english

Visvesvaraya Technological University, Belagavi
Government Engineering College, Hassan 573 201

Project Report On

CURED TOBACCO LEAVES GRADING SYSTEM

4GH15CS026	Hemanth M R
4GH15CS070	Arpitha V N
4GH16CS407	Pallavi M S
4GH16CS409	Rani H S

Under the Guidance of

Mr. M T ThirtheGowda *B.E.,M.Tech.,MISTE.*
Assistant Professor
Dept. of Computer Science & Engineering
Government Engineering College, Hassan

Department of Computer Science & Engineering
Government Engineering College, Hassan
September 2018-19

Government Engineering College, Hassan-573 201
Visvesvaraya Technological University, Belagavi

Certificate

This is to certify that the project work entitled **"Cured Tobacco Leaves Grading System"** is a bonafide work carried out by **Hemanth M R(4GH15CS026),Arpitha V N(4GH15CS070),Pallavi M S(4GH16CS407),Rani H S (4GH16CS409)** in fulfillment of the award of the degree of Bachelor of Engineering in Computer Science Engineering of Visvesvaraya Technological University, Belagavi, during the year 2018-19. It is certified that all corrections / suggestions indicated during internal evaluation have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

Guide | *Head of the Department*

Mr.M T ThirtheGowda
B.E.,M.Tech.,MISTE.
Asst. Professor
Dept of CS & E, GEC, Hassan

Mr.Chethan K C *B.E.,M.Tech.,*
Professor and Head of the Department
Dept of CS & E, GEC, Hassan

Principal

Dr.K C Ravishankar *B.E.,M.Tech.,Ph.D.*
Principal
GEC, Hassan-573 201

Date :
Place : Hassan

Examiners : 1._____

2._____

Acknowledgement

At the outset I express my most sincere grateful thanks to my **Guide, Mr. M T ThirtheGowda, Assistant Professor, Department of CS & E**, for his continuous support and advice not only during the course of the project but also during the period of my stay in GECH.

I express my gratitude to **Mr. Ranganatha S**, Project Coordinator and Assistant Professor, Department of CS E, for his encouragement and support throughout the work.

I express my gratitude to **Mr. Chethan K C**, Professor and Head, Department of CS & E, for his encouragement and support throughout the work.

I wish to express thanks to our beloved **Principal**, **Dr. K C Ravishankar**, for encouragement throughout my studies.

Finally, I express my gratitude to all teaching and non-teaching staff of Department of CS & E, my fellow classmates and my parents for their timely support and suggestions.

Hemanth M R (4GH15CS026)

Arpitha V N (4GH15CS070)

Pallavi M S (4GH16CS407)

Rani H S (4GH16CS409)

Table of Contents

english

List of Figures

Abstract

Most of classification, quality evaluation or grading of the flue-cured tobacco leaves are manually operated, which relies on the judgmental experience of experts, and inevitably limited by personal, physical and environmental factors. The classification and the quality evaluation are therefore subjective and experientially based.A grading system based on image processing techniques was developed for automatically inspecting and grading flue-cured tobacco leaves.

Determination of the current tobacco grade classification performed by the grader with a variety of human frailties.Therefore it is necessary to develop classification automation tools. Classification is done on two major classes namely class-1, class-2 for obtaining global efficiency on the test set consisting about images of each cluster.The decision on grades was made based on nearest neighbour method. A comparative study is performed on the results from the proposed model with existing models, state of the art models on tobacco leaf classification.

Chapter 1

Introduction

Tobacco is one of the most successful commercial crops cultivated on this planet. China, India, Brazil and USA are the major producers of tobacco worldwide and these four nations alone contribute around 86 percent of the global production. India is the top two contributors to the global tobacco production and its estimated that around 750 M kgs of tobacco is being produced in the area around 0.45 M hectares.

Tobacco is cultivated in many parts of the world because of its high economic value.Quality inspection of tobacco leaves plays a crucial role in quality assurance of tobacco productions. After curing, the tobacco leaves are inspected and graded according to their color intensity, maturity, leaf structure, body, oil, length, appearance, waste and other characteristics.

At present, the grading process is performed manually throughout the world. The grading process is extremely labor-intensive and millions of man-days are required to grade each years crop. A high level of skill is required to the graders, but still many mistakes are made by them because the process is highly subjective.So graders are eager for equipments that can help them grade tobacco leaves. If we can use machine vision technology and design algorithms to grade tobacco leaves automatically, it will be very useful for improving the level and efficiency of tobacco grading, arbitrating the dispute of the quality of tobacco leaves between buyer and seller.

Because of the diversity and complexity of tobacco leaves, most of the classification and the quality evaluation of the flue-cured tobacco leaves are manually operated. It is a rigorous task. The tobacco leaves must be carefully classified by size, texture and colour, all aided by a well-seasoned experts feeling about the fine properties of the leaves. Errors often occur when the experts are tired, and the results of classification and quality evaluation relies on the judgmental experience of experts and many other factors, such as the emotion of experts, the human eyesight, the condition of illumination, etc. The grading process is extremely laborious, making the classification and the quality evaluation subjective and experientially based, while the efficiency and

the stability of error rate are not satisfying enough. New technology and equipments are needed to automate the quality inspection process of tobacco leaves.

1.1 What is Machine Learning ?

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. The aim of machine learning is to allow the computers learn automatically without human intervention and adjust actions accordingly.

Machine learning tasks are typically classified into several broad categories:

- **Supervised learning :** The computer is presented with example inputs and their desired outputs, given by a teacher, and the goal is to learn a general rule that maps inputs to outputs. As special cases, the input signal can be only partially available, or restricted to special feedback.
- **Semi-supervised learning :** The computer is given only an incomplete training signal: a training set with some (often many) of the target outputs missing.
- **Active learning :** The computer can only obtain training labels for a limited set of instances (based on a budget), and also has to optimize its choice of objects to acquire labels for. When used interactively, these can be presented to the user for labeling.
- **Unsupervised learning :** No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself
- **Reinforcement learning :** Data (in form of rewards and punishments) are given only as feedback to the programs actions in a dynamic environment, such as driving a vehicle or playing a game against an opponent.

