "203 Calories/100g." \

"Chicken can be a great option for people with diabetes. All cuts of chicken are high in protein and many are low in fat. When prepared in a healthy way, chicken can be a great ingredient in a healthy diabetic eating plan"

elif preds == "chocolate\_cake": msg = "371 Calories/100g." \

"Just because you have diabetes doesn't mean you can't enjoy chocolate cake as part of ahealthy, balanced diet"

elif preds == "deviled\_eggs": msg = "201 Calories/100g." \

"Eggs are a versatile food and a great source of protein. The American Diabetes Association considers eggs an excellent choice for people with diabetes"

elif preds == "donuts":

msg = "452 Calories/100g." \

"Small amounts of sweets can be included in a healthy diet, even if you have diabetes"

elif preds == "french\_fires": msg = "312 Calories/100g." \

"French fries are a food you may want to steer clear of, especially if you have diabetes. Potatoes themselves are relatively high in carbs"

elif preds == "frozen\_yogurt": msg = "159 Calories/100g." \

"Frozen yogurt is a good alternative to ice cream.

Diabetes-friendly desserts are available in most stores and are as easy to prepare at home as any other sweet treat"

elif preds == "ice\_cream":

msg = "207 Calories/100g." \

"Despite what many naysayers will tell you, people with diabetes CAN (and do) eat ice cream. Sure, ice cream can’t compete with, say, a salad when it comes to nutrition"

else:

msg = "404 Calories/100g." \

"Just be sure to keep an eye on your portions. Go for whole wheat pasta, which will increase your fiber, vitamins, and minerals, and reduce any blood sugar spikes when compared to white pasta"

arr = msg.split('.') print(arr)

return render\_template("template.html", text=preds, msg = arr, image\_name=fn)

return render\_template("index.html")

if name == ' main ': app.run(debug=True)

**TEST CASES:**

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Result** |
| Input text | Tested for the classification  of Food image Classification | Success |

# TEST CASES MODEL BUILDING:

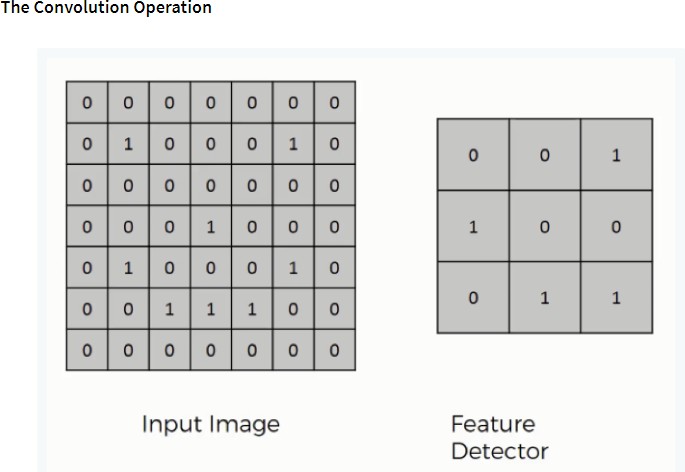
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Test cases** | **I/O** | **Expected O/T** | **Actual O/T** | **P/F** |
| 1 | Read the dataset. | Dataset path. | Dataset need to read  successfully. | Dataset fetched  successfully. | P |
| 2 | Performing pre-processing on the dataset | Pre- processing part takes  place | Pre-processing should be performed on  dataset | Pre- processing successfully  completed. | P |
| 3 | Model Building | Model Building for the clean data | Need to create model using required  algorithms | Model Created Successfully. | P |
| 4 | Classification | Input image provided. | Output should be the Food Image  Classification | Model classified successfully | P |

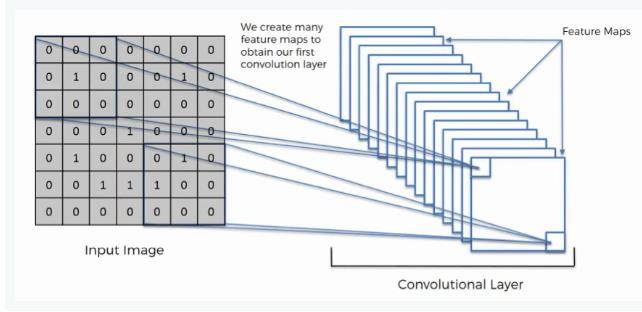
**ALGORITHMS:**

**Convolutional Neural Network**

## Step1: convolutional operation

The first building block in our plan of attack is convolution operation. In this step, we will touch on feature detectors, which basically serve as the neural network's filters. We will also discuss feature maps, learning the parameters of such maps, how patterns are detected, the layers of detection, and how the findings are mapped out.

