Automatic grain classification

CS39440 Major Project Report

Author: Kacper Dziedzic ([ktd1@aber.ac.uk](mailto:ktd1@aber.ac.uk))

Supervisor: Dr Fred Labrosse ([ffl@aber.ac.uk](mailto:ffl@aber.ac.uk))

28th April 2024

Version: 1.0

This report was submitted as partial fulfilment of a BSc degree in Computer Science

(G400)

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Wales, U.K.

Declaration of originality

I confirm that:

* This submission is my own work, except where clearly indicated.
* I understand that there are severe penalties for Unacceptable Academic Practice, which can lead to loss of marks or even the withholding of a degree.
* I have read the regulations on Unacceptable Academic Practice from the University’s Academic Registry (AR) and the relevant sections of the current Student Handbook of the Department of Computer Science.
* In submitting this work, I understand and agree to abide by the University’s regulations governing these issues.

Name Kacper Dziedzic

Date 28th April 2024

Consent to share this work

By including my name below, I hereby agree to this project's report and technical work being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Name Kacper Dziedzic

Date 28th April 2024

Acknowledgements

I would like to give a big thank you to Dr Fred Labrosse for guiding me through the project, for giving me plenty feedback, supplying me with the necessary hardware, and criticism whenever needed which helped form the final project. Without his help and support this project would not have been possible.

I would also like to give a quick thanks to Marc Loosely for supplying me with the majority of the data I had available for the project.

Abstract

There are biologists who are using grain for research purposes and currently need to manually sort it by hand. The purpose of this project is by using large amount of supplied data, to sort this grain into the 3 different categories: wholegrain, groats and broken grain. The goal of this project is by using computer vision to identify and detect this grain, using programming to extract values from the individual grains, store this data and then finally, train a machine learning model to detect the different types of grain. This task would provide valuable insight on what features, values and other things that can be extracted out of the image of grain can be used to raise the machine learning models accuracy and effectiveness. This project would involve computer vision, programming, data mining and data analysis to create a final system that could *potentially* be implemented onto a real-world situation. The data and research acquired, would do a huge service to the next step of full automation using machinery.

Contents

1 Background, Analysis & Processes

* 1. Background
  2. Analysis
  3. Processes

2

1. References
2. Appendices

A Use of Third-Party Code, Libraries and Generative AI

Project Explanation

Introduction

Automatic Grain Detection

Image Preprocessing

OpenCV

Order of Operations

Changing Brightness + Contrast

Canny Edge Detection

Colour Blob Detection

Binary Thresholding

Otsu Thresholding

Watershed Algorithm

Image Features

Colour Models

RGB

HSV

HSL

YCrCb

Conclusion

Area + Perimeter

Compactness

Hu Moments

Rectangularity

Circularity

Aspect Ratio

Potential Future features

Feature Extraction Methods

Impacts Of Certain Features

Machine Learning Models

Machine Learning Models Attributes

Accuracy progression

Programming

Appendices

Section 1 Background Analysis & Processes

* 1. Introduction

Grain is a major commodity, it is a large umbrella term encompassing many different types of crops, including rice, wheat, maize, oats, etc. Grain is a major source of food for our society as it can be processed into many different types of products and therefore needs to be studied in many ways, for example: safety or enhancing genetics for bigger harvests. For my task I have been given oats, the goal for this task is to take these oats and using computer vision techniques, distinguish between them automatically.

This task is to allow biologists to use machine learning and computer vision to automatically sort hundreds of grains automatically using machinery, removing the need for manual sorting. This project has several components, including computer vision, data mining, data analysis and real-world application using a JeVois camera.

* 1. Tools Used

1.2.1 Python

This project is completed in python, python is a great language for machine learning, data mining and all-around data science while being very quick to write and simple to use. For this project I am using PyCharm Community Version by JetBrains as it is a free version of PyCharm suitable for educational or recreational purposes. It contains a very modern UI allowing me to complete work at a faster pace than in the default python IDLE.

