

MAKERERE

P.O. Box 7062 Kampala Uganda
E-mail: info@cis.mak.ac.ug
URL: <http://www.cis.ac.ug>



UNIVERSITY

Tel: +256 - 41 - 540628 / 534560 / 1-97
URL: +256 - 41 - 540620

COLLEGE OF COMPUTING & INFORMATION SCIENCES

Computer Vision based automated whitefly count in cassava

by

CSC 11-41

**Department of Computer Science
School of Computing & Informatics Technology**

*A Project Report Submitted to the
School of Computing and Informatics Technology for the Study Leading to
a Project Report in Partial Fulfilment of the requirements for the
Award of the Degree of Bachelor of Science in
Computer Science of Makerere University*

Supervisor:

Dr. Ernest Mwebaze

Department of Computer Science
School of Computing & Informatics Technology
emwebaze@cis.mak.ac.ug Tel: +256414540628 Fax: +256414540620

February, 2018

Declaration

We Group CSC16-41 do hereby declare that this Project Report is original and has not been published and/or submitted for any other degree award to any other University before.

#	Names	Registration Number	Signature
1	Spice Diana	16/U/2345/PS	
2	Ibrahim Abiriga	16/T/8702/EVE	
3	Moses Golola	16/U/9867/PS	
4	Kim Kardashian	16/U/324	

Date:

Approval

This Project Report has been submitted for examination with the approval of the following supervisor.

Signed:

Date:

Dr. Robert Mugabe
BSc. CS, MSc. CS,
School of Computing & IT
College of Computing & IT
Makerere University
rmugabe@cis.mak.ac.ug

Dedication

We dedicate this report to the Almighty God without whom we can do nothing. We further dedicate it to our parents and guardians for their unceasing and selfless support throughout our stay in this university.

Acknowledgement

We are deeply indebted to our project supervisor Dr. Robert Mugabe whose unlimited steadfast support and inspirations have made this project a great success. In a very special way, we thank him for every support he has rendered unto us to see that we succeed in this challenging study.

Special thanks go to our friends and families who have contained the hectic moments and stress we have been through during the course of the research project.

We thank the school for giving us the grand opportunity to work as a team which has indeed promoted our team work spirit and communication skills. We also thank the individual group members for the good team spirit and solidarity.

Contents

Declaration	i
Approval	ii
Dedication	iii
Acknowledgement	iv
1 Introduction	1
1.1 Background	1
1.1.1 Disease detection in Cassava	1
1.2 Problem Statement	2
1.3 Main Objective	2
1.3.1 Specific Objectives	2
1.4 Scope of the study	3
1.5 Significance of the study	3
2 Literature Review	4
2.1 Introduction	4
2.2 Computer Vision	4
2.2.1 Fundamentals of Computer Vision	5
2.3 Object Detection	5
2.4 Review of Object Detection Techniques	6
2.4.1 Haartraining	6
2.4.2 Knowledge and Motion Based Methods	7
2.4.3 Background Subtraction Method	7
2.4.4 Template matching approach	8

Chapter 1

Introduction

1.1 Background

Cassava Mosaic Disease is majorly spread by the whiteflies which move from leaf to leaf in different cassava plantations. The whiteflies, comprising only the family Aleyrodidae, are small hemipterans and more than 1550 species have been described. Whiteflies typically feed on the underside of plant leaves. The ability of the whitefly to carry and spread disease is the widest impact they have had on global food production.

In the tropics and subtropics, whiteflies (Hemiptera: Aleyrodidae) have become one of the most serious crop protection problems. Economic losses are estimated in the hundreds of millions of dollars. While several species of whitefly cause crop losses through direct feeding, a species complex, or group of whiteflies in the genus *Bemisia* are important in the transmission of plant diseases. *Bemisia tabaci* and *B. argentifolii*, transmit African cassava mosaic.

1.1.1 Disease detection in Cassava

However, there is continuous need for timely and accurate information pertaining to their spread from region to region for proper management of the CMD incidence and severity. Information about the infestation rates should be got. This information would be used in monitoring and forecasting the spread of CMD over time and planning appropriate interventions to avert crises.

Conversely, such information is difficult to obtain at present, due to challenges such as the unavailability of suitable technical staff with the expertise to detect the white flies, the time and cost incurred by transport to rural

regions of the country, unavailability of salaries for the field staff, and impassable roads during rainy seasons in some regions of the country, and the time taken to coordinate paper reports.

We propose a technique that facilitates automated whitefly count based on computer vision.