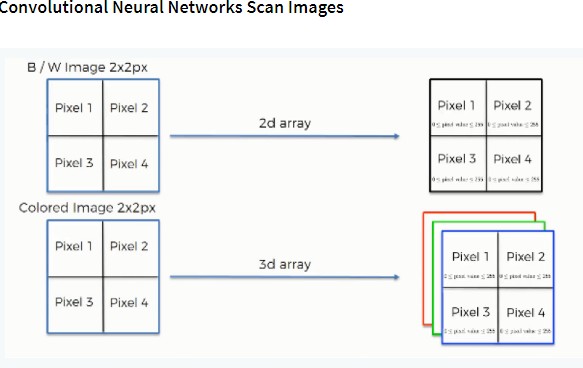




## Step (1b): ReLU Layer

The second part of this step will involve the Rectified Linear Unit or Relook. We will cover Relook layers and explore how linearity functions in the context of Convolutional Neural Networks.

Not necessary for understanding CNN's, but there's no harm in a quick lesson to improve your skills.



## Step 2: Conv2D

Keras Conv2D is 2D Convolution Layer; this layer creates a convolution kernel that is wind with layers input which helps produce a tensor of outputs.

Kernel: In image processing kernel is a convolution matrix or masks which can be used for blurring, sharpening, embossing, edge detection, and more by doing a convolution between a kernel and an image

## Step 3: Flattening

This will be a brief breakdown of the flattening process and how we move from pooled to flattened layers when working with Convolutional Neural Networks.

## Step 4: Full Connection

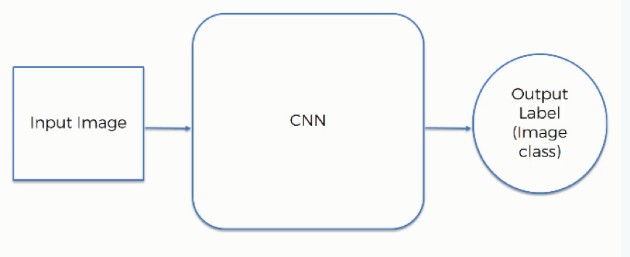
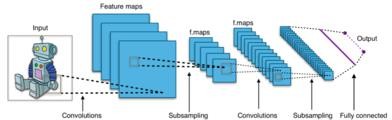
In this part, everything that we covered throughout the section will be merged together. By learning this, you'll get to envision a fuller picture of how ConvolutionalNeural Networks operate and how the "neurons" that are finally produced learn the classification of images

## Summary

In the end, we'll wrap everything up and give a quick recap of the concept covered in the section. If you feel like it will do you any benefit (and it probably will), you should check out the extra tutorial in which Softmax and Cross-Entropy are covered. It's not mandatory for the course, but you will likely come across these concepts when working with Convolutional Neural Networks and it will do you a lot of good to be familiar with them.

## Convolutional neural network (CNN):

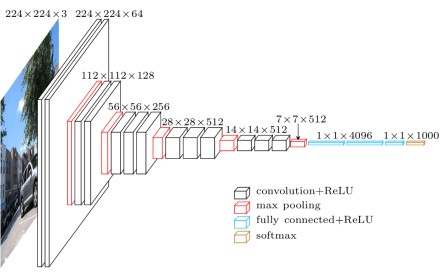
A convolutional neural network consists of an input layer, hidden layers and an output layer. In any feed-forward neural network, any middle layers are called hidden because their inputs and outputs are masked by the activation function and final convolution.