I am using Python version 3.10.

1.2.2 Python Packages

There are several crucial packages that I am using throughout my project, they provide many utilities, methods and visualizations that help within my project.

NumPy 1.26.4

NumPy is a package dedicated to scientific computation within Python. It contains powerful arrays which I use within the project as well as mathematical functions that are more efficient than iterating several times.

OpenCV-python 4.9.0.80

OpenCV is a computer vision module which gives several computer vision functionalities such as image preprocessing, several algorithms within computer vision, many image operations, editing and display windows. It is crucial for data extraction from test images.

Python-weka-wrapper3 0.2.14

Weka wrapper 3 is a python package which takes the functionality of Weka the open-source machine learning software written in java, but it applies it to python by allowing for its use within a java virtual machine. This package will allow me to implement machine learning creation actively within the code.

Python-javabridge 4.0.3

A package which allows for python to communicate with java using a java virtual machine. Only use is for weka-wrapper-3 to communicate with my python code.

SciPy 1.13.0

A package for scientific computing within Python providing fundamental algorithms and other foundational code or methods for use within the code.

Matplotlib 3.8.4

A package dedicated to data visualization, contains any graph I would ever need to use with a lot of customizability and ease of use. This project utilizes a lot of data making this crucial for data analysis.

1.2.3 Weka

Weka is an open-source machine learning software that can help with machine learning models by testing the efficiency of machine learning models, analysing the data distributions, testing the accuracy of machine learning models and generating these models. I will be using the Weka workbench as well as the python package weka-wrapper3 for automation.

1.2.4 GitHub & File Persistence

For my project a place to store the files using version control will be important. There are many situations that could occur and having a remote store of the project will be vital in keeping the files from corruption and deletion. Version control also has the added benefit of the ability to revert the files to an older version. I decided to use GitHub over GitLab due to it being a personal account which I will have access to later in the future. I also have more experience using GitHub making it a safer choice. In addition to this I am keeping the project files on 2 separate devices to lower risk of any file related problems.

*All used Software and Libraries licencing allows for educational use.*

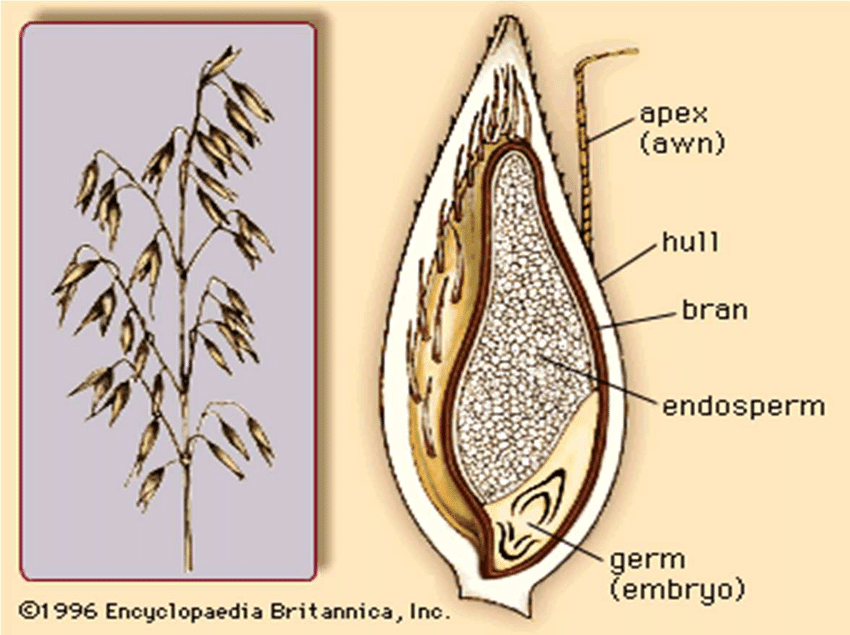
Analysis

Grain is a term used to describe a small, hard dry fruit, with or without an attached hull layer of any cereal & pseudocereal. The grain I am tasked with is oats and in this project the goal is trying to distinguish between 3 different types of oat grains.

