POTATO LEAF ANALYSIS

System Requirement Specification (SRS)

Project work Phase 1 (EAI753)

BACHELOR OF TECHNOLOGY (CSE-AI)

PROJECT GUIDE:	SUBMITTED BY:
Mr. Ashish Bishnoi	**Aniket Sinha (TCA1959047)**
Dr. Saurabh Pathak	**Dwiz (TCA1959045)**

Dec, 2022

FACULTY OF ENGINEERING & COMPUTING SCIENCES

TEERTHANKER MAHAVEER UNIVERSITY, MORADABAD

Table of Contents

1 Introduction

POTATO LEAF ANALYSIS

1.1 Problem Statement

Potatoes contain vitamins, minerals, and fiber. They're rich in vitamin C, which is an antioxidant. Potatoes were a life-saving food source in early times because the vitamin C prevented scurvy. Another major nutrient in potatoes is potassium, an electrolyte which aids in the workings of our heart, muscles, and nervous system. Potato skin contains fiber, which is important for digestive health. All of these shows the importance of potato in human's food and their nutrients.

But there is huge loss of farmer in potato farming because of potato diseases and unable to timely treatment of know potato diseases, which may lead to shortage of potato in current market and the huge loss for farmers directly lead to less farming of potato plants. Some of the most important diseases worldwide are late blight (*Phytophthora infestans*), early blight (*Alternaria solani*), stem canker (*Rhizoctonia solani*), potato wart (*Synchytrium endobioticum*), powdery scab (*Spongospora subterranea*), bacterial wilt (*Ralstonia solanacearum*), black leg (*Pectobacterium* spp.)

By understanding the importance of potato in human food and the huge loss of potato in current market because of unable to timely treatment of potato diseases, it's become important to resolve these problems by analysis the type of potato diseases in early stage so that it can easily cure from the diseases before it destroys the plant.

1.2 Scope of the project

Potato leaf analysis is important for fast growing potato farming in this modern world, as we all know Potato is consumed by more than 90% of world's population on regular basis.

It is very important that the potato which farmer grows must be healthy and disease free, so here our project comes into the picture where we detect the quality of potato leaf to check whether it is good or not on early stage which can save farmers from very heavy losses and can cure their crops by using right fertilizers on right time.

Need: Client Framers needs web application and Android app that allows customers to discover the type of diseases of potatoes.

Scope: we will create the app for farmers that includes all necessary algorithms and data to provide users with the right diseases for their needs, as well as functionality for those users to check the disease and make cure of the disease through the app. This includes complex data regarding disease, environment, budgets and user types.

Deliverables:

- Functional application with comprehensive data on what cure is appropriate for different type of potatoes diseases.
- A profile page for each type of diseases with a list of appropriate cure and percentage of damage in the plant.
- User profile functionality so users can save recommendations and favorites.

Exclusions:

- Client's website is not included in the scope of the project, so all links should be provided by the client.
- Logo and other graphic design elements related to the client's business

are excluded unless directly necessary for the app only.

Assumptions: we have assumed the client has already developed relevant logos, graphic design, font choices, key colors and created any relevant web pages needed.

1.3 Definitions, Acronyms, and Abbreviations

Tool Name	**Version**	**Purpose**
Python	*3.7*	*Primary programming Language*
Google colab	*1.0.0*	*Editor or IDLE*
Tensorflow	*V2.10.0*	*Open-source Library for ML and DL*
React & JS	*18.2.0*	*Frontend*
GCP	*1.14.4*	*Could service platform*
Sklearn	*.21.0*	*Machine learning library*
Reactive Native	*0.70*	*Mobile Application*
FastAPI	*.10.0.1*	*Used for API sharing and creation*
Matplotlib	*3.0*	*Used for graph and plotting*
Pandas	*1.3.5*	*Data analysis tool*

1.4 References

- *https://www.mdpi.com/2079-9292/10/17/2064/pdf#:~:text=A%20novel%20deep%20learning%20technique%20called%20Potato%20Leaf%20Disease%20Detection,diseases%20from%*

20potato%20leaf%20images.&text=The%20proposed%20method%20has%20an,of%2D%20the%2Dart%20models.

https://www.analyticsvidhya.com/blog/2021/12/end-to-end-potato-leaf-disease- prediction-project-a-complete-guide/

- *https://www.researchgate.net/publication/342325881_Potato_Leaf_Diseases_Detection_Using_Deep_Learning*

2 Project Description

- Agriculture is one of the essential sectors for the survival of humankind. At the same time, digitalization touching across all the fields that became easier to handle various difficult tasks. Adapting technology as well as digitalization is very crucial for the field of agriculture to benefit the farmer as well as the consumer.
- Due to adopting technology and regular monitoring, one can able to identify the diseases at the very initial stages and those can be eradicated to obtain a better yield of the crop. In this document, a methodology was proposed for the detection as well as the classification of diseases that occur for the potato plants.
- We are developing the web application and also android based mobile application for the farmers, so that they can easily detect the type of diseases occur in the potato plant. Our application is mainly developed to detect or classify the early blight and late blight diseases in the plant including the healthy status of different potato plants. Because these two types of disease contribute the major loss in the potato market.

