Bournemouth University

FACULTY OF SCIENCE & TECHNOLOGY

BSc (Hons) [Degree Title]

May 2021

What's Wrong With My Crop? Using Convolutional Neural Networks to Detect Crop Defects

by

Ryan Syme

Faculty of Science & Technology

Department of Computing and Informatics

Final Year Project

Abstract

[The text within the square brackets must be deleted along with the square brackets when finalising your own abstract.

The abstract for an undergraduate dissertation should be between 200 - 350 words.

Arial, Normal, 11pt with 1.2 or 1.5 line spacing should be used. The text in this part has 1.5 line spacing.

An abstract is a brief, accurate and comprehensive summary of the entire dissertation. It is the first thing to be read by your examiners to help them know the brief content of the dissertation. It also serves as a “sales pitch” to form the first impression of your work.

A good abstract should be accurate, self-contained, concise, specific and clear. A quick way to assess the quality of your abstract is to check whether it answers the questions why, how, what and so what.

Researching the efficacy of using CNN’s (Convolutional neural networks to identify crop defects) and creating a suitable platform for users to interact with the network.

It is easier to write the Abstract the last.]

Dissertation Declaration

[The text within the square brackets must be deleted along with the square brackets when finalising your declaration.

Note if your project is CONFIDENTIAL because of your client, you will need to adapt this declaration based on the agreement between you and your client accordingly. Do not forget to state the name of your client clearly. You must contact and inform Project Coordinator if your project is CONFIDENTIAL.]

I agree that, should the University wish to retain it for reference purposes, a copy of my dissertation may be held by Bournemouth University normally for a period of 3 academic years. I understand that once the retention period has expired my dissertation will be destroyed.

Confidentiality

I confirm that this dissertation does not contain information of a commercial or confidential nature or include personal information other than that which would normally be in the public domain unless the relevant permissions have been obtained. In particular any information which identifies a particular individual's religious or political beliefs, information relating to their health, ethnicity, criminal history or sex life has been anonymised unless permission has been granted for its publication from the person to whom it relates.

Copyright

The copyright for this dissertation remains with me.

Requests for Information

I agree that this dissertation may be made available as the result of a request for information under the Freedom of Information Act.

Signed: ____

Name: [Your name]

Date: [Date of signing this declaration]

Programme: [Your degree title]

Original Work Declaration

This dissertation and the project that it is based on are my own work, except where stated, in accordance with University regulations.

Signed: ______________________________

Name: [Your name]

Date: [Date of signing this declaration]

Acknowledgements

[The text within the square brackets must be deleted along with the square brackets when finalising your own acknowledgements.

Arial, Normal, 11pt with 1.2 or 1.5 line spacing should be used. The text in this part has 1.5 line spacing.

This is your opportunity to mention individuals who have been particularly helpful. Reading the acknowledgements in the past dissertations in the project library will give you an idea of the ways in which different kinds of help have been appreciated and mentioned.]

Contents

List of Figures

List of Tables

Chapter 1 - Background and Lit Review

1.1 Context

the application area / industry / domain

1.2 Technological Aspects

lorem ipsum

Chapter 2 - Introduction

2.1 Context

With the increased availability of smartphones Sta (2021), digital cameras Ima (2021) and Internet access Wik (2021) Glo (2021). Coupled with the increased interest in home food cultivation Goo (2021) and the large number of people reliant on food grown in smallholdings Walpole and Hutton (2013). The ability to identify defects with crops using technology has potential to be impactful to many people.

In 2009 a study was conducted using deep learning to identify three different disease classes on rice plants. The results showed over 70% classification accuracy on 50 sample images. Anthonys and Wickramarachchi (2009)

Seven years later and there have been great successes in identifying crop disease with CNN's. In this study deep learning gave 99.35% accuracy on a held-out test set over 14 crop species and 26 diseases (or absense thereof). Mohanty et al. (2016). This study utilized two established CNN architectures, namely AlexNet Krizhevsky et al. (2012) GoogLeNet. Szegedy et al. (2015) With GoogLeNet achieving a higher F1 score in almost all cases.

2.2 Problem Definition

As it stands there are currently (09/02/2021) no easily found [1] web interfaces for interacting with a crop defect identification service.

Although there exists very capable publicly available image classification networks. Yan (2021) There is no bespoke application catering soley to crop defect identification, that has the benefit of providing recourse information to the user.