**Fig 2. CNN Architecture**

**Visual Geometry Group (VGG16):**

The VGG network architecture was introduced by Simonyan and Zisserman in their 2014 paper, Very Deep Convolutional Networks for Large Scale Image Recognition. This network is characterized by its simplicity, using only 3×3 convolutional layers stacked on top of each other in increasing depth. Reducing volume size is handled by max pooling. Two fully-connected layers, each with 4,096 nodes are then followed by a SoftMax classifier. The “16” stand for the number of weight layers in the network.



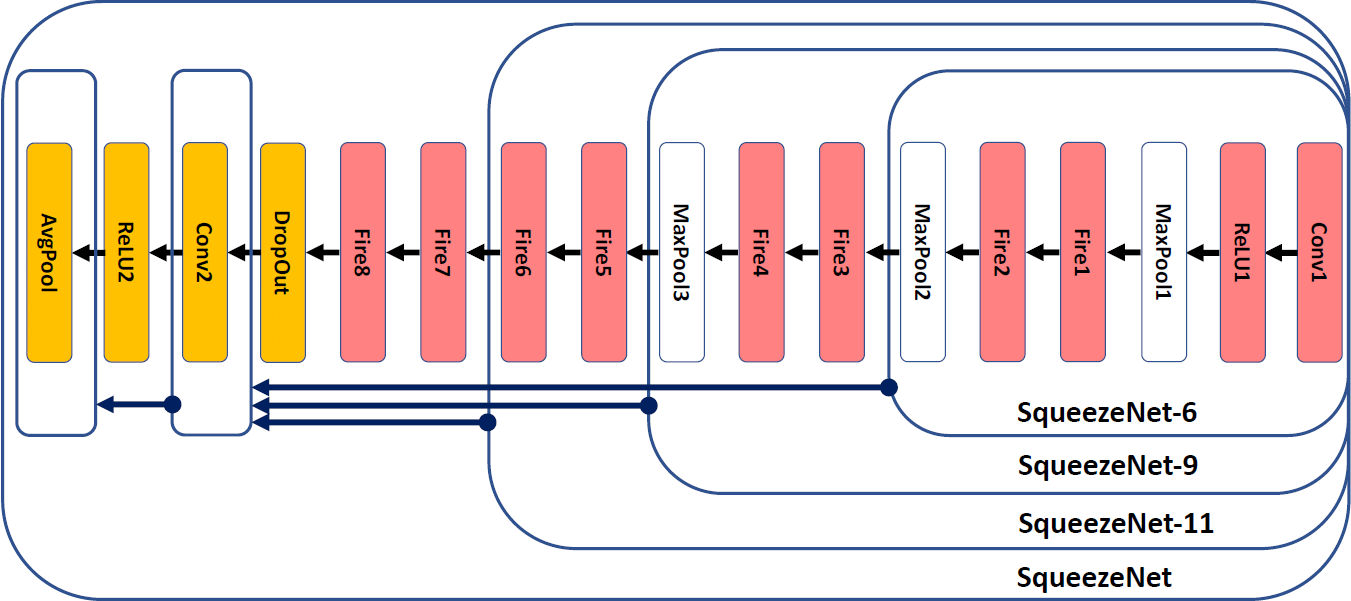
**Fig-3. VGG16 Architecture**

# SqueezeNet:

SqueezeNet provides a smart architecture as well as a quantitative analysis. For the same accuracy of AlexNet, SqueezeNet can be 3 times faster and 500 times smaller.

The main ideas of SqueezeNet are:

* + - Using 1x1(point-wise) filters to replace 3x3 filters, as the former only 1/9 of computation.
    - Using 1x1 filters as a bottleneck layer to reduce depth to reduce computation of the following 3x3 filters.
    - Down sample late to keep a big feature map.