1.1.1 What Is Deep Learning ?

Learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. In an image recognition application, the raw input may be a matrix of pixels; the first representational layer may abstract the pixels and encode edges; the second layer may compose and encode arrangements of edges; the third layer may encode a nose and eyes; and the fourth layer may recognize that the image contains a face. Importantly, a deep learning process can learn which features to optimally place in which level on its own.

Deep learning architectures are often constructed with a greedy layer-by-layer method. Deep learning helps to disentangle these abstractions and pick out which features improve performance.

For supervised learning tasks, deep learning methods obviate feature engineering, by translating the data into compact intermediate representations akin to principal components, and derive layered structures that remove redundancy in representation.

Deep learning algorithms can be applied to unsupervised learning tasks. This is an important benefit because unlabeled data are more abundant than labeled data. Examples of deep structures that can be trained in an unsupervised manner are neural history compressors and deep belief networks

1.2 Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. In July 2018, Van Rossum stepped down as the leader in the language community after 30 years.

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

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of Pythons other implementations. Python and CPython are managed by the non-profit Python Software Foundation.

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.

1.3 KNN

K Nearest Neighbor (KNN from now on) is one of those algorithms that are very simple to understand but works incredibly well in practice.This is why it is called the K Nearest Neighbours algorithm.

Most people learn the algorithm and do not use it much which is a pity as a clever use of KNN can make things very simple. It also might surprise many to know that KNN is one of the top 10 data mining algorithms.

The purpose of the k Nearest Neighbours (KNN) algorithm is to use a database in which the data points are separated into several separate classes to predict the classification of a new sample point.

Also it is surprisingly versatile and its applications range from vision to proteins to computational geometry to graphs and so on .

We consider each of the characteristics in our training set as a different dimension in some space, and take the value an observation has for this characteristic to be its coordinate in that dimension, so getting a set of points in space.

Can then consider the similarity of two points to be the distance between them in this space under some appropriate metric.

Way in which the algorithm decides which of the points from the training set are similar enough to be considered when choosing the class to predict for a new observation is to pick the k closest data points to the new observation, and to take the most common class among these.

1.4 Introduction to Image Processing

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image.Image processing is one of the rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps:

- Importing the image via image acquisition tools.
- Analysing and manipulating the image.

- Output in which result can be altered image or report that is based on image analysis.

There are two types of methods used for image processing namely,

Analogue image processing: Analog or visual techniques of image processing can be used for the hard copies like printouts and photographs.

- Image analysts use various fundamentals of interpretation while using these visual techniques. The image processing is not just confined to area that has to be studied but on knowledge of analyst.
- It is another important tool in image processing through visual techniques.
- Analysts use various fundamentals of interpretation while using these visual techniques.

Digital image processing: Digital Processing techniques help in manipulation of the digital images by using computers.

- Raw data from imaging sensors from satellite platform contains deficiencies.
- Get over such flaws and to get originality of information, it has to undergo various phases of processing.
- The three general phases that all types of data have to undergo while using digital technique are Pre- processing, enhancement and display, information extraction.

1.5 Feature Extraction

Features are the information that are extracted from an image. These are real-valued numbers (integers, float or binary). There are a wider range of feature extraction algorithms in Computer Vision. When deciding about the features that could quantify leaves, we could possibly think of Color, Texture and Shape as the primary ones. This is an obvious choice to globally quantify and represent the leaves.

1.5.1 Global Feature Descriptors

These are the feature descriptors that quantifies an image globally. These dont have the concept of interest points and thus, takes in the entire image for processing. Some of the commonly used global feature descriptors are

- Color - Color Channel Statistics (Mean, Standard Deviation) and Color Histogram.

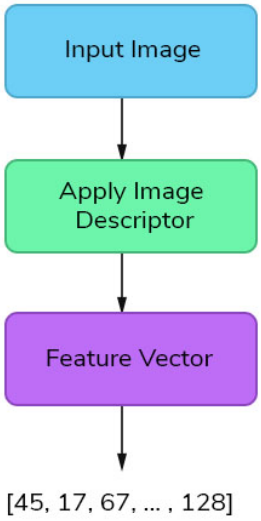

Figure 1.1: feature extraction

- Shape - Hu Moments, Zernike Moments.
- Texture - Haralick Texture, Local Binary Patterns (LBP)

1.5.2 Local Feature Descriptors

These are the feature descriptors that quantifies local regions of an image. Interest points are determined in the entire image and image patches/regions surrounding those interest points are considered for analysis. Some of the commonly used local feature descriptors are

- SIFT (Scale Invariant Feature Transform)
- SURF (Speeded Up Robust Features)
- ORB (Oriented Fast and Rotated BRIEF)

1.5.3 Combining Global Features

There are two popular ways to combine these feature vectors.

- For global feature vectors, we just concatenate each feature vector to form a single global feature vector. This is the approach we will be using in this tutorial.

- For local feature vectors as well as combination of global and local feature vectors, we need something called as Bag of Visual Words (BOVW). This approach is not discussed in this tutorial, but there are lots of resources to learn this technique. Normally, it uses Vocabulory builder, K-Means clustering, Linear SVM, and Td-Idf vectorization.

1.6 Functions for global feature descriptors

1.6.1 Hu Moments

To extract Hu Moments features from the image, we use cv2.HuMoments() function provided by OpenCV. The argument to this function is the moments of the image cv2.moments() flatenned. It means we compute the moments of the image and convert it to a vector using flatten(). Before doing that, we convert our color image into a grayscale image as moments expect images to be grayscale.