2.3 Proposed Solution

To provide a web service that interacts with a convolutional neural network (CNN) backend to diagnose crop defects such as, nurturing problems e.g. lack of water/nitrogen/C02, too hot/cold.

[1] (i.e. not present in the first 3 pages of a google search for 'crop defect identification' and 'What's wrong with my crop')

And external threats such as crop disease/pest infestation. The interface will be simple and intuitive as possible. The UI should minimise points of interaction and streamline the process of uploading a crop image to be analysed. The web service will return information regarding the percentage likelihood of each kind of crop defect, including images that are of similar nature to the one analysed.

2.4 Aims and Objectives

These should be SMART with clear success criteria defined

- have a working REST API. The API will provide information regarding the likelihood of each kind of crop defect, when served an image via a link to a relational database. In addition to other metrics such as similar images and time to compute. The API will be robust enough to handle the receipt of erroneous requests.
- A python backend that will handle image classification using a CNN.
- The CNN should be able to classify at least 7 different defects across at least two different plant species.
- The CNN should acheive at least 80% accuracy at classifying all different classes of defect in a held out test set that contains an equal number of each class.
- A UI that will allow the user to upload an image to be analysed.
- The UI will display information regarding the likelihood of each kind of possible defect.
- To display the relevant images that fit the description of the most likely defects.
- To display recourse information to rectify the defect.
- Collecting, cleaning and pre-processing the image data.
- Artificially grow the dataset by performing translations/rotations/adding noise to the images to make the training data more comprehensive.
- Include regularisation techniques to the NN to prevent overfitting.

2.5 Risk Table

(ID, name, likelihood, impact, control mechanisms / accept)

Table 2.1: Risks Table

ID	Name	Likelihood	Impact	Control Mechanism
1	Improper Time Management	med/low	high	Follow the Gannt chart
2	HDD/storage failure	low	high	All work will be backed
3	Illness/Injury	med	med	Should the need arise I will apply for an e
4	RSI (repetetive strain injury)	med	low	Work with proper postureand set up workstation p
5	Eye strain	med	low	Ensure room is well lit when working
6	Incorrect Task Prioritisation	med	med	Iteratively re-asses the work being done and co
7	Postural problems	med	low	Work with proper postureand set up workstation p

2.6 Overview

Introducing rest of dissertation (with cross references to sections)

Chapter 3 - Methodology

3.1 Project management methodology

I will use a cyclical, evolutionary method. This will involve:

- Requirements elicitation.
  - This involves determening the needs of the user and defining requirements to meet those ends.
- Feature design (UI).
  - Features will be designed at first using wireframe models. Then on later iterations, colour and shading will be added alongside further usability considerations such as highlight on hover etc.
- Feature implementation research.
  - This step involves determining the apropriate technologies and libraries to achieve the design. This is nececarry to realize the constraints that are imposed by the implementation method and know to what extent the design is feasible.
- Feature implementation.
  - Writing the code to create the feature.
- Feature testing.
  - Initially testing will be done manually with valid values until later iterations whereby extraneous values will be introduced. Once the feature is in it's final iterations a unit test will be introduced.
- Evaluation.
  - Does the feature meet the requirements and fulfill the needs of the user?

This workflow will consist of a single cyclical workflow, with two nested "sub workflows" whereby upon completion of a step, it is sometimes nececarry to loop back on oneself to perform futher refinement. As illustrated by the diagram below. Throughout the project the focus of the workflow will shift as illustrated by the diagram below.