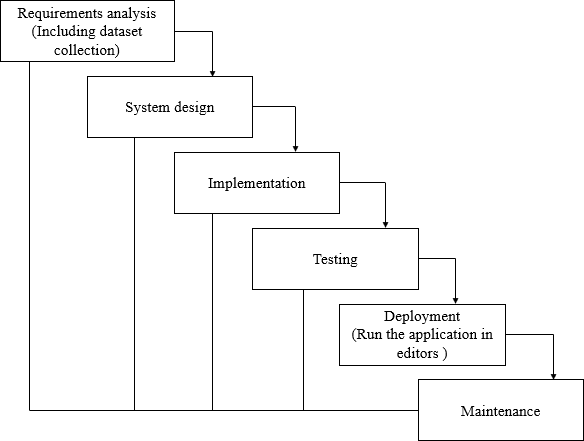
The building brick of SqueezeNet is called fire module, which contains two layers: a squeeze layer and an expand layer. A SqueezeNet stackes a bunch of fire modules and a few pooling layers. The squeeze layer and expand layer keep the same feature map size, while the former reduces the depth to a smaller number, the later increase it. The squeezing (bottleneck layer) and expansion behavior is common in neural architectures. Another common pattern is increasing depth while reducing feature map size to get high level abstract.

The squeeze module only contains 1x1 filters, which means it works like a fully- connected layer working on feature points in the same position. In other words, it doesn’t have the ability of spatial abstract. As its name says, one of its benifits is to reduce the depth of feature map. Reducing depth means the following 3x3 filters in

the expand layer has fewer computation to do. It boosts the speed as a 3x3 filter need as 9 times computation as a 1x1 filter. By intuition, too much squeezing limits information flow; too few 3x3 filters limit space resolution. The following charts provide a quantitative analysis.

## SOFTWARE DEVELOPMENT LIFE CYCLE – SDLC:

In our project we use waterfall model as our software development cycle because of its step-by-step procedure while implementing.



**Fig 3**: Waterfall Model

* **Requirement Gathering and analysis** − all possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
* **System Design** − the requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in

specifying hardware and system requirements and helps in defining the overall system architecture.

* **Implementation** − with inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
* **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
* **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

# 7.3 SYSTEM TESTING

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.

**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 Component.

**FUNCTIONAL TESTING:**

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.

**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.

**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.

**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.

# 8. SCREENSHOTS

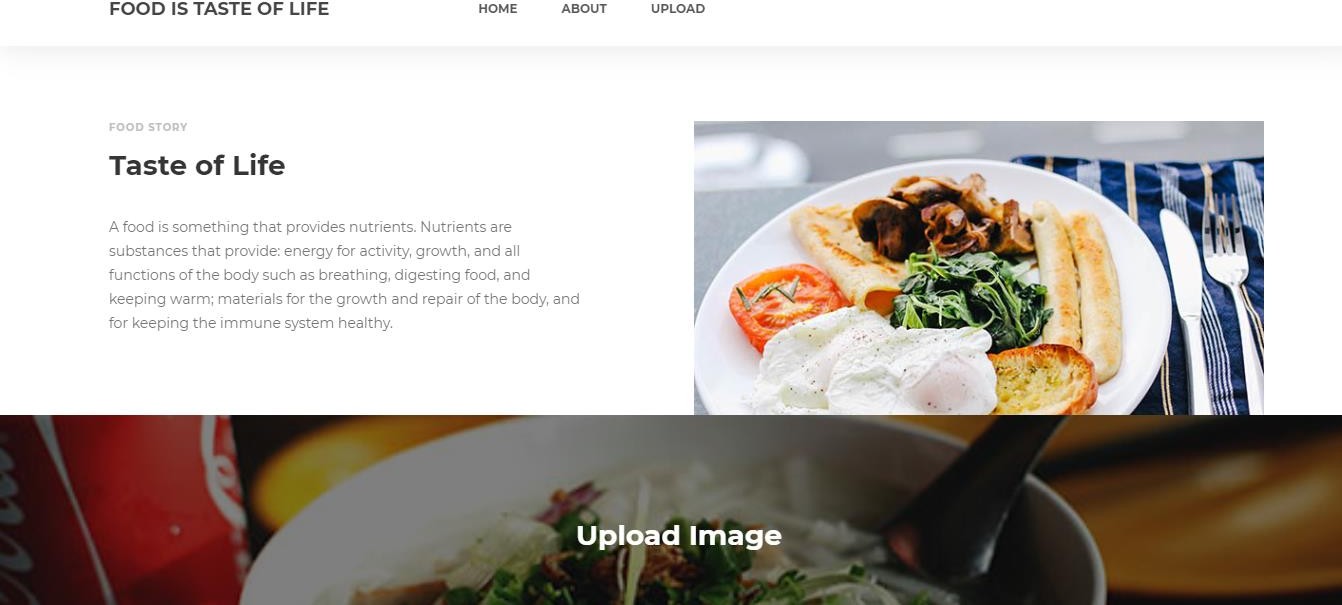
**Home:** In our project, we are classifying the presence of Food Image Classification, with the help of deep learning and Transfer learning.