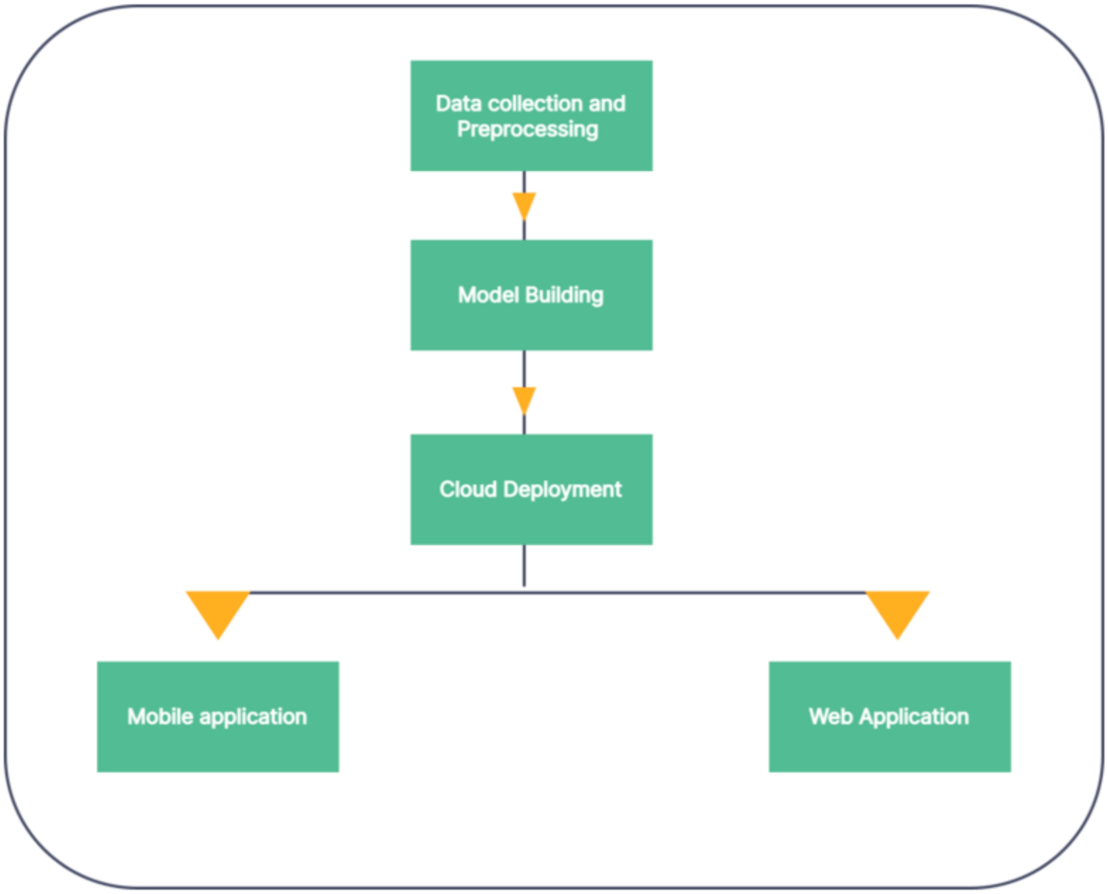
2.1 Scope of the work

- we will create the app for farmers that includes all necessary algorithms and data to provide users with the right diseases for their needs, as well as functionality for those users to check the disease and make cure of the disease through the app. This includes complex data regarding disease, environment, budgets and user types.
- Both Web application and mobile application can able to analysis the potato leaf health status to classify weather it is healthy or not.
- Functional application with comprehensive data on what cure is appropriate for different type of potatoes diseases
- The application is built for only early blight and late blight diseases.
- User just need to click the photo of the potato leaf and application automatically upload the photo for the analysis.
- The model is deployed on the google cloud platform and its API will used in both applications.

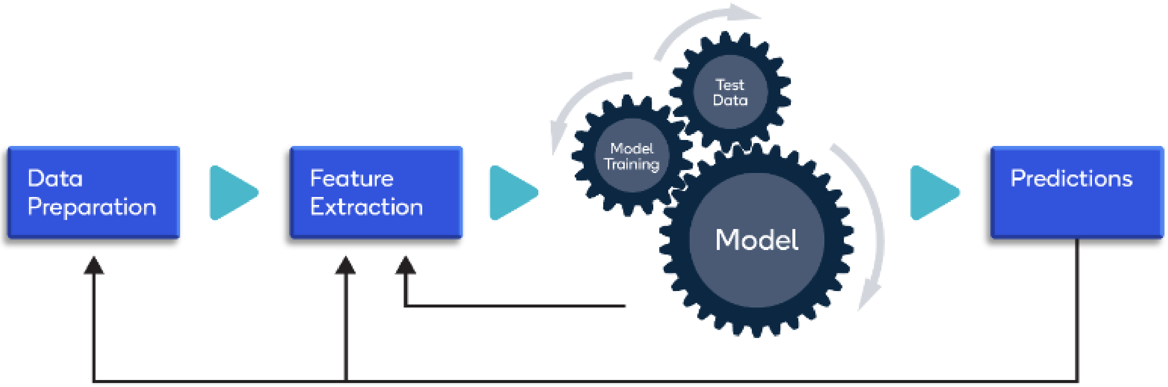
2.2 Project Modules

There is main 5 module in this project:

1. Data Module
2. Model module
3. Cloud module
4. Mobile application module
5. Web application module



1. **Data Module: -** Before training any machine learning model it's important to collect the necessary data. In this module the initial data collection and data preprocessing has been done. Data processing helps to feature selection and finding the correlations between different attributes of dataset.



Within the data preparation stage are the data collection and data pre-processing stages.

Data collection

Collecting data for training the ML model is the basic step in the machine learning pipeline. The predictions made by ML systems can only be as good as the data on which they have been trained. Following is some of the problems that can arise in data collection:

- Inaccurate data: - The collected data could be unrelated to the problem statement.
- Missing data: - Sub-data could be missing. That could take the form of empty values in columns or missing images for some class of prediction.
- Data imbalance: - Some classes or categories in the data may have a disproportionately high or low number of corresponding samples. As a result, they risk being under-represented in the model.
- Data bias: - Depending on how the data, subjects and labels themselves are chosen, the model could propagate inherent biases on gender, politics, age or region, for example. Data bias is difficult to detect and remove.

Several techniques can be applied to address those problems: -

- Pre-cleaned, freely available datasets. If the problem statement (for example, image classification, object recognition) aligns with a clean, pre-existing, properly formulated dataset, then take advantage of existing, open-source expertise.
- Web crawling and scraping. Automated tools, bots and headless browsers can crawl and scrape websites for data.
- Private data. ML engineers can create their own data. This is helpful when the amount of data required to train the model is small and the problem statement is too specific to generalize over an open-source dataset.
- Custom data. Agencies can create or crowdsource the data for a fee.

Data pre-processing

Real-world raw data and images are often incomplete, inconsistent and lacking in certain behaviours or trends. They are also likely to contain many errors. So, once collected, they are pre-processed into a format the machine learning algorithm can use for the model.

Pre-processing includes a number of techniques and actions:

- Data cleaning: - These techniques, manual and automated, remove data incorrectly added or classified.
- Data imputations: - Most ML frameworks include methods and APIs for balancing or filling in missing data. Techniques generally include imputing missing values with standard deviation, mean, median and k-nearest neighbors (k-NN) of the data in the given field.

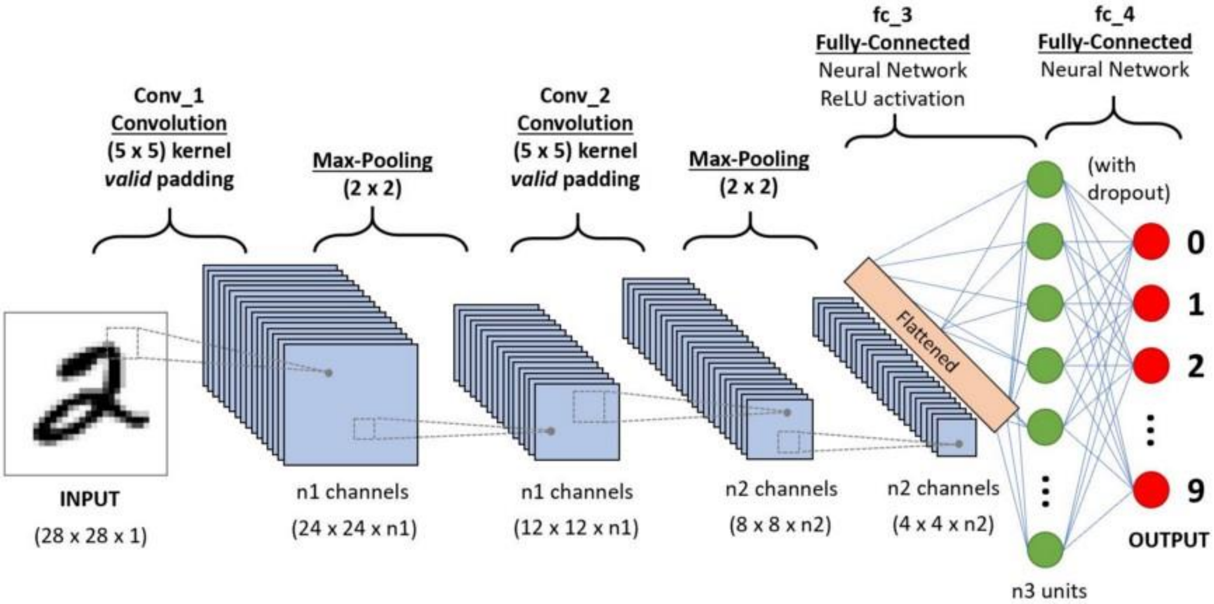
- Oversampling: - Bias or imbalance in the dataset can be corrected by generating more observations/samples with methods like repetition, bootstrapping or Synthetic Minority Over-Sampling Technique (SMOTE), and then adding them to the under-represented classes.
- Data integration: -Combining multiple datasets to get a large corpus can overcome incompleteness in a single dataset.
- Data normalization: - The size of a dataset affects the memory and processing required for iterations during training. Normalization reduces the size by reducing the order and magnitude of data.