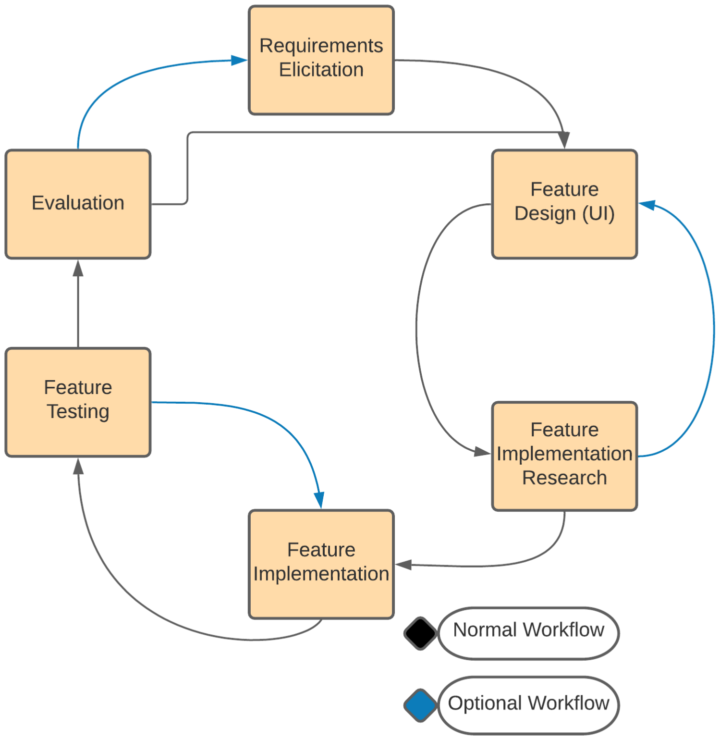

Figure 1: Development Lifecycle

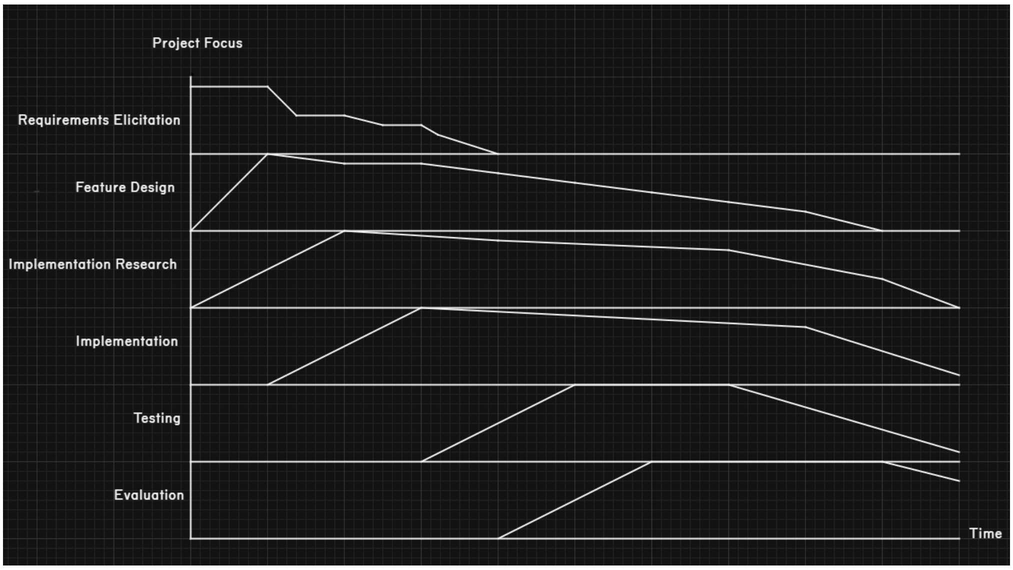

Figure 2: Project Focus Over Time

3.2 Evaluation Design

(what method(s), used how, with what and how many participants?)

3.3 Requirements Elicitation

How will requirements of the software be determined.

3.4 Feature management

To track the creation and completion of features, a Kanban board will be used. This will include columns for 'To do', 'Doing' and 'Done'.

3.5 Design Methods

- Requirements Elicitation
  - To better conceptualize the needs of the user. Use case diagrams and activity diagrams will be utilized.
- User Interface
  - Wireframes will be uitilized to establish interface element placement i.e. layout.
  - More detailed mockups will be created when the earlier wireframes are constructed as prototypes and the concept is proved acheivable.
  - A colour picker will be utilized to define the colour scheme.
  - In later iterations of the design, once there is a functioning UI, usability will continue to be refined with the help of existing usability research, to guide the usage of font/colour/highlight on hover/font size etc.
  - Additionally once a desktop friendly layout has been established, work will begin on optimizing a version for mobile.
- Back-End
  - UML will be used to show the overall design of the system through structural diagrams. These will show the interfaces of the classes and how they will interact with one another.

3.6 Testing methods

3.7 Version control

I will be using Git and Github. This will allow the creation of branches to explore experimental parts of the soloution space without disrupting the progress of the main branch. If the experimental implementation is successfull it will be merged with the main branch. It also allows the development of features in paralel, with any conflicts in their implementation being resolved at the merge stage. The inclusion of a remote repository allows for work to continue on a seperate machine if nececarry and later be synced with the local main branch.