1.6.2 Haralick Textures

To extract Haralick Texture features from the image, we make use of mahotas library. The function we will be using is mahotas.features.haralick(). Before doing that, we convert our color image into a grayscale image as haralick feature descriptor expect images to be grayscale.

1.6.3 Color Histogram

To extract Color Histogram features from the image, we use cv2.calcHist() function provided by OpenCV. The arguments it expects are the image, channels, mask, histSize (bins) and ranges for each channel [typically 0-256). We then normalize the histogram using normalize() function of OpenCV and return a flattened version of this normalized matrix using flatten().

1.7 Training classifiers

After extracting, concatenating and saving global features and labels from our training dataset, its time to train our system. To do that, we need to create our Machine Learning models. For creating our machine learning models, we take the help of scikit-learn.

We will choose Logistic Regression, Linear Discriminant Analysis, K-Nearest Neighbors, Decision Trees, Random Forests, Gaussian Naive Bayes and Support Vector Machine as our machine learning models. To understand these algorithms, please go through Professor Andrew NGs amazing Machine Learning course at Coursera or you could look into this awesome playlist of Dr.Noureddin Sadawi.

Furthermore, we will use train test split function provided by scikit learn to split our training dataset into train data and test data. By this way, we train the models with the train data and test the trained model with the unseen test data. The split size is decided by the test size parameter.

We import all the necessary libraries to work with and create a models list. This list will have all our machine learning models that will get trained with our locally stored features. During import of our features from the locally saved .h5 file format, it is always a good practice to check its shape. To do that, we make use of np.array() function to convert the .h5 data into a numpy array and then print its shape.

Finally, we train each of our machine learning model and check the cross-validation results.

Chapter 2

Libraries and Tools

2.1 Spyder:

Spyder, the Scientific Python Development Environment, is a free integrated development environment (IDE) that is included with Anaconda. It includes editing, interactive testing, debugging and introspection features.

Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts.

It features a unique combination of the advanced editing, analysis, debugging and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection and beautiful visualization capabilities of a scientific package.

2.2 OpenCV:

OpenCV supports a wide variety of programming languages such as C++, Python, Java, etc., and is available on different platforms including Windows, Linux, OS X, Android, and iOS. Interfaces for high-speed GPU operations based on CUDA and OpenCL are also under active development.

OpenCV-Python is the Python API for OpenCV, combining the best qualities of the OpenCV C++ API and the Python language.

OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.

2.3 Scikit-learn:

- Simple and efficient tools for data mining and data analysis.
- Accessible to everybody, and reusable in various contexts
- Simple and efficient tools for data mining and data analysis.
- Built on NumPy, SciPy, and matplotlib.
- source, commercially usable - BSD license.

2.4 Mahotas:

Mahotas is a computer vision and image processing library for Python.

It includes many algorithms implemented in C++ for speed while operating in numpy arrays and with a very clean Python interface.

Mahotas currently has over 100 functions for image processing and computer vision and it keeps growing. Some examples of mahotas functionality:

- convex points calculations.
- convolution.
- Sobel edge detection.
- morphological processing

2.5 NumPy:

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object.
- sophisticated (broadcasting) functions.
- tools for integrating C/C++ and Fortran code.
- useful linear algebra, Fourier transform, and random number capabilities.

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

2.6 SciPy:

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

Matplotlib tries to make easy things easy and hard things possible.

You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

2.7 h5Py:

The h5py package is a Pythonic interface to the HDF5 binary data format.

It lets you store huge amounts of numerical data, and easily manipulate that data from NumPy.

example, you can slice into multi-terabyte datasets stored on disk, as if they were real NumPy arrays. Thousands of datasets can be stored in a single file, categorized and tagged however you want.

H5py uses straightforward NumPy and Python metaphors, like dictionary and NumPy array syntax.

For example, you can iterate over datasets in a file, or check out the .shape or .dtype attributes of datasets. You don't need to know anything special about HDF5 to get started.

In addition to the easy-to-use high level interface, h5py rests on a object-oriented Cython wrapping of the HDF5 C API. Almost anything you can do from C in HDF5, you can do from h5py.

2.8 Keras

Keras is an open source neural network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, or Theano. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System),and its primary author and maintainer is Francois Chollet, a Google engineer.

They contains numerous implementations of commonly used neural network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier. The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel.

Keras allows users to productize deep models on smartphones (iOS and Android), on the web, or on the Java Virtual Machine. It also allows use of distributed training of deep learning models on clusters of Graphics Processing Units (GPU) and Tensor processing units (TPU).

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.

2.9 TensorFlow

TensorFlow is an open source software library released in 2015 by Google to make it easier for developers to design, build, and train deep learning models. TensorFlow originated as an internal library that Google developers used to build models in-house, and we expect additional functionality to be added to the open source version as they are tested and vetted in the internal flavor.

At a high level, TensorFlow is a Python library that allows users to express arbitrary computation as a graph of data flows. Nodes in this graph represent mathematical operations, whereas edges represent data that is communicated from one node to another. Data in TensorFlow are represented as tensors, which are multidimensional arrays.

Chapter 3

Requirement Specification

A System Requirements Specification (SRS) (also known as a Software Requirements Specification) is a document or set of documentation that describes the features and behaviour of a system or software application. It includes a variety of elements (see below) that attempts to define the intended functionality required by the customer to satisfy their different users.

Depending on the methodology employed the level of formality and detail in the SRS will vary, but in general a SRS should include a description of the SRS will vary, but in general a SRS should include a description of the functional requirements, system requirements, constraints, assumptions and acceptance criteria.

3.1 Functional Requirements

Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. Functional requirements are supported by non-functional requirements (also known as "quality requirements"), which impose constraints on the design or implementation (such as performance requirements, security, or reliability).