## Fig: 12.1 Home page

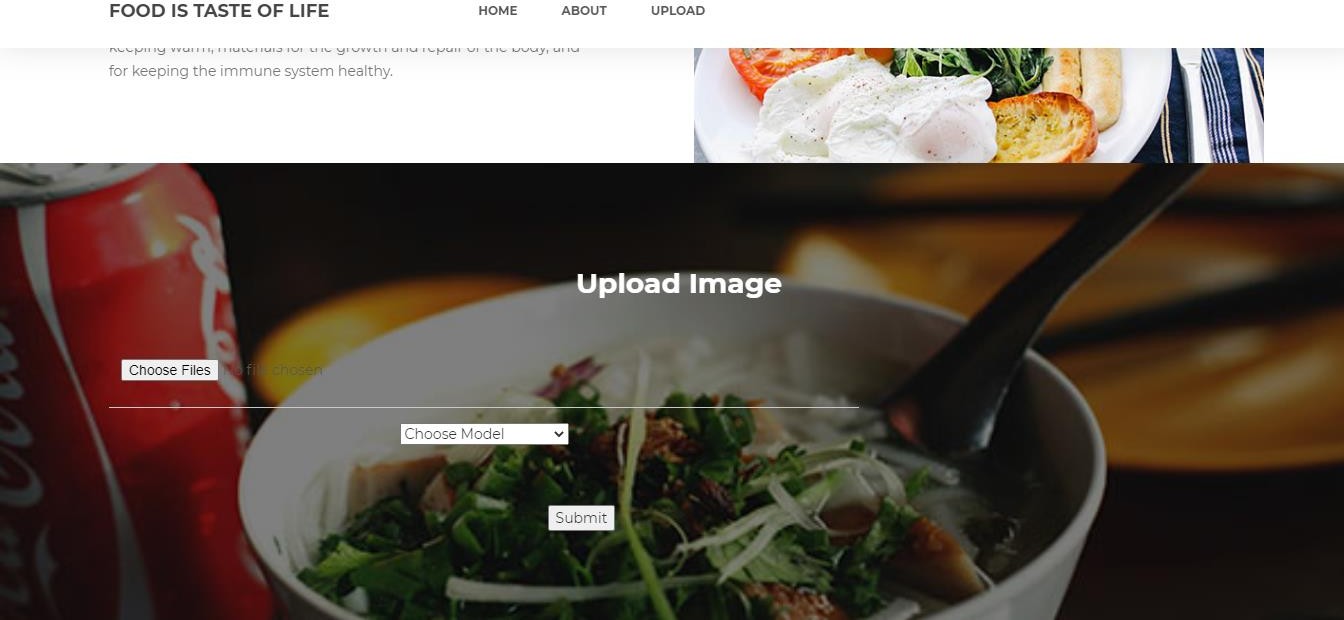
**About Project:** Here the user will get a breif idea about the project.