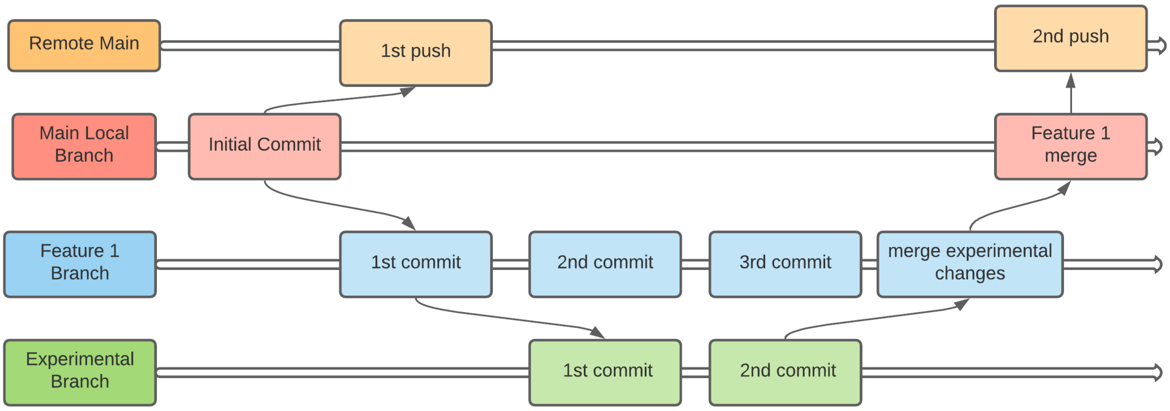

Figure 3: Example Workflow To Highlight Branch Usage

3.8 Evaluation methods

The main method that will be utilized to determine the quality of the user interface will be the System Usability Scale (SUS) which can be seen here. The data will be collected via online questionare. Aditional metrics that focus on the evaluation of the CNN will be:

- Time to train the network on available hardware
  - The constraint here being if the network cannot be trained on the available hardware in under sixteen hours. Purely for practical considerations.
- Accuracy of CNN predictions. (which will be most effective when there are equal numbers of samples belonging to each class) $Accuracy = \frac{CorrectPredicitons}{TotalPredictions}$ Else if the samples are sqewed, the network could be a faliure at detecting a specific under-represented class, yet still score high accuracy.

- Precision. This is the number of correctly predicted images out of all predictions of that class. $Precision = \frac{CorrectlyPredictedforClass}{TotalPredictedforClass}$ The network is precice for a class when the predictions it does make are correct. Precicion cannot be used in isolation due to the fact that the network can have a high precicion for a class but still fail to identify the majority of images for that class. Succeeding soley on the fact that the images it has classified are correct.
- Recall. Is the correct number of predictions for a class out of the number present of that class. $Recall = \frac{CorrectPredictedforClass}{No.PresentForClass}$ This metric can also not be used in isolation due to the fact it does not take in to account the number of false positives. i.e. The number of images incorectly classified as the class in question. For example, if an image dataset contained three classes A, B, C, and the classfiier labeled all images A. The recall for A would be 100 percent.
- F1 score. This metric tries to find the balance between precision and recall and can be expressed as $F1 = 2 \times \frac{1}{\frac{1}{precicion} + \frac{1}{recall}}$