Generally, functional requirements are expressed in the form "system must do requirement". Functional requirements specify particular results of a system. This should be contrasted with non-functional requirements, which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system.

3.2 Non-functional Requirements

A non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions.The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually architecturally significant requirements.

Non-functional requirements are in the form of "system shall be ¡requirement¿", an overall property of the system as a whole or of a particular aspect and not a specific function. The system's overall properties commonly mark the difference between whether the development project has succeeded or failed.Non-functional requirements are often called "quality attributes" of a system. Other terms for non-functional requirements are "qualities", "quality goals", "quality of service requirements", "constraints", "non-behavioral requirements", or "technical requirements".

3.3 Hardware Requirements

Processor : Intel CORE i5 or higher
RAM : 4GB RAM
Monitor : EGVGA Compatible
Keyboard : Normal keyboard(QWERTY)
Hard Disk : 500GB or higher recommended

3.4 Software Requirements

Operating system : GNU/Linux.
Language Tool : Python.
Libraries : Python Libraries.
IDE : Anaconda.

Chapter 4

Literature Survey

Authors in [1] have proposed system which finds size of different fruits and accordingly different fruits can be sorted using fuzzy logic, here author proposed matlab for the features extraction and for making GUI.

Authors in [2] have developed an automated grading system using image processing.

Authors in [3] have presented a study on colour image processing based intelligent fruit sorting system. In this he used Fruit sorting by classic Bayes classifier, whose parameters were obtained by a study module.

Authors in [4] define the process of colour classification, it involves extraction of useful information concerning the spectral properties of object surfaces and discovering the best match from a set of known descriptions or class models to implement the recognition task.

The Nandi.C.S [5] paper presents a computer vision based system for automatic grading and sorting of agricultural products like Mango based on maturity level. The application of machine vision based system, aimed to replace manual based technique for grading and sorting of fruit.Some work in this direction has already been undertaken.

For example, Tucker and Chakrabarty [6] have recently produced software that uses image segmentation and classification techniques to identify blight and rust lesions on oats and sunflower leaves for the purpose of rapid disease assessment in the field. For tobacco, there has been some research into the estimation of leaf area in Japanese burley and also into a Cuban dark tobacco variety called corojo [7]. Both of these projects conclude that leaf area can be estimated with only small errors, in the case of the cultivar they were using, from simply-extracted features such as leaf length and width. Such feature extraction is what this study seeks to automate by applying

machine vision techniques to the classification of leaf shape, in order to estimate the plant position from which the leaf was reaped.

Authors in paper [8] proposed that Image preprocessing include the detection and restoration of bad lines, geometric rectification or image registration, radiometric calibration and atmospheric correction, and topographic correction. If different ancillary data are used, data conversion among different sources or formats and quality evaluation of these data are also necessary before they can be incorporated into a classification procedure. Accurate geometric rectification or image registration of remotely sensed data is a prerequisite for a combination of different source data in a classification process.

Authors in paper [9] proposed that Selecting suitable variables is a critical step for successfully implementing an image classification. Many potential variables be used in image classification, including spectral signatures, vegetation indices, transformed images, textural or contextual information, multitemporal images, multisensor images, and ancillary data. Due to different capabilities in landcover separability, the use of too many variables in a classification procedure may decrease classification accuracy .

Authors in paper [10] proposed that Image segmentation performance of tobacco leaves in natural environment, an automatic segmentation model of leaf with active gradient and local information is proposed. Firstly, a segmented monotone decreasing edge composite function is proposed to accelerate the evolution of the level set curve in the gradient smooth region. Secondly, Canny edge detection operator gradient is introduced into the model as the global information. In the process of the evolution of the level set function, the guidance information of the energy function is used to guide the curve evolution according to the local information of the image, and the smooth contour curve is obtained. And the main direction of the evolution of the level set curve is controlled according to the global gradient information, which effectively overcomes the local minima in the process of the evolution of the level set function. Finally, the Heaviside function is introduced into the energy function to smooth the contours of the motion and to increase the penalty function to calibrate the deviation of the level set function so that the level set is smooth and closed. The results showed that the model of tobacco leaf edge profile curve could be obtained in the model of tobacco leaf covered by bare soil. In the complex background, the model can segment the leaves of the tobacco with uneven illumination, shadow and weed background, and it is better to realize the ideal extraction of the edge of the blade.

Authors in paper [11] proposed that A grading system based on image processing techniques is developed for automatic inspection and grading of tobacco leaves. The system used machine vision in extraction and analysis of color, size, shape, surface texture and vein. A two-dimensional feature space is proposed to express feature

distribution of tobacco leaves. The space is found to be well confined in an elliptic region. A database is constructed to record the feature distribution of standard contrast tobacco leaves prepared by experts through visual evaluation. The decision on grades is made based on the so-called nearest-neighbor method for which the overall difference among features between the measured tobacco leaves and the standard contrast samples were used as a target parameter for judgment. This system can be easily trained by users with the knowledge of the feature distribution information of different tobacco leaves.

Authors in paper [12] proposed that Fuzzy approach to classify the colour images based on their content, to pose a query in terms of natural language and fuse the queries based on neural networks for fast and efficient retrieval. Number of experiments is conducted for classification and retrieval of images on sets of images and promising results were obtained. The results were analysed and compared with other similar image retrieval system.

Authors in paper [13] proposed that Capturing of a leave on the tree.Converted to gray scale image and then scaled the size to minimize the training time of neural network. After that, noise is added in order to create efficient and different input data for the neural network. Thus, the network has input layer with neurons. After several experiments, it has been found the optimum range for the number of neurons for the hidden layers. Number of output neurons were used in order to recognize number of leaves types.