2. **Model Module: -** In this module we created, trained and save our deep learning model using pickle library. This module takes care of splitting of data, model training where we are using convolutional neural network algorithm and measuring the accuracy of model is also done in this module.

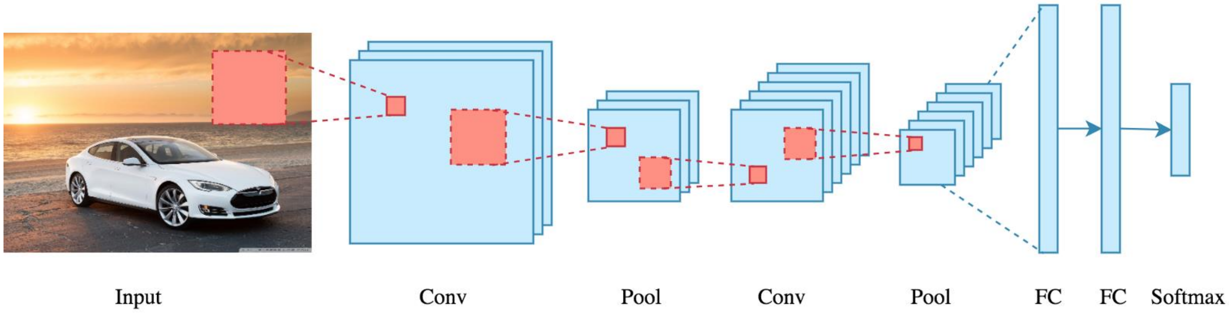
   In the past few decades, Deep Learning has proved to be a very powerful tool because of its ability to handle large amounts of data. The interest to use hidden layers has surpassed traditional techniques, especially in pattern recognition. One of the most popular deep neural networks is Convolutional Neural Networks in deep learning.

Background of CNNs

CNN’s were first developed and used around the 1980s. The most that a CNN could do at that time was recognize handwritten digits. It was mostly used in the postal sectors to read zip codes, pin codes, etc. The important thing to remember about any deep learning model is that it requires a large amount of data to train and also requires a lot of computing resources. This was a major drawback for CNNs at that period and hence CNNs were only limited to the postal sectors and it failed to enter the world of machine learning.



In deep learning, a **convolutional neural network** (**CNN/ConvNet**) is a class of deep neural networks, most commonly applied to analyse visual imagery. Now when we think of a neural network, we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics **convolution** is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

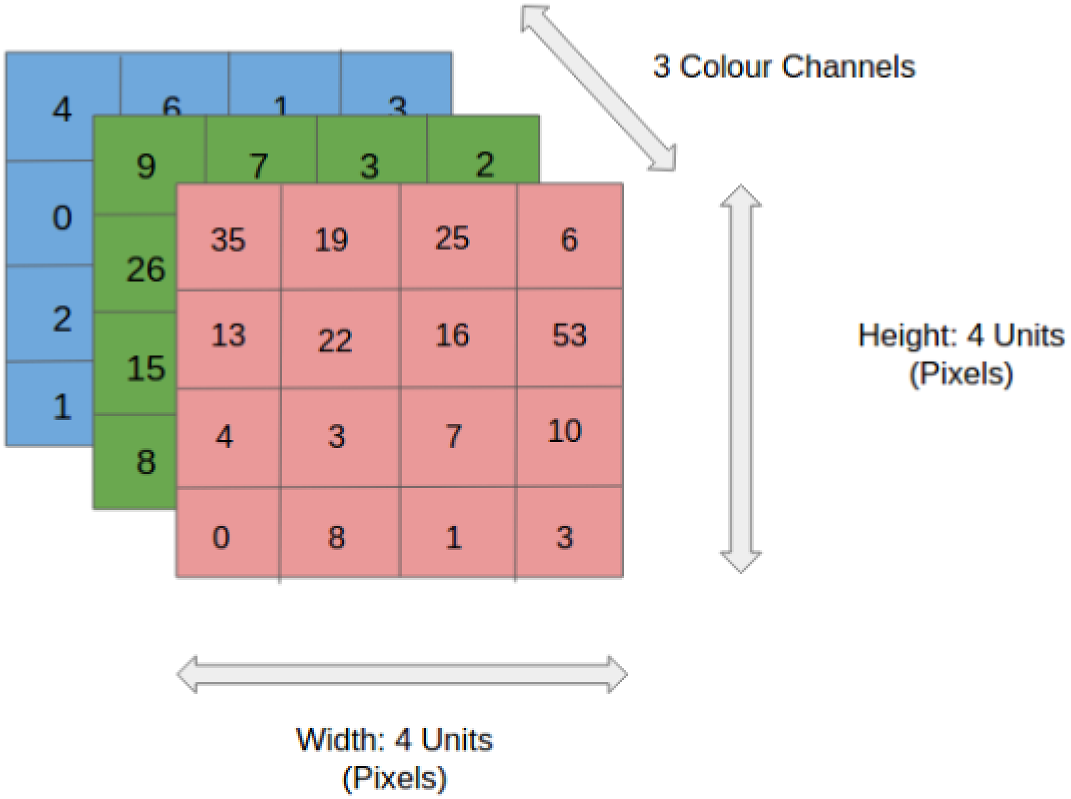


But we don't really need to go behind the mathematics part to understand what a CNN is or how it works.

Bottom line is that the role of the ConvNet is to reduce the images into a form that is easier to process, without losing features that are critical for getting a good prediction.

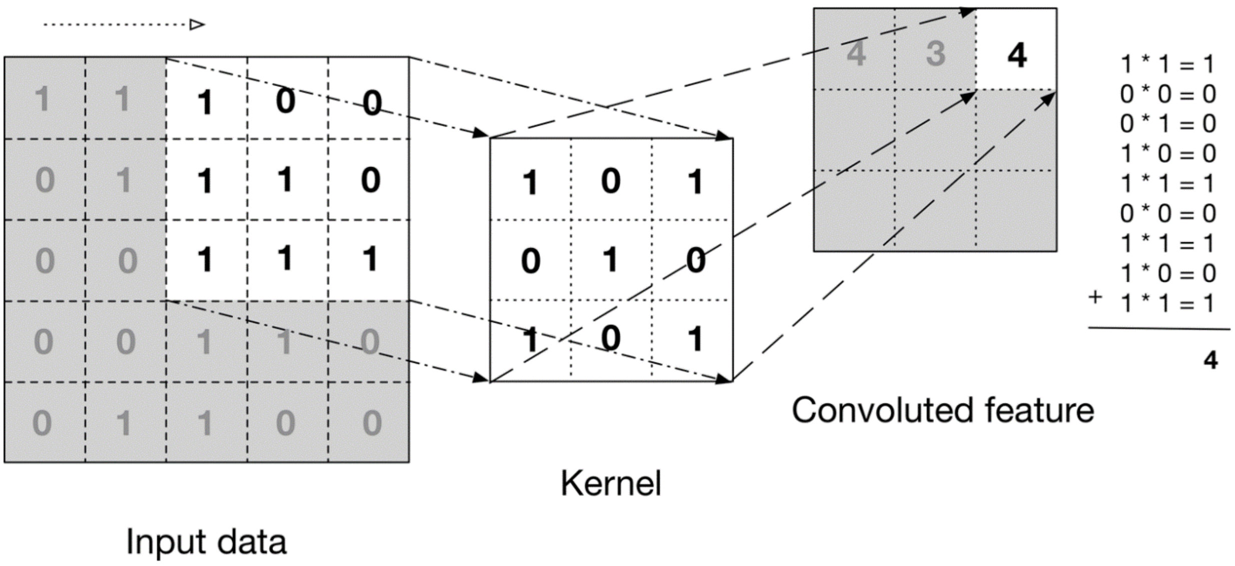
How does it work?