3.9 Initial Designs

Firstly I have created a wireframe UI

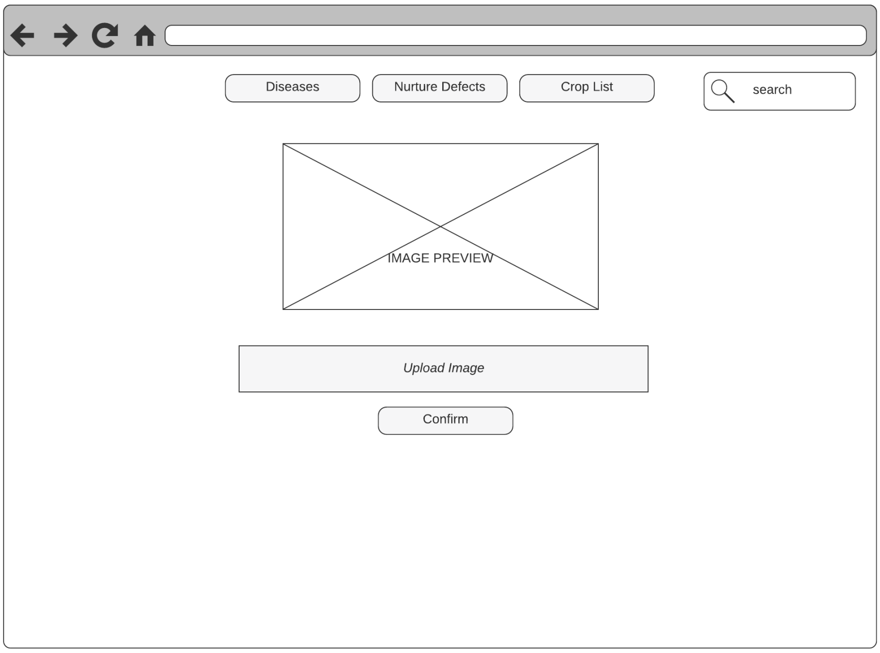

Figure 4: Homepage Wireframe

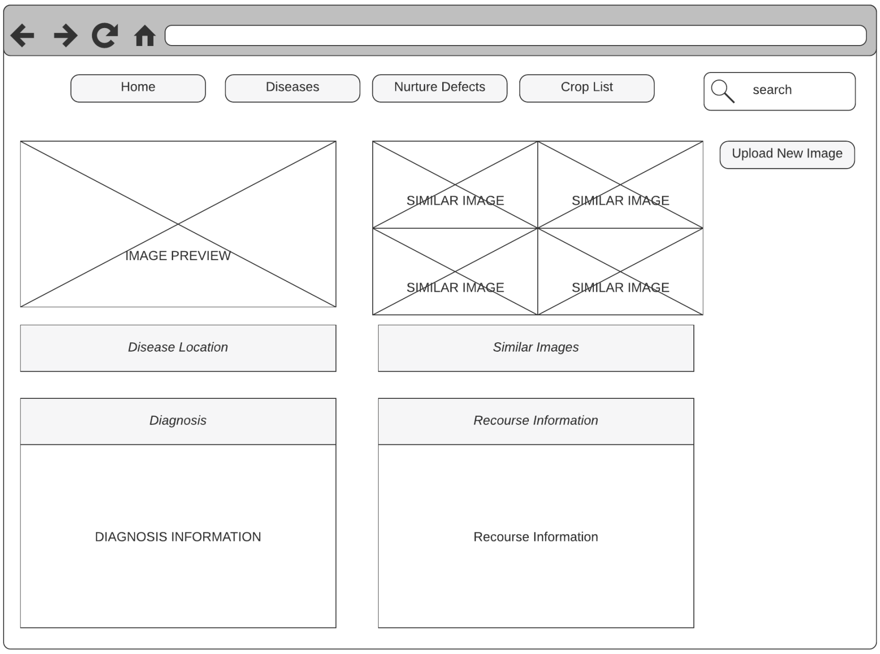

Figure 5: Defect Information Wireframe

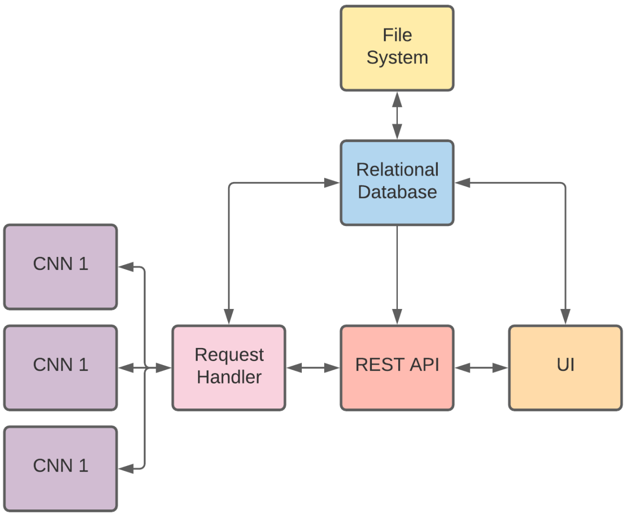

Figure 6: System Overview

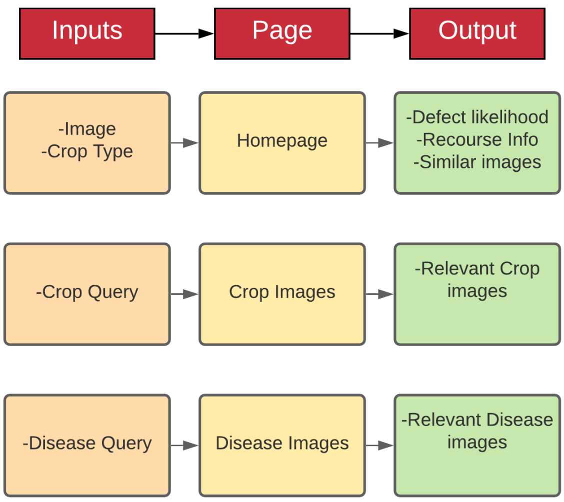

Figure 7: Input/Output overview

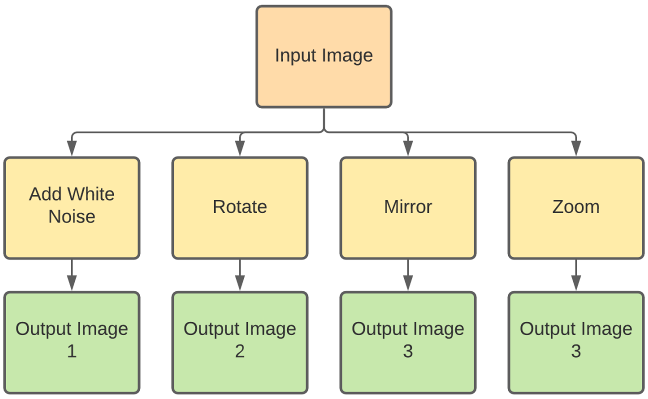

Figure 8: Input/Data Augmentation Methods

3.10 Employed Technologies

- Vue.js
- Bootstrap
- MySql
  - The SQL database will mainly be responsible for storing filepaths to images. This gives the ability to use an SQL statement to fetch certain groups of images for display or training.
  - The database will also store recourse information.

- Jupyter Notebook
- Tensorflow Keras
- numpy

3.11 Requirements

Must Have

- Ability for user to upload an image
- CNN that is capable of classifying at least 2 different defects across 2 different plant species.
- An API that allows communication from the UI to the CNN
- API must be able to receive images.
- API must return defect information.
- API must return recourse information.
- API must return relevant images to the detected defect.

Should Have

- A page to allow users to see a gallery of images sorted by defect type.
- A page to allow users to see a gallery of images sorted by crop type.

Could Have

- Ability for users to add additional information about the crop to determine the defect.

Won't Have

3.12 Testing and Implementation details

3.13 Justification of Implementation Choices

Chapter 4 - Results and Discussion

4.1 Main Results

lorem ipsum

4.2 Evaluation Results

lorem ipsum

Chapter 5 - Conclusion

5.1 Section One

a dissertation is a substantial document, it is convenient to break it up into smaller pieces. In this template we therefore give every chapter its own file. The chapters (and appendices) are gathered together in `dissertation.tex`, which is the master file describing the overall structure of the document. `dissertation.tex` starts with the line