Authors in paper [14] proposed that An image processing system of tobacco leaves grading is Actually, the lighted cabinet is the same one as Zhang has used , but all other equipments, such as computer and camera,has been updated. The image processing system is consisted of a color camera.

All the authors mentioned above have done grading of different fruits and leaves. In the similar way we are grading the flue-cured tobacco leaves based on the digital image processing and the fuzzy sets theory in this project.

Chapter 5

Objectives

Human eyes may fail to recognize the minute differences in colour, shape or texture of the leaf. But the system is developed to recognize quality of leaves more accurately. The separation of cured Tobacco leaves manually consumes more time and also it requires human resource. If it is done by the system, then the accuracy will be improved and without any manual intervention the quality of the leaves may be detected. The lot may contain the things other than leaves. This can be detected accurately in less amount of time and can be removed for further analysis it is possible to cure the leaves in the required conditions.

The system classifies the leaves as Quality 1, Quality 2 based on shape, colour and texture. As the quality of the leaf will be known prior, one can have idea about further processes and curing leaves. As the leaves are classified by the system, Time taken by the system will be very less. System may minimize the errors which could be common in manual method of classification. This system reduces human efforts and manual intervention. Based on this result, one can decide how much temperature is given to a particular leaf.

Chapter 6

System Design

6.1 Existing System

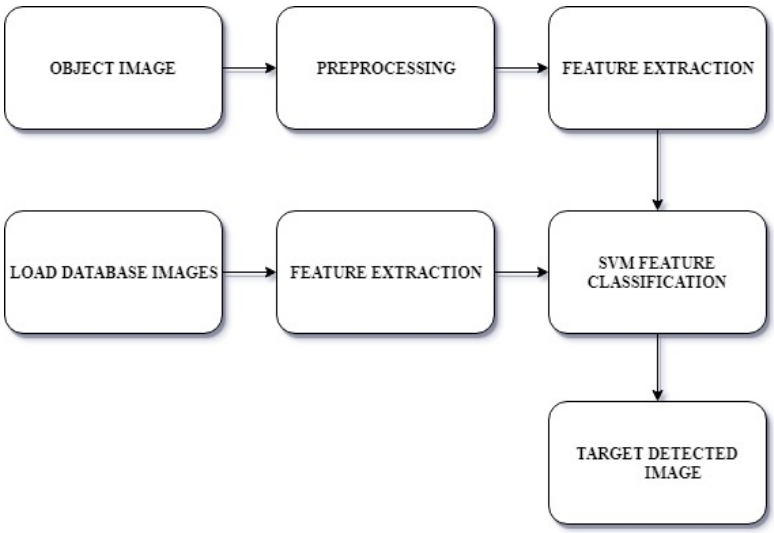

Figure 6.1: existing System

In this System using support vector machine method for clssification and classification made based on color. The main disadvantage of the SVM algorithm is that it has several key parameters that need to be set correctly to achieve the best classification results for any given problem. Parameters that may result in an excellent classification accuracy for problem A, may result in a poor classification accuracy for problem B. The user may, therefore, have to experiment with a number of different parameter settings in order to achieve a satisfactory result. The main parameters that the user should experiment with are the SVM kernel type (which can be set by the setKernelType(UINT kernelType) method), the SVM type (which can be set by the

setSVMType(UINT svmType) method), and the kernel-specific parameters (such as gamma, degree, nu, etc.).

Choosing a good kernel function is not easy. Long training time on large data sets Difficult to understand and interpret the final model, variable weights and individual impact Since the final model is not so easy to see, we can not do small calibrations to the model hence its tough to incorporate our business logic

6.2 Proposed design

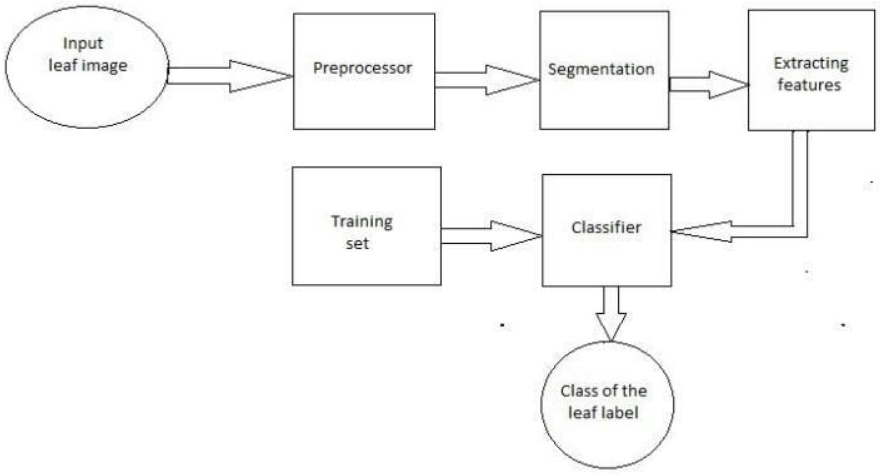

Figure 6.2: Proposed design

The method of procedure in this project is summarised in Figure 6.2. Given the statement of the problem and definition of the aim of the project, as stated in introduction,In this System using k nearest neighbour method for clssification and classification made based on color, texture and shape.

Each leaf was then prepared and photographed under suitable and stringently consistent conditions of lighting and scale. Later, each photographic image was digitised for use in the computer classifier algorithms. To ensure the highest standards of consistency, images were pre-processed prior to the isolation (segmentation) of the main objects in each image.

Once each object had been unambiguously recognised, features could be extracted from it. Such features may be as simple to extract- as, say, the length or pixel area of the segmented object, or they may be. The results of quite extensive mathematical derivation. Whichever the case, the process of choosing appropriate features for a

given classification purposes is notoriously heuristic and often adhoc and this dissertation will cover in detail the reasons behind the feature choices that were made in this project.