1.2 Problem Statement

Agriculturalists at Namulonge Research Center have over the years adopted the traditional manual counting of whiteflies which solely depends on observers skill. The whole process is very tedious considering that the whiteflies are very small and volatile so it is possible to lose track of the numbers. In addition, a single cassava leaf usually has very many whiteflies on it which increases the difficulty in counting and consumes a lot of time. The fact that these flies reside on the back side of the cassava leaves makes the process all the more complicated because in the event of turning the leaf, the flies keep flying away which is really frustrating. The researchers handle large volumes of data from many sources and in such a case, data should be easy to collect manage and used to curb the spread of cassava mosaic disease but this has not been easy with the manual method. Coming up with a technique to automatically identify and count the whiteflies based on computer vision will be of great benefit to both the researchers and agriculturalists because it will ease and fasten their tasks.

1.3 Main Objective

The major objective of the study was to develop a real time mobile application that automates the process of detection and counting of whiteflies on cassava leaves using computer vision techniques.

1.3.1 Specific Objectives

The specific objectives of the study were:

- i. To carry out a maiden study on the literature of existing computer vision techniques used in object detection.
- ii. To carry out a preliminary study on the existing system used to identify and count whiteflies by agricultural researchers.
- iii. To design the system for the automated whitefly count.

- iv. To implement the designed system.
- v. Finally, to carry out the testing process of the implemented system.

1.4 Scope of the study

This project limited its self to a mechanism by which farmers place their basic camera phones over the leaf infested with whiteflies, the images are then diagnosed real time for the presence of white flies and a count is provided. The study is limited to the automation of whitefly count as it is part of a bigger project aimed at classifying the levels of cassava mosaic infection on cassava plants.

1.5 Significance of the study

The proposed system will provide the following advantages: Provide an efficient way of counting and tracking the spread of the white flies in the country at large. The project will also help the farmers and researchers to know the different prevalent infestation rates of the white flies in particular regions. The project once completed will also reduce on the expenses incurred by the research center especially in facilitating various field staff that are deployed in various regions to try and track the spread of the white flies. It will also provide room for further study in this area as well as provide reference literature for future scholars intending to build on this area of research.

Chapter 2

Literature Review

2.1 Introduction

The main purpose of this chapter is to present some general consensus on the theoretical support and previous empirical studies on the various computer vision and object detection techniques. In this respect, the chapter provides an empirical platform for use in the investigation of which training techniques should be used in computer vision and which computer vision libraries are most appropriate.

2.2 Computer Vision

Computer vision is the science and technology of machines that see, where see in this case means that the machine is able to extract information from an image that is necessary to solve some task. As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, or multi-dimensional data from a medical scanner [?]. As a technological discipline, computer vision seeks to apply its theories and models to the construction of computer vision systems. Examples of applications of computer vision include systems for:

- Controlling processes (e.g., an industrial robot or an autonomous vehicle).
- Detecting events (e.g., for visual surveillance or people counting).
- Organizing information (e.g., for indexing databases of images and image sequences).

- Modeling objects or environments (e.g., industrial inspection, medical image analysis or topographical modeling).
- Signal processing and pattern recognition.

Computer vision is closely related to the study of biological vision. The field of biological vision studies and models the physiological processes behind visual perception in humans and other animals. Computer vision, on the other hand, studies and describes the processes implemented in software and hardware behind artificial vision systems. Interdisciplinary exchange between biological and computer vision has proven fruitful for both fields. Sub-domains of computer vision include scene reconstruction, event detection, video tracking, object recognition, learning, indexing, motion estimation, and image restoration.

2.2.1 Fundamentals of Computer Vision

Computer vision has experienced growth with its applications expanding in diverse fields: medical diagnostic imaging; factory automation; remote sensing; forensics; autonomous vehicle and robot guidance [?]. Computer vision is the construction of explicit and meaningful descriptions of physical objects from images. The term which is synonymous with machine vision embodies several processes.

Images are acquired with a physical image sensor and dedicated computing hardware and software are used to analyze the images with the objective of performing a predefined visual task. Machine vision is also recognized as the integrated use of devices for non-contact optical sensing and computing and decision processes to receive and interpret an image of a real scene automatically. The technology aims to duplicate the effect of human vision by electronically perceiving and understanding an image [?].

2.3 Object Detection

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings or cars) in digital images and videos [?]. Well-researched domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance. Examples of object detection tasks are face, eye and nose detection as well as logo detection.

Given an image, object detection is to determine whether or not the specified object is present, and, if present, determine the locations and sizes of each object. Kalinke also suggested that object detection and recognition mainly focuses on:- Representation: How to represent an object. Learning: Machine learning algorithms to learn the common property of a class of objects. Recognition: Identify the object in an image using models.

2.4 Review of Object Detection Techniques

2.4.1 Haartraining

This is a technique used to train the boosted cascade object detector. It uses haar-like features and is used to train classifiers for image detection systems. OpenCV refers to this detector as the Haar classifier because it uses Haar features or, more precisely, Haar-like wavelets that consist of adding and subtracting rectangular image regions before thresholding the result. We also note that the training (`createsamples ()`), `haartraining ()` and detecting (`cvHaarDetectObjects ()`) code works well on any objects (not just faces) that are consistently textured and mostly rigid.