## Fig:12.2 About project

**Upload Image:**

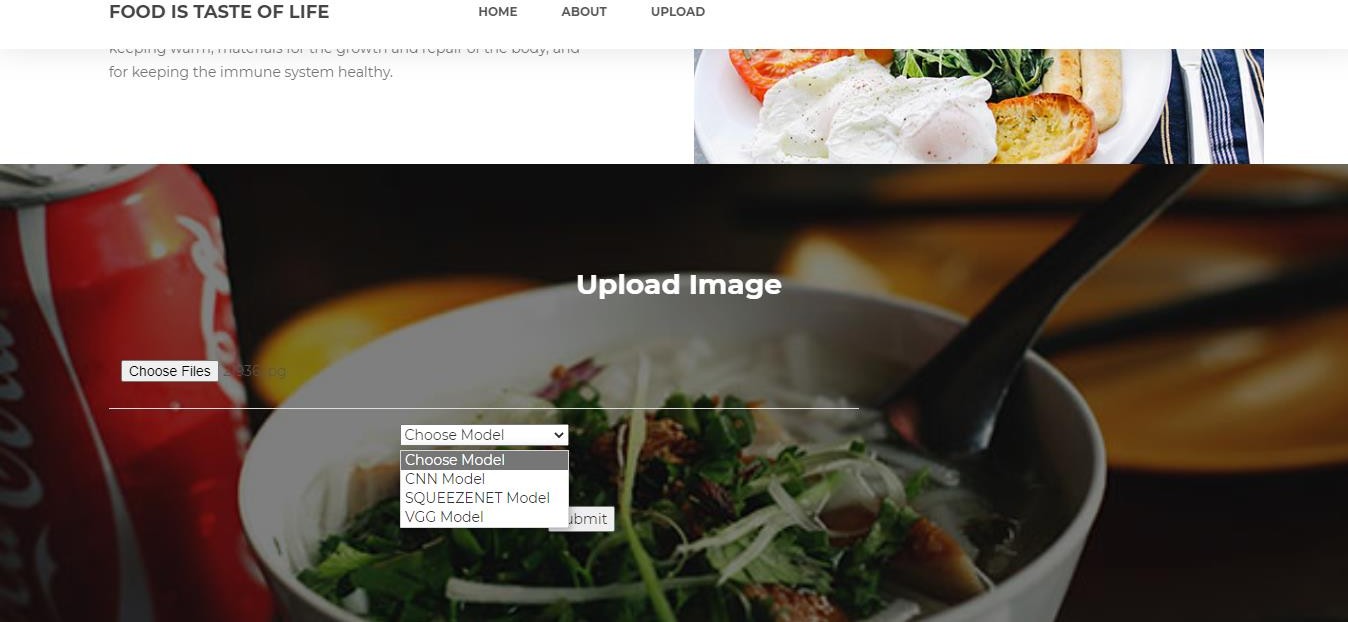
Here the images can be uploaded those which are to be classified.



## Fig: 12.3 Upload Image

**Model choosing:**

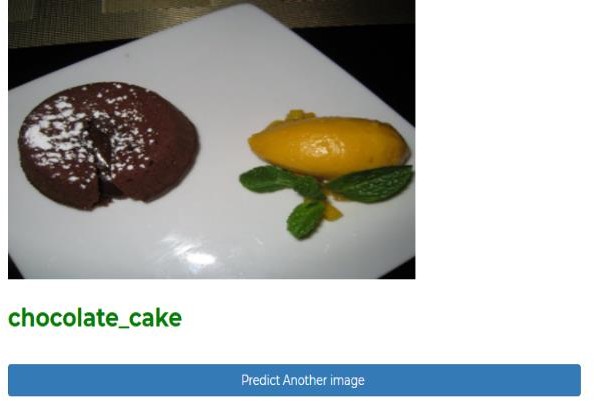
Here the model can be selected, by which the image is to be classified.



## Fig: 12.4 Model Choosing

**Classified output:**

The uploaded image is classified as the Food image Classification.



# 9. CONCLUSION AND FUTURE WORK

**9.1 CONCLUSION:**

In this project we have successfully classified the images of Food image classification, are either affected with the Food items name using the deep learning and Transfer learning. Here, we have considered the dataset of Food101 which will be of different types and different Food items and trained using CNN, along with some SqueezeNet, VGG16 transfer learning method. After the training we have tested by uploading the image and classified it. we proposed deep learning algorithms which are SqueezeNet and VGG 16 Net which are the neural networks for the task of food classification was successfully completed with the better accuracy. The system is able to classify the food contained image. Automated food image classification using deep learning has shown great promise in recent years, and has the potential to revolutionize the food industry. By training deep neural networks on large datasets of food images, researchers have been able to develop highly accurate classifiers that can identify different types of food with remarkable precision

**9.2 FUTURE WORK:**

This can be utilized in future to classify the types of different food items classification with image and names to Predicated to name.