Arising from these results, it has been possible to give some objective criteria for the grading of tobacco leaves by colour,shape and texture and to draw certain conclusions regarding the feasibility of the machine vision grading of flue-cured tobacco leaves

Below is the list of few of the reasons to choose K-NN machine learning algorithm:

1. K-NN is pretty intuitive and simple: K-NN algorithm is very simple to understand and equally easy to implement. To classify the new data point K-NN algorithm reads through whole dataset to find out K nearest neighbors.

2.K-NN has no assumptions: K-NN is a non-parametric algorithm which means there are assumptions to be met to implement K-NN. Parametric models like linear regression has lots of assumptions to be met by data before it can be implemented which is not the case with K-NN.

3.It constantly evolves: Given its an instance-based learning; k-NN is a memory-based approach. The classifier immediately adapts as we collect new training data. It allows the algorithm to respond quickly to changes in the input during real-time use.

Chapter 7

Snapshots

This chapter consists the results drawn after successful execution of our project.

7.1 Data Augmentation

- The input image which can be used to generate many images.

Figure 7.1: Input Image

- With only a few operations, a single image can be augmented to produce large numbers of new images.

Figure 7.2: Augmented Image by rotate90

Figure 7.3: Augmented Image by rotate270

Figure 7.4: Augmented Image by flip left right

Figure 7.5: Augmented Image by flip top bottom

7.2 Image Resizing

- Change the basewidth to any other number if we need a different width for images.

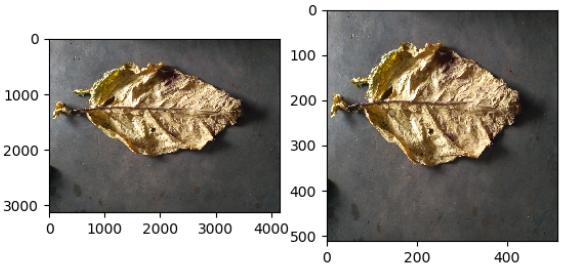

Figure 7.6: Resizing the Image

7.3 Image Denoising

- Remove the noice from color images.

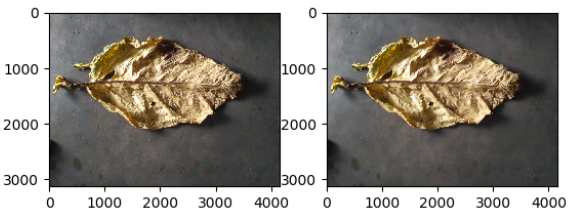

Figure 7.7: Denoising the image

7.4 Training classifiers

- For each of the training label name, we iterate through the corresponding folder to get all the images inside it.

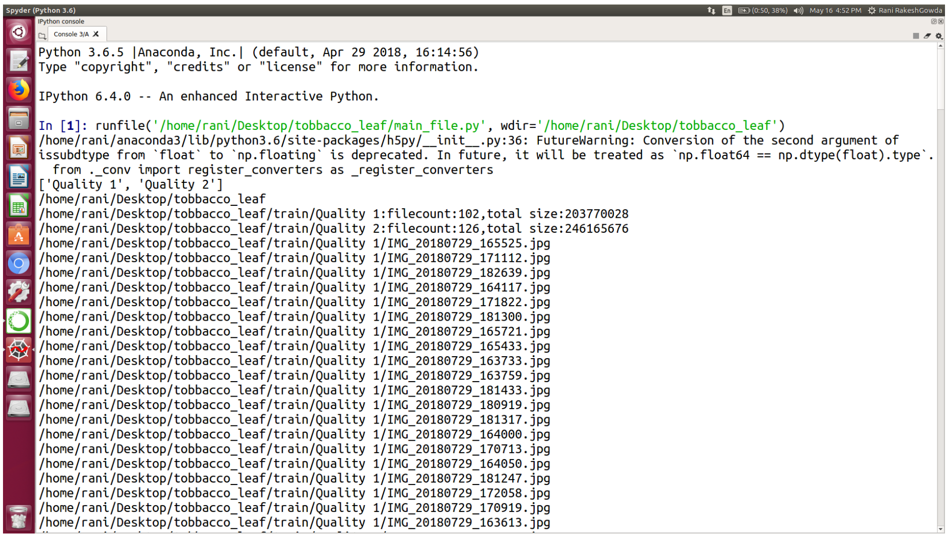

Figure 7.8: Training the data in each folder

7.4.1 Training results

- After extracting, concatenating and saving global features and labels from our training dataset, its time to train our system.

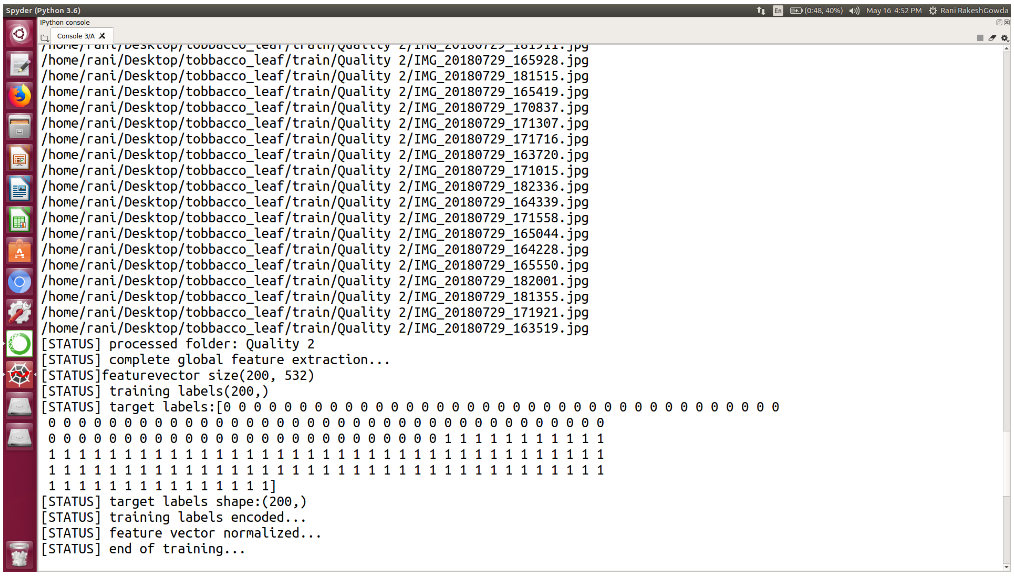

Figure 7.9: Training results

7.4.2 Machine learning Algorithm Comparison

- Will be splitting our training dataset into train data as well as test data.