Sample Creation

For training, a training sample must be collected. There are two sample types i.e. Negative samples and Positive samples. Negative samples are non-object images. They are taken from arbitrary images and must not contain object representation. Positive samples are object images. A positive image contains the target object which you want machine to detect.

Haar Like Features

A recognition process can be much more efficient if it is based on the detection of features that encode some information about the class to be detected. This is a description of a case of Haar-like features that encode the existence of oriented contrasts between regions in the image [8]. A set of these features can be used to encode the contrasts exhibited by a human face and their special relationships. Haar-like features are so called because they are computed similar to the coefficients in Haar wavelet transforms.

The object detector of OpenCV has been initially proposed and improved by Rainer Lienhart [?]. First, a classifier (namely a cascade of boosted classifiers working with haar-like features) is trained with a few hundreds of

sample views of a particular object (i.e., a face or a car), called positive examples, that are scaled to the same size (say, 20x20), and negative examples - arbitrary images of the same size.

Performing haartraining for face detection insists that after a classifier is trained, it can be applied to a region of interest (of the same size as used during the training) in an input image.

The classifier outputs a "1" if the region is likely to show the object (i.e., face/car), and "0" otherwise. To search for the object in the whole image one can move the search window across the image and check every location using the classifier. The classifier is designed so that it can be easily "resized" in order to be able to find the objects of interest at different sizes, which is more efficient than resizing the image itself. So, to find an object of an unknown size in the image the scan procedure should be done several times at different scales [?].

2.4.2 Knowledge and Motion Based Methods

Mobile and Stationary Computer Vision based Traffic Surveillance Techniques for Advanced ITS Applications uses Knowledge-based methods that employ a priori knowledge to hypothesize vehicle locations in an image [13]. The useful information can be used includes symmetry, color, shadow, corners, horizontal/vertical edges, texture and vehicle lights. The Knowledge-based methods discussed above use spatial features to distinguish between vehicles and background.

Another important cue that can be used is the relative motion obtained via the calculation of optical flow. Approaching vehicles at an opposite direction produce a diverging flow, which can be quantitatively distinguished from the flow caused by the car ego-motion. On the other hand, departing or passing vehicles produce a converging flow. In contrast to dense optical flow, sparse optical flow utilizes image features, such as corners, local minima and maxima, or Color Blob. Although it can only produce a sparse flow, feature based method can provide sufficient information for vehicle detection.

The above methods also use the statistic model-based algorithms to learn the characteristics of the vehicle class from a set of training images which capture the variability in vehicle appearance. Usually, the variability of the non-vehicle class (background) is also modeled to improve performance.

2.4.3 Background Subtraction Method

Research on Computer Vision for the OU Fish Visualization describes this method as a generic algorithm based on motion tracking and is used to

address the problem of detecting moving objects [14]. The first stage of the process takes an incoming frame and subtracts it from a background image. This is then thresholded and passed to a blob filter that keeps only shapes of a certain size. The result is a rough segmentation of fish (object) from the video image. As a final step, the segmentation mask is fed back in to the system to update the background.

In other words, it is used as a motion mask for preventing foreground objects from being added to the background image. When a new frame comes in to the system, the motion mask is used to pull out only background pixels and those pixels are accumulated into the background image. The motion mask feedback is not flawless however and in some cases can result in motion mask creep. This is where the motion mask prevents any new information from being added to the background making the background information more and more out of date.

If the motion mask is identifying a fish, this is not a problem since identifying a fish is the goal, but if the motion mask is erroneously marking a location where some other type of movement is occurring such as water motion or changes in lighting, then the erroneous segmentation will actually be amplified as the segmented region grows.

Fortunately the fact that the fish are constantly moving can be used to prevent this kind of mask creep. If a pixel hasnt been updated for a given period of time because it has been constantly blocked by the motion mask, the motion mask is overridden and the pixel is updated anyway. If the pixel the next new pixel at this location is still different enough (meaning its likely a fish), then it will still be detected as a motion pixel. When added to the system, this timeout feature vastly improved the segmentation results by reducing the number false positives.

2.4.4 Template matching approach

Template matching approach is widely used in image processing to localize and identify shapes in an image. In this approach, one looks for parts in an image which match a template (or model) [?]. This technique is majorly used in visual pattern recognition, where one compares the template function to the input image by maximizing the spatial cross-correlation or by minimizing a distance, that provides the matching rate.

The strategy of this approach is: for each possible position (in the image), each possible rotation, or each other geometric transformation of the template, compare each pixels neighborhood to this template. After computing the matching rate for each possibility, select the largest one that exceeds a predefined threshold. The biggest disadvantage to this technique is that it

is a very expensive operation while dealing with big templates and/or large sets of images.