Before we go to the working of CNN's let's cover the basics such as what is an image and how is it represented. An RGB image is nothing but a matrix of pixel values having three planes whereas a grayscale image is the same but it has a single plane. Take a look at this image to understand more.

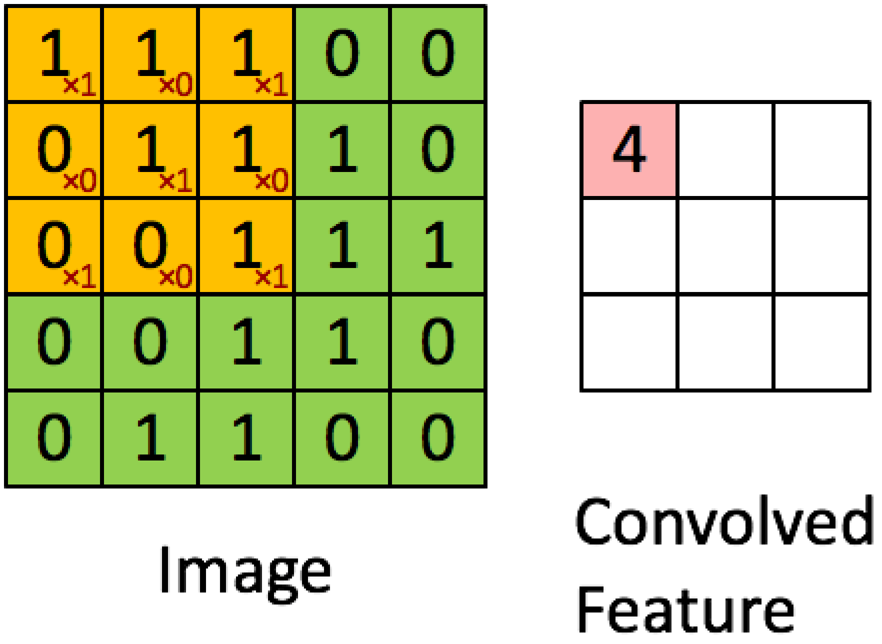


35	19	25	6
13	22	16	53
4	3	7	10
0	8	1	3

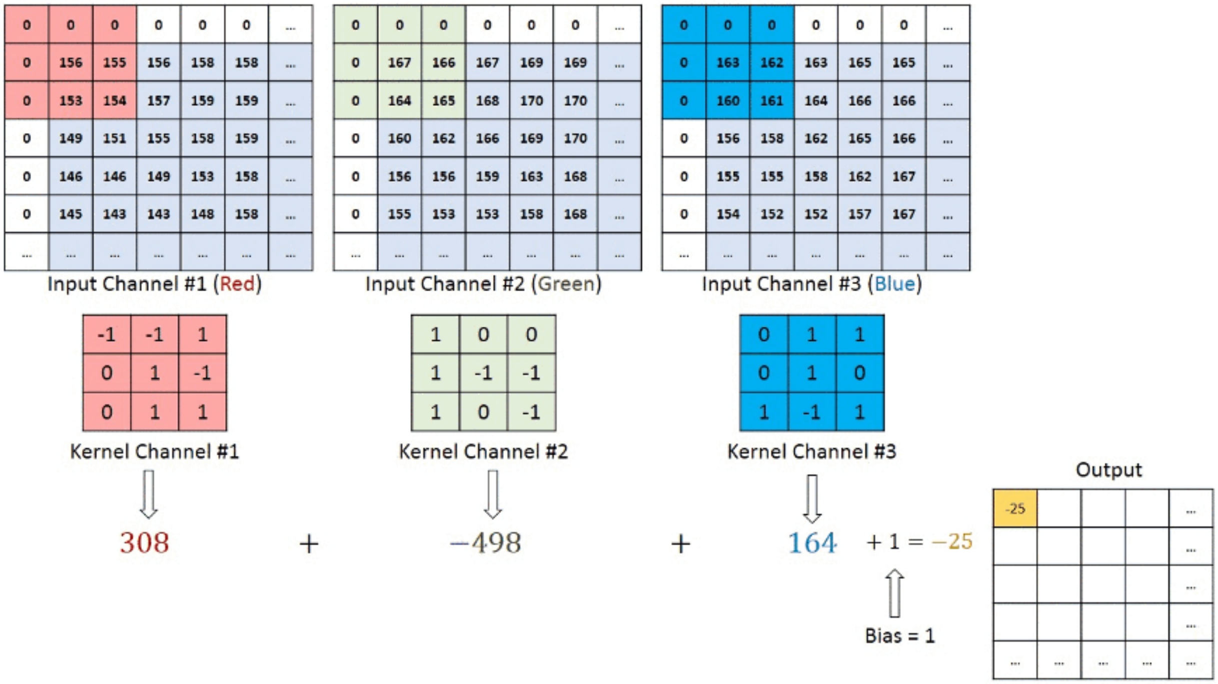
For simplicity, let's stick with grayscale images as we try to understand how CNNs work.



The above image shows what a convolution is. We take a filter/kernel(3×3 matrix) and apply it to the input image to get the convolved feature. This convolved feature is passed on to the next layer.