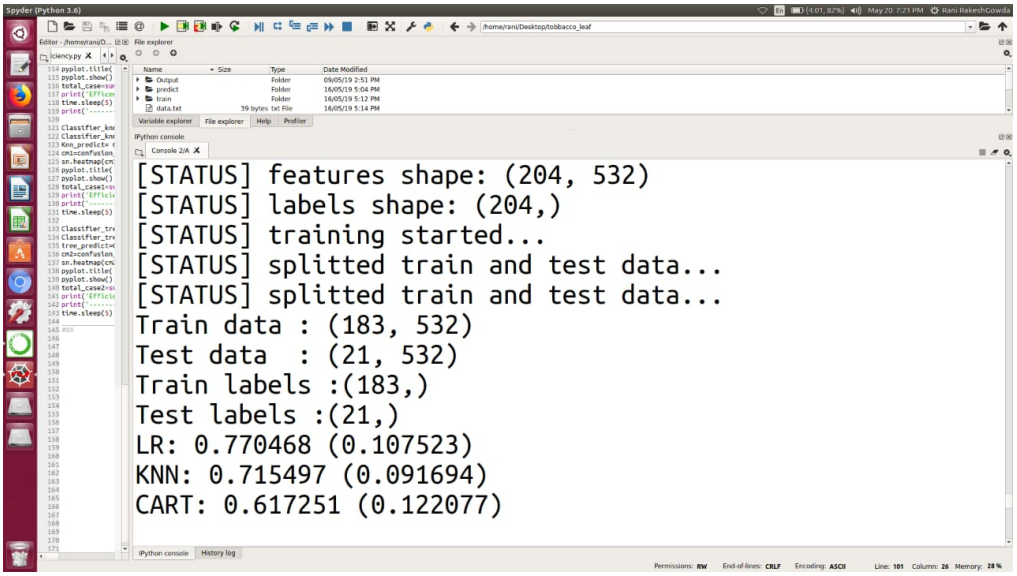

Figure 7.10: . Comparison values of different machine learning classifiers

7.4.3 Comparison chart

- We train each of our machine learning model and check the cross-validation results.

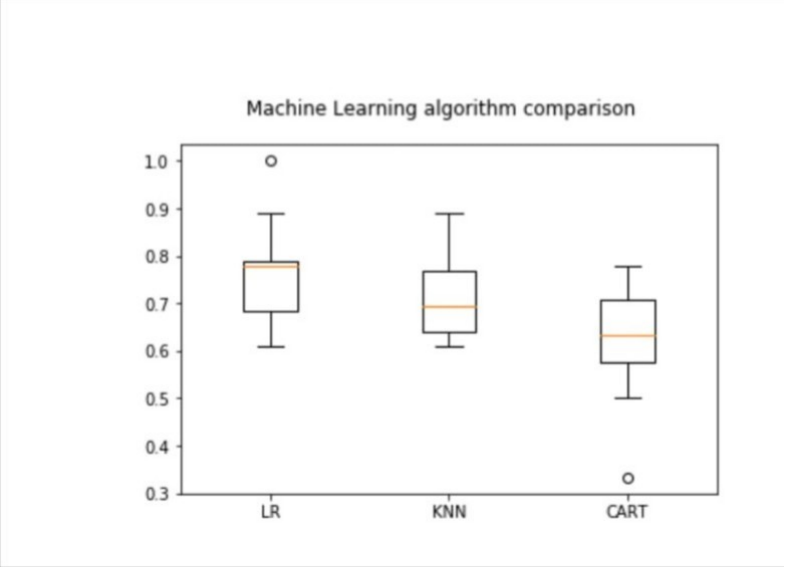

Figure 7.11: . Comparison chart of different machine learning classifiers used (Y-axis: Accuracy)

7.4.4 Confusion Matrix and Accuracy for LR

- This is mainly due to the number of images we use per class.We need large amounts of data to get better accuracy.

Figure 7.12: Conclusion Matrix and Accuracy

7.4.5 Confusion Matrix and Accuracy for KNN

- This is mainly due to the number of images we use per class.We need large amounts of data to get better accuracy.

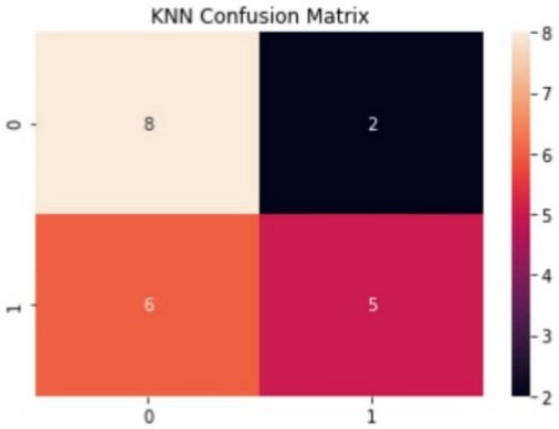

Figure 7.13: Conclusion Matrix and Accuracy

7.4.6 Confusion Matrix and Accuracy for CART

- This is mainly due to the number of images we use per class.We need large amounts of data to get better accuracy.