# 10. REFERENCES

1. Zhou, L., Zhang, C., Liu, F., Qiu, Z., & He, Y, “Application of Deep Learning in Food: A Review,” Comprehensive Reviews in Food Science and Food Safety, vol. 18, pp. 1793-1811, 2019.
2. Farinella, G. M., Moltisanti, M., &Battiato, S., “Classifying food images represented as Bag of Textons,” IEEE International Conference on Image Processing (ICIP), Paris, pp. 5212-5216, doi: 10.1109/ICIP.2014.7026055, 2014.
3. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., & Oliva, A., “Learning deep features for scene recognition using places database,” Proceedings of the 27th International Conference on Neural Information Processing Systems, vol. 1, pp. 487-495, ACM, 2014.
4. Rahmani, G. A.,“Efficient Combination of Texture and Color Features in a New Spectral Clustering Method for PolSARImageSegmentation/National Academy Science Letters, vol. 40, pp. 117-120, 2017, https://doi.org/10.1007/s40009- 016-0513-6.
5. Wang, M., Wan, Y., Ye, Z., & Lai, X.,“Remote sensing imageclassification based on the optimal support vector machine andmodified binary coded ant colony optimization algorithm,’Information Sciences, vol. 402, pp. 50-68, 2017, https://doi.org/10.1016Zj.ins.2017.03.027.
6. Xia, J., Ghamisi, P., Yokoya, N., & Iwasaki, A., “Random Forest Ensembles and Extended Multiextinction Profiles for Hyperspectral Image Classification,” IEEE Transactions on Geoscience and Remote Sensing, vol. 56 , pp. 202-216, 2018, doi:10.1109/TGRS.2017.2744662.
7. Kaymak, S., Helwan, A., &Uzun, D., “Breast cancer image classification using artificial neural networks,” Procedia Computer Science, vol. 120, pp. 126-131, 2017, https://doi.org/10.1016/j.procs.2017.11.219.
8. Chaib, S., Liu, H., Gu, Y., & Yao, H., “Deep Feature Fusionfor VHR Remote Sensing Scene Classification,” IEEE Transactions on Geoscience and Remote Sensing, vol. 55, pp. 4775-4784, 2017, doi:10.1109/TGRS.2017.2700322.
9. Simard, P. Y., Steinkraus, D., & Platt, J. C., “Best Practices for Convolutional Neural Networks,” 12th International Conference on Document Analysis and Recognition, vol. 2. IEEE Computer Society, 2003.
10. Bazargani, Anjos, M. &., Lobo, A. &Mollahosseini, F. &,Shahbazkia, A. &, & Hamid, “Affine Image Registration Transformation Estimation Using a Real Coded,” Proceedings of the 14th annual conference companion on Genetic and evolutionary computation, pp. 1459-1460, ACM, 2012, https://doi.org/10.1145/2330784.2330990).
11. Keutzer, F. N., “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size,” ICLR, 2016.
12. Kurama, V. (2020, June 5). A Review of Popular Deep Learning Architectures: ResNet, InceptionV3, and SqueezeNet. Retrieved January 25, 2021, June 5, 2020 from Paper space Blog: https://blog.paperspace.com/popular-deep-learning-architecturesresnetinceptionv3 Squeezenet/#:~:text= The%20

SqueezeNet%20architecture%20is%20comprised,3%20%C3%97%203%20

convolution%20 filters. &text=Meanwhile%2C%20a%20Deep%20Compression%20SqueezeNet,and%20a.

1. Simonyan, K., & Zisserman, A., “Very Deep Convolutional Networks for Large-Scale Image Recognition,” ICLR. arXiv, 1409.1556, 2015.
2. Ghazia, M. M., Yanikoglu, B., &Aptoula, E., “Plant identification using deep neural networks via optimization of transfer learning parameters,” Neurocomputing,vol.235,pp.228-235, 2017, https://doi.org/10.1016/j.neucom.2017.01.018.