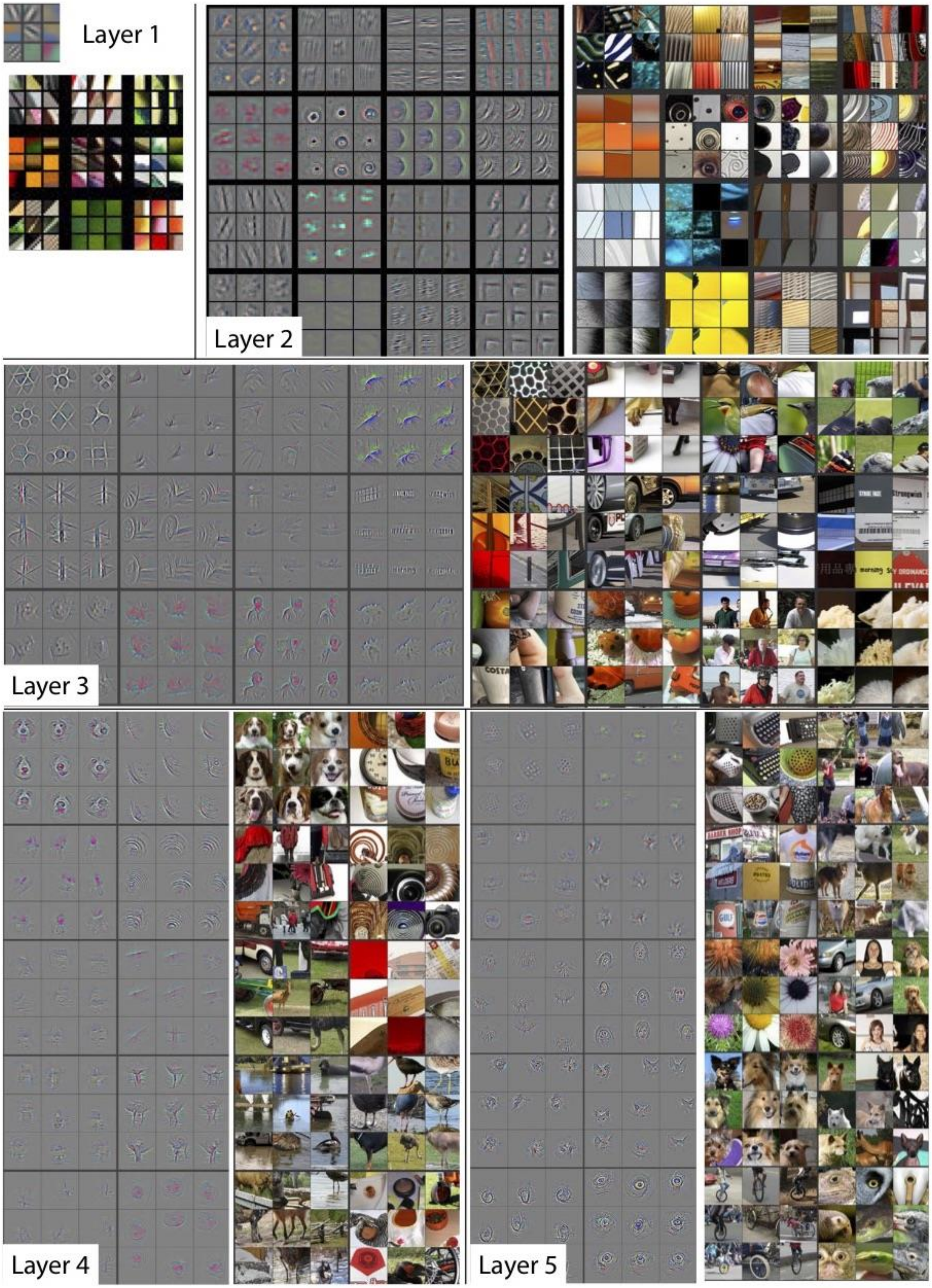


In the case of RGB color, channel take a look at this animation to understand its working

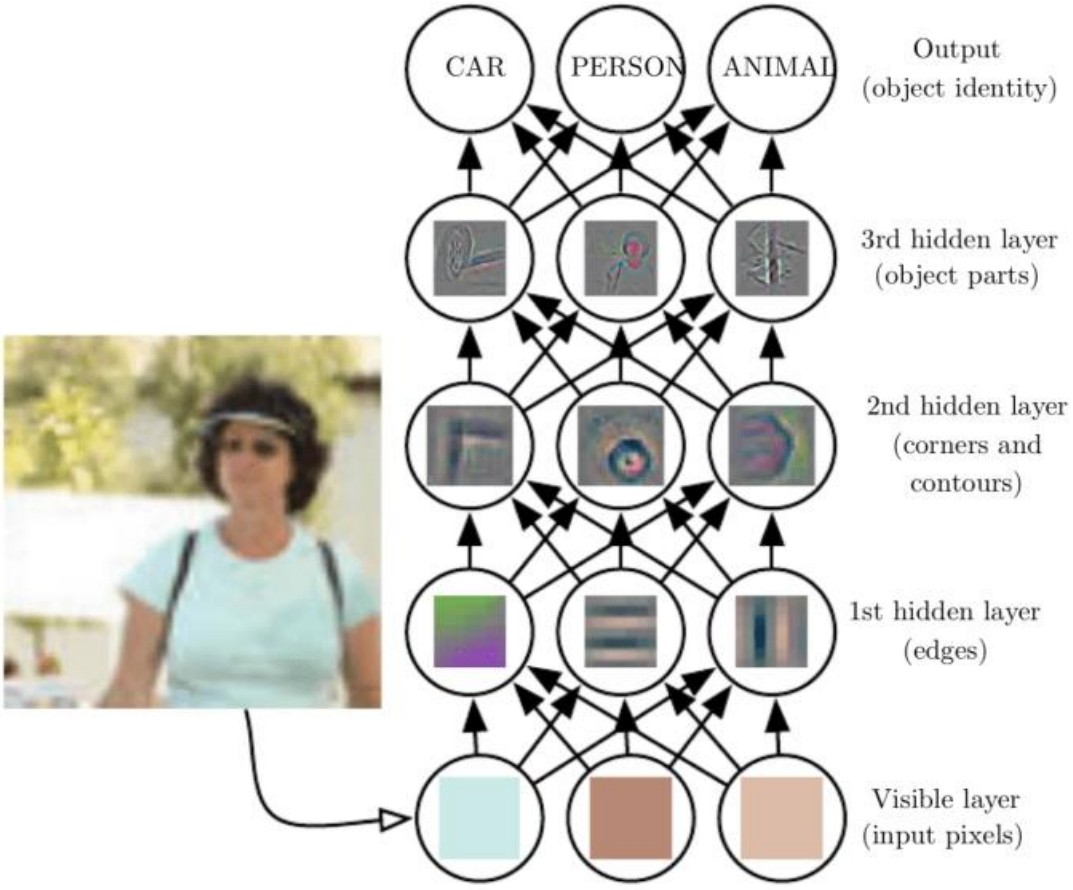


Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts, are mathematical functions that calculate the weighted sum of multiple inputs and outputs an activation value. When you input an image in a ConvNet, each layer generates several activation functions that are passed on to the next layer.

The first layer usually extracts basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. As we move deeper into the network it can identify even more complex features such as objects, faces, etc.

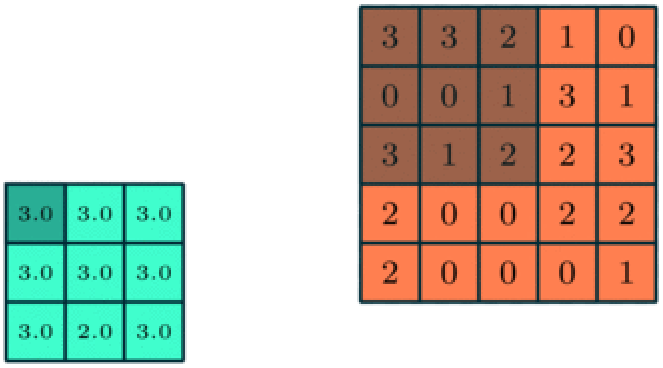

Layer 1
Layer 2
Layer 3
Layer 4
Layer 5

Based on the activation map of the final convolution layer, the classification layer outputs a set of confidence scores (values between 0 and 1) that specify how likely the image is to belong to a "class." For instance, if you have a ConvNet that detects cats, dogs, and horses, the output of the final layer is the possibility that the input image contains any of those animals.