REFERENCES

2021. URL https://yandex.com/images/.

2021. ArabSat 5C - Internet by Satellite in Africa. URL https://www.globaltt.com/en/coverages-Arabsat5C_C.html.

2021. best vegetables to grow - Explore - Google Trends. URL https://trends.google.com/trends/explore?q=bestvegetablestogrow&date=all&geo=US.

2021. Digital Camera Market Share, Size, Trends and Forecast 2021-2026. URL https://www.imarcgroup.com/digital-camera-market.

2021. File:Internet users per 100 inhabitants ITU.svg - Wikipedia. URL https://en.wikipedia.org/wiki/File:Internet_users_per_100_inhabitants_ITU.svg.

2021. • Smartphone users 2020 — Statista. URL https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/.

Anthonys, G. and Wickramarachchi, N., 2009. An image recognition system for crop disease identification of paddy fields in Sri Lanka. *ICIIS 2009 - 4th International Conference on Industrial and Information Systems 2009, Conference Proceedings*, 403–407.

Krizhevsky, A., Sutskever, I. and Hinton, G. E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. Technical report. URL http://code.google.com/p/cuda-convnet/.

Mohanty, S. P., Hughes, D. P. and Salathé, M., 2016. Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science*, 7 (September), 1419. URL http://journal.frontiersin.org/article/10.3389/fpls.2016.01419/full.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Walpole, S. J. R. A. B. C. S.-H. B. B. H. S. M. M. A. F. M. B. M. G. S. C. C. N. U. R. L. J. M. G. M., M. and Hutton, J., 2013. Enabling poor rural people to overcome poverty Smallholders, food security, and the environment. Technical report. URL https://www.ifad.org/documents/38714170/39135645/smallholders_report.pdf/133e8903-0204-4e7d-a780-bca847933f2e.

Appendix A - Project Proposal

Appendix B - Ethics Checklist