Figure 7.14: Conclusion Matrix and Accuracy

7.4.7 Final result

- we train each of our machine learning model and check the cross-validation results.

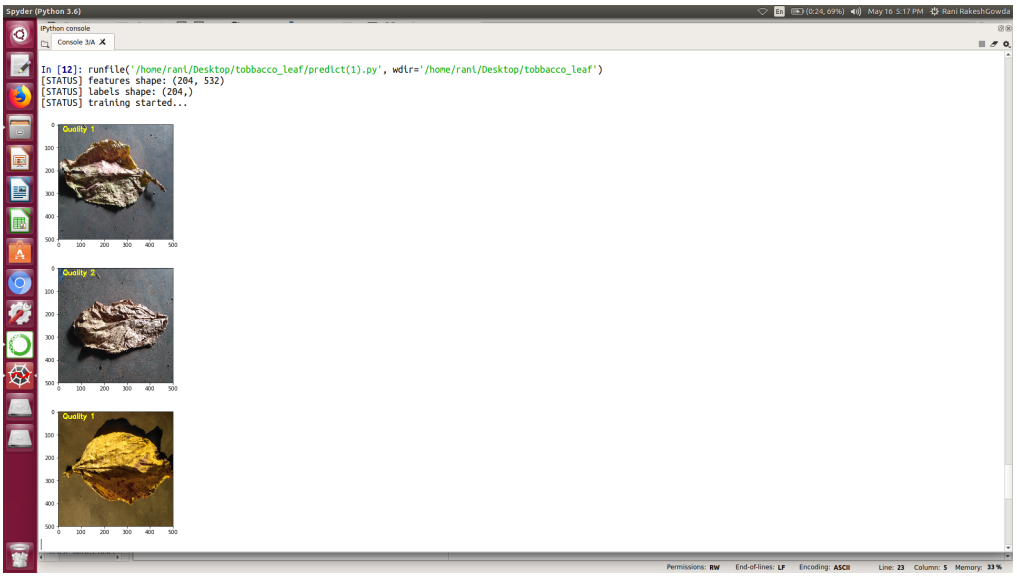

Figure 7.15: Resulting input images

Chapter 8

Conclusion

Proposed Tobacco Leaves Classification System performance even on limited training samples (image data set) compared to existing models. The proposed model achieved a global correctness of 71.5 perent. On application, the state of art image classification technique of different Algorithm for the model.This can be applied on samples in which its hard to describe the features.

This project proposes an automatic grading method of tobacco leaves based on the digital image processing and KNN method.The project is to find a feasible method to realize the automatic grading of tobacco leaves. We made a attempt to explore the applicability of the concepts of symbolic data to grade cured tobacco leaves. The newly proposed model has an ability to capture the variations of the features in training samples of cured tobacco leaves. In the proposed method we represent each cured tobacco leaf sample by three features color, texture and shape. The symbolic representation reduces the time taken to grade a given test sample of a cured tobacco leaf, as there is only one representative vector instead of n (training samples) number of representative vectors in the knowledge base. In order to investigate the effectiveness and robustness of the proposed method, we have conducted extensive experiments on our own dataset.

References

[1] Zhang, J.; Sokhansanj, S.; Wu, S.; Fang, R.; Yang, W.; Winter, P. A trainable grading system for tobacco leaves. Comput. Electron. Agric.1997, 16, 231244.

[2] Zhang, J.; Sokhansanj, S.; Wu, S.; Fang, R.; Yang, W.; Winter, P. A transformation technique from RGB signals to the Munsell system for color analysis of tobacco leaves. Comput. Electron. Agric. 1998, 19, 155166.

[3] Garcia, M.; Barreiro, P.; Ruiz, A.M.; Alonso, R.; Judez, L. Development of a virtual expert for color classification of tobacco leaves. Proc. Sens. Decis. Support Syst. Agric. Food Ind. Environ. 1998, 1, 105117.

[4] Zhang, F.; Fang, R.; Cai, J. Study of getting tobacco leaf weight based on neural network. Trans.Chin. Soc. Agric. Mach. 2000, 31, 6164.

[5] MacCormac, J. On-line image processing for tobacco grading in Zimbabwe. In Proceedings ofIEEE International Symposium on Industrial Electronics, Budapest, Hungary, 13 June 1993;pp. 327331.

[6] Zhang, F.; Fang, R.; Cai, J. Image retrieveal of standary tobacco leaf database. Trans. Chin. Soc.Agric. Mach. 2001, 32, 6667.

[7] Zhang, H.; Han, L.; Wang, Z. A fuzzy classification system and its application. Int. Conf. Mach.Learn. Cybern. 2003, 4, 25822586.

[8] Ma, W.; He, L.; Xu, S.; Chen, J.; Wu, Z. Image segmentation based on transmission characteristics of flue-cured tobacco leaves. Trans. Chin. Soc. Agricult. Eng. 2006, 22, 1341.

[9] CTRI-Rajahmundry. Tobacco in Indian Economy. http://www.ctri.org.in/fortobacco Economy.php 2015.

[10] LeCun, Yann, et al. 1998. Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE 86 (11): 22782324.

[11] Guru, D.S., et al. 2011. Min-max Representation of Features for Grading Cured Tobacco Leaves. Statistics and Applications 9 (12): 1529.

[12] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems, 10971105.

[13] Prasad, V., T.S. Rao, and M. Babu. 2016. Thyroid Disease Diagnosis Via Hybrid Architecture Composing Rough Data Sets Theory and Machine Learning Algorithms. Soft Computing 20 (3): 11791189.

[14] Lawrence, Steve, et al. 1997. Face Recognition: A Convolutional Neural-Network Approach. Neural Networks, IEEE Transactions on 8 (1): 98113.