<u>Pooling layer</u>: -

Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to **decrease the computational power required to process the data** by reducing the dimensions. There are two types of pooling average pooling and max pooling. I've only had experience with Max Pooling so far, I haven't faced any difficulties.

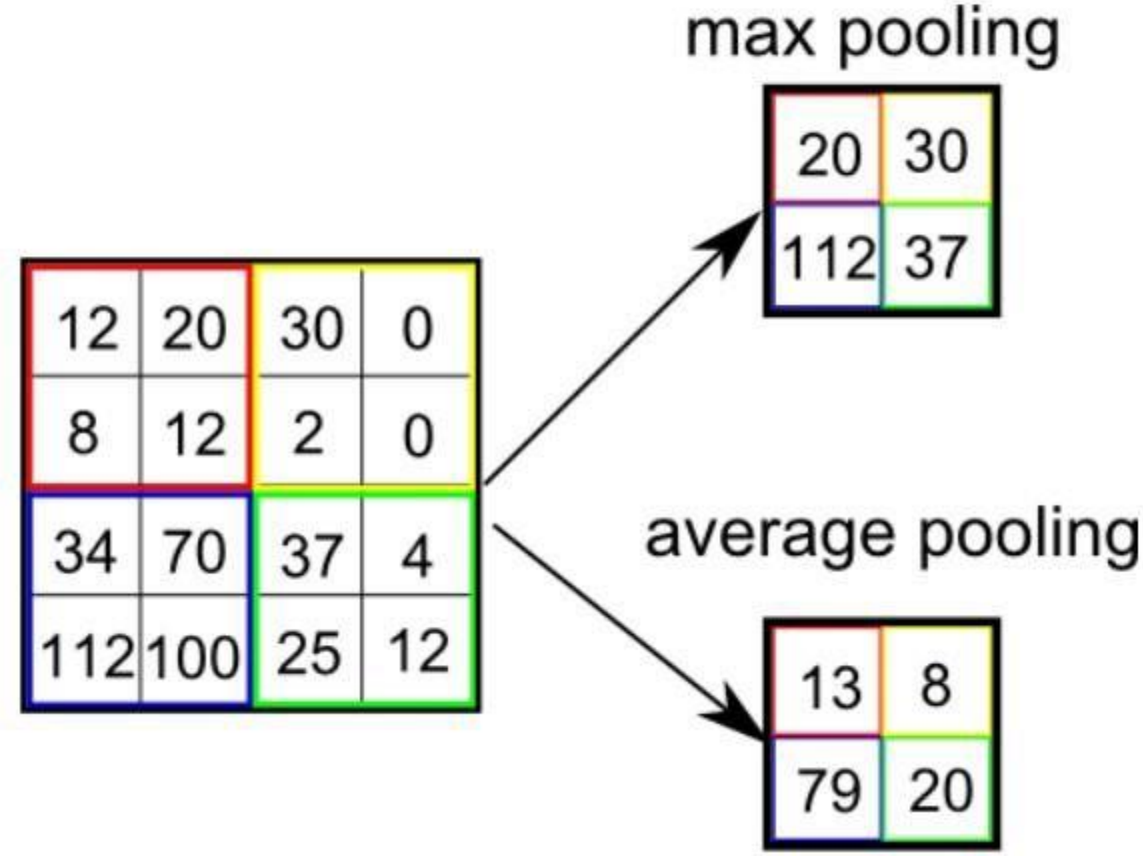


3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

So what we do in Max Pooling is we find the maximum value of a pixel from a portion of the image covered by the kernel. Max Pooling also performs as a **Noise Suppressant**. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction.

On the other hand, **Average Pooling** returns the **average of all the values** from the portion of the image covered by the Kernel. Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that **Max Pooling performs a lot better than Average Pooling**.



Limitations

Despite the power and resource complexity of CNNs, they provide in-depth results. At the root of it all, it is just recognizing patterns and details that are so minute and inconspicuous that it goes unnoticed to the human eye. But when it comes to **understanding** the contents of an image it fails.

Let’s take a look at this example. When we pass the below image to a CNN it detects a person in their mid-30s and a child probably around 10 years. But when we look at the same image we start thinking of multiple different scenarios. Maybe it’s a father and son day out, a picnic or maybe they are camping. Maybe it is a school ground and the child scored a goal and his dad is happy so he lifts him.

These limitations are more than evident when it comes to practical applications. For example, CNN's were widely used to moderate content on social media. But despite the vast resources of images and videos that they were trained on it still isn't able to completely block and remove inappropriate content. As it turns out it flagged a 30,000-year statue with nudity on Facebook.

Cloud Module: - This module is the created to deployed the save model over the cloud so that we can use the API of the save model in multiple application and it also provide the ease of backend server. By using the cloud technology, we don't need to worry about backend deployment.

Application Module: - This module divided into two-part mobile application and web application in which mobile application take care android application and web application is used for website. Mobile application contains the design, implementation of functions in mobile application including the API integration in backend, where web application module has the frontend of website, react & JavaScript is used for frontend and in backend FastAPI is used of API integration.

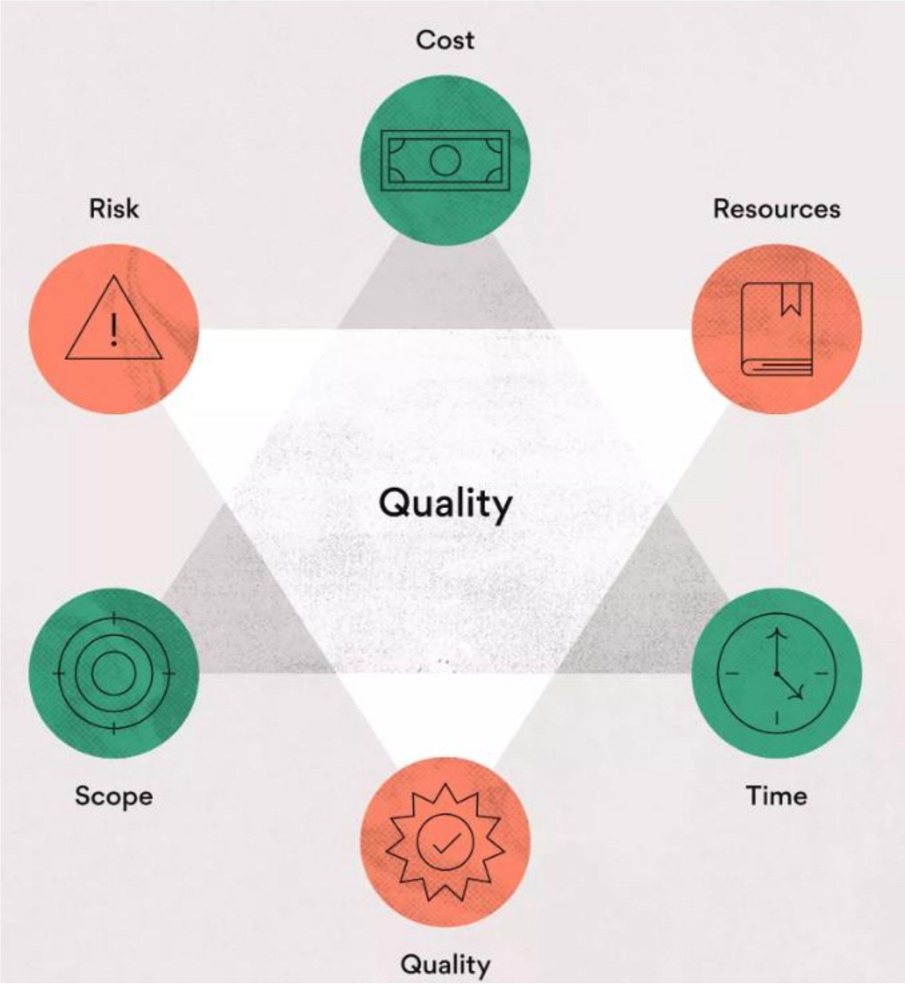
2.3 User Characteristics

Potato leaf analysis system is intended for three types of users: -

1. **Farmers**: - These are our major users for which we building this application, they have access to input the image and get the predicted output including some others required operations like review and feedback.
2. **Doctors**: - These our second major user. The user interface of these is same as farmer but they might have some extra feature as compare to others user.