*Figure 1* shows what the structure of a complete oat kernel looks like.

The first is a “Wholegrain” which is comprised of the endosperm, germ, bran and usually the brush, which together make the wholegrain. This project will include the Husk of the grain within this term.

*Figure 1 oat cross section*



Groats is defined as the original Wholegrain that has been peeled/hulled to reveal the bran and remove the husk completely.

Broken grain will be defined as any grain that has been broken in any way shape or form. This could include a small piece broken off or the grain split into two. Broken grain can still have the husk, but it is likely it won’t as it acts as a protective layer to the insides of the grain.

These 3 grain types have many clear differences when looking at them however this

This is a system that requires several independent steps that come together overall. Since the project is

to automatically detect and identify grain the first step is analysis of the training data that that had been provided. I was supplied with over 8Gb of raw .tiff images. According to *Figure 2*:



*Figure 2 training data, wholegrain, groats, broken grain*

It clearly shows that the grains are coloured bright yellow/brown on a dark background with some unnecessary parts on the image. By checking the properties of the images, they are all exactly 2550 x 3509 pixels and all precisely 25.6mb large with no compression with the section containing the grains being in the exact same area and having the exact same size for every image.

The fist goal of the project is to get contours of every individual grain, trying to account for overlapping.

The first step is to investigate image preprocessing and image modification. To reduce processing time and unnecessary data, the initial step would be to crop the image to the desired area. From looking at all the images the area containing the grains is consistently: (160-1240) x & (2350-3450) y. This removes a large portion of the image, reducing the total pixels available on the image which ultimately would speed up processing time drastically. Maximising the efficiency of the program without reducing the quality of the data is key to data mining and this simple cropping process reduces the total number of pixels from approx. 9 million to approx. 1 million.

Just by looking at Figure 1 we can see that the bright grains contrast the black surface. This can be applied using computer vision to separate the grains from the background using the large contrast.

Process

Before starting any programming, I needed to establish a methodology that would suit my work and suit me personally. Knowing this project would involve a lot of iterative programming, testing and then getting results or data mining I decided to research methodologies which involved creating a set list of things to do and completing them before moving on to another task.

Scrum and Kanban both had similarities to what I was searching for as Scrum generally involves doing initial research, then creating a broken-down list of tasks to work on to complete part of the project called the sprint backlog. This is done iteratively on a cycle which would allow for a planning phase, working phase and testing at the end of each cycle.

The Kanban methodology involves using a “Kanban board” which contains a complete list of tasks To Do, In Progress and Done. Kanban would help create a flow of tasks to follow in order with a good visual list showing how much has been done and how much is left to do.

Final decision was to go with an adapted Kanban methodology which includes adding tasks to a work list whenever possible but including research and testing in periods between tasks. This would allow for a steady workflow with clear tasks to complete as well as periods of time to do research on areas I need to work on as well as testing to find errors and improve my program. The general structure of what the workflow would look like is, initial research on the topic, finding relevant packages and software to use, then creating the kanban board and putting anything that I know I need to do onto the board as soon as possible marking tasks as urgent if they are. The tasks would include programming, research, testing or any combination of the three. I would start with “urgent” tasks then moving onto less important tasks once the “urgent” ones are completed.

Testing is done throughout the entirety of the project due to the nature of this task not having much to do with having a final program but finding what data is most valuable and impactful to the results of the machine learning algorithms.

*Figure 3* shows an example of my kanban board mid project.

INSERT KANBAN HERE

References

Appendices

1. Use of Third-Party Code, Libraries and Generative AI

No generative AI tools have been used for this work.