3. **The admin**: - These are that users which have full access of web application and have the right to make change in the web application.

2.4 Constraints of project



Scope

Project scope refers to a project's magnitude in terms of quality, detail, and deliverables. Time and money are dependencies of project scope, because as the project scope grows, the project will require more time and money to complete.

You'll need to be aware of scope creep throughout each project phase and work hard to prevent it. You can prevent scope creep by creating detailed project plans and getting project stakeholders to sign off on everything before production begins.

Cost

Cost constraints include the project budget as a whole and anything of financial value required for your project. Items that may be a cost constraint include:

- Project cost
- Team member salaries
- Cost of equipment
- Cost of facilities
- Repair costs
- Material costs

Include any items in this section that require you to pull from your company's financial resources.

Time

Time management is essential for project success, and there are various time constraints you'll face during each phase of your project. When you try to increase your project timeline, there will be consequences like extended deadlines, adjustments to the team calendar, or less time for planning.

Time elements in your project that can lead to constraints may include:

- Overall project timeline
- Hours worked on project
- Internal calendars and goalposts
- Time allotted for planning and strategy
- Number of project phases

Scope, cost, and time are called the iron triangle because these three constraints can be difficult to manoeuvre around each other while maintaining project quality. For example, if you cut your budget or increase your scope, you'll likely need to compensate by loosening up on time. You can do this by extending deadlines, adding hours, or adjusting your project schedule.

Risk

Project risks are any unexpected occurrences that can affect your project. While most project risks are negative, some can be positive. For example, a new technology may be released while your project is in progress. This technology may help you finish your project quicker or it may cause more competition in the market and reduce your product value.

You can determine project risks using risk analysis and risk management strategies to keep them at bay. Some risks you may face include:

- Stretched resources
- Operational mishaps
- Low performance
- Lack of clarity
- Scope creep
- High costs
- Time crunch

Use a risk register to assess the likelihood and severity for each project risk, then mitigate the most likely and severe risks first.

Resources

Resources tie closely with cost constraints on your project because these project requirements cost money. Without proper resource allocation, can experience lower project quality, an increased budget, and timeline delays.

Some resources to consider include:

- People
- Equipment or materials
- Facilities

- Software

Quality

Project quality is also its own constraint because there are aspects of the project that can result in poor quality that aren't necessarily related to cost, time, resources, risk, or scope. These include:

- Lack of communication
- Poor design or development skills
- Too many projects' changes

2.5 Assumptions and Dependencies

Data:

- Feature expectations are captured in a schema.
- All features are beneficial.
- No feature's cost is too much.
- Features adhere to meta-level requirements.
- The data pipeline has appropriate privacy controls.
- New features can be added quickly.
- All input feature code is tested.

Model:

- Model specs are reviewed and submitted.
- Offline and online metrics correlate.

- All hyperparameters have been tuned.
- The impact of model staleness is known.
- A simple model is not better.
- Model quality is sufficient on important data slices.
- The model is tested for considerations of inclusion.

Infrastructure:

- Training is reproducible.
- Model specs are unit tested.
- The ML pipeline is integration tested.
- Model quality is validated before serving.
- The model is debug gable.
- Models are canaried before serving.
- Serving models can be rolled back.

Monitoring:

- Dependency changes result in notification.
- Data invariants hold for inputs.
- Training and serving are not skewed.
- Models are not too stale.
- Models are numerically stable.
- Computing performance has not regressed.
- Prediction quality has not regressed.

Model deployment

Be sure to have a versioning system in place for:

- Model parameters
- Model configuration
- Feature pipeline

- Training dataset
- Validation dataset

A common way to deploy a model is to package the system into a Docker container and expose a REST API for inference.

2.6 Specific Requirements

2.7 Functions

Functional requirements describe the system functionality, while the non-functional requirements describe system properties and constraints.

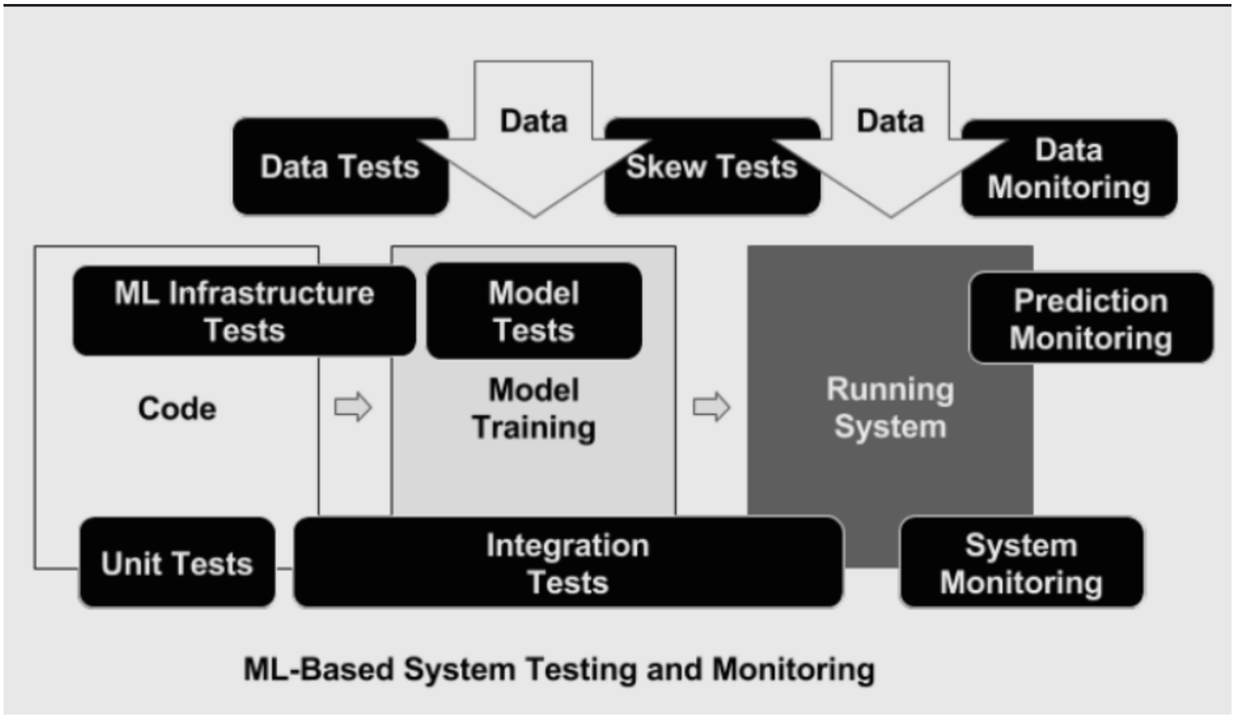
Functional requirements capture the intended behavior of the system. These behaviors may be expressed as services, tasks, or the functions the system is required to perform. This lays out important concepts and discusses capturing functional requirements in such a way they can drive architectural decisions and be used to validate the architecture. Features may be additional functionality, or differ from basic functionality along some quality attribute. In the proposed system, concert assesses the compliance of a workflow by analyzing the five established elements required to check for the rule adherence in workflows: activities, data, location, resources, and time limits.

Application 1.0 Potato leaf analysis

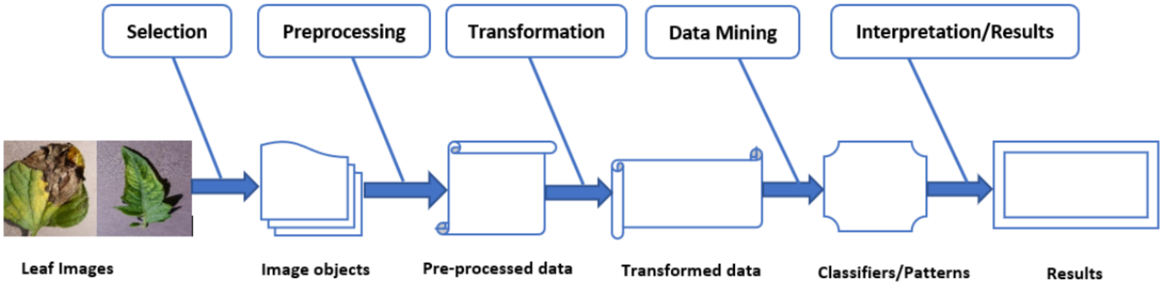
Input	Selection of the link to the potato leaf Module from the home main page and user will provide input image.
Action	All users are transferred to application Console Page.
Output	The output will provide the prediction of leaf image as per model predict the output.
Notes	Project information should be drawn from a potato leaf created database.

Priority High

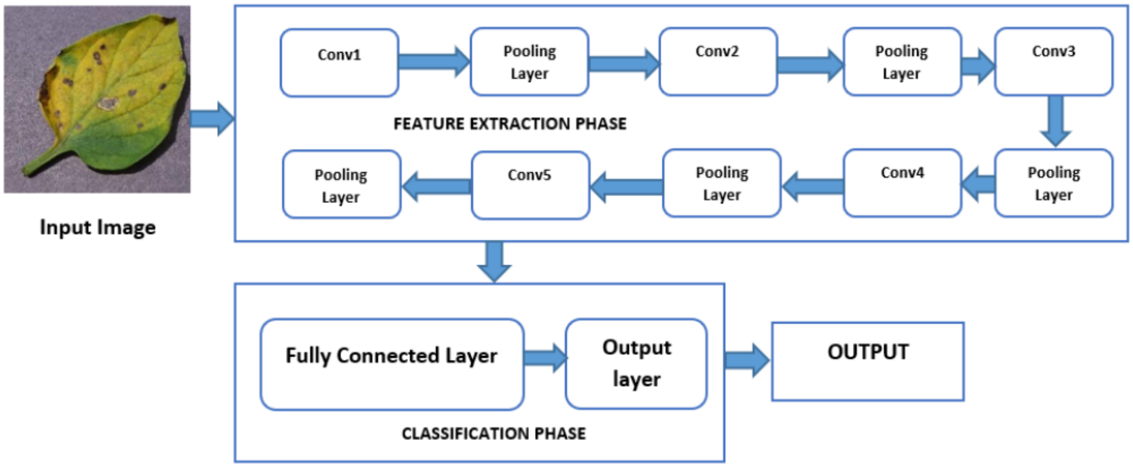
2.8 Standards Compliance(diagrams)



The way of ML model fit and train



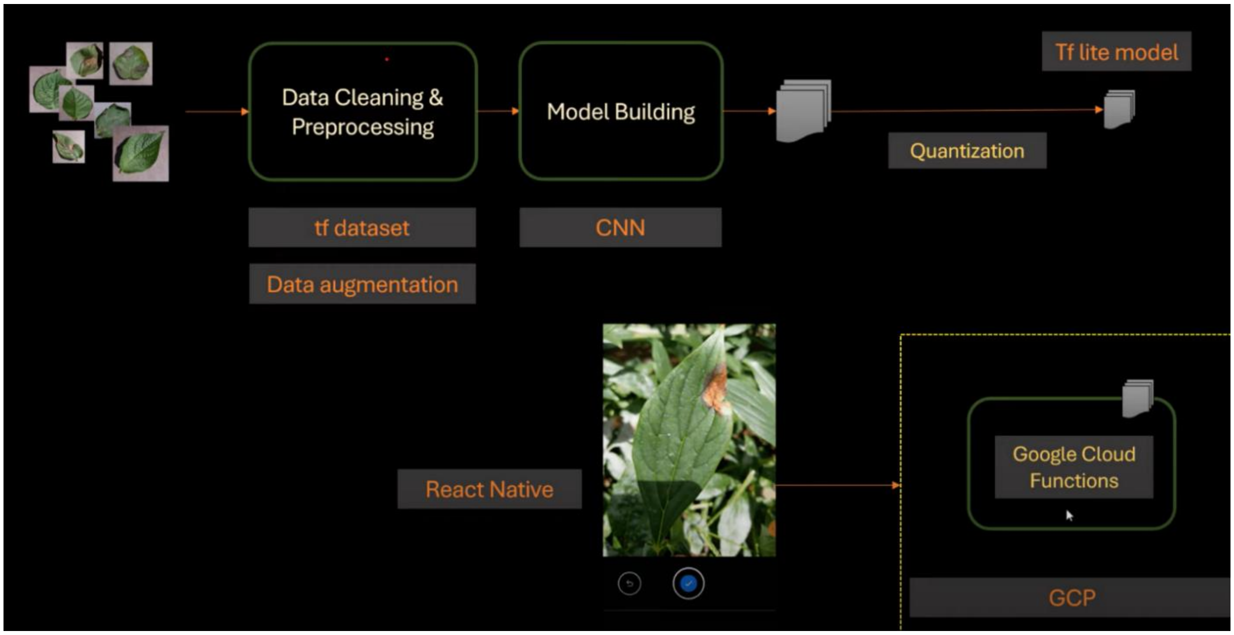
KDD methodology in reference with tomato leaf disease image classification



CNN architecture

Interpretation or evaluation of the results

Accuracy, precision, recall, and f1 score are the predominant metrics considered in this study to measure performance of the models. Tests are carried out on three CNN models with multiple layers and transfer learning approach along with three other machine learning algorithms SVM, Random Forest and XGBoost to evaluate the effectiveness of the models with the below-mentioned metrics.



2.9 Software and Hardware Requirements

Software:

a) Front-end

I. React Js
II. React Native
III. HTML
IV. CSS
V. Bootstrap

b) Back-end

I. Tf Serving
II. FastAPI
III. Google cloud platform
IV. TensorFlow

c) Model Building

I. Convolutional Neural Network
II. Data Augmentation
III. Tf dataset
IV. TensorFlow

Hardware:

- RAM: 8GB
- Hard Disk: SSD 556GB
- OS: Windows 11, Android device
- Camera