Dandelion Weed Detection and Recognition for a Weed Removal Robot

Ibrahim Babiker

A Thesis

In the Department of

Mechanical, Industrial and Aerospace Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Applied Science at Concordia University

Montreal, Quebec, Canada

August 2020

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared,

By: **Ibrahim Babiker**

Entitled: Dandelion Weed Detection and Recognition for a Weed Removal Robot

And submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science

Complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Youmin Zhang	Chair
Dr. Weiping Zhu	External Examiner
Dr. Youmin Zhang	Examiner
Dr. Wen-Fang Xie	Thesis Supervisor

Approved by: Dr. Mamoun Medraj
Chair of Department of Graduate Program Director

Date

Dr. Mourad Debbabi
Dean, Faculty of Engineering and Computer Science

Abstract

Dandelion Weed Detection and Recognition for a Weed Removal Robot

Ibrahim Babiker

Current research in agricultural weeding automation attempts to develop accurate methods of distinguishing between crop and weed. Consequently, the use of computer vision has become a cornerstone in these endeavours. Some recent methods employ pattern recognition techniques that involve hierarchical feature groupings. The application generally applies some form of machine learning. Furthermore, using convolutional neural networks (CNN), many techniques implement complex architectures that not only classify but also detect and locate objects. These detection problems generally involve datasets taken under artificial or controlled lighting conditions where foreground elements (i.e. weed and crop) are easily distinguishable from the background (usually soil) by virtue of their distinct hue and textures. Plant overlap is generally limited to being between foreground elements. The research in this thesis addresses the challenges overlooked by agricultural weeding by focusing on weeding in lawn grass with two distinct approaches. First, a pattern recognition methodology is developed to distinguish dandelion weed centers from grass using the morphological attributes of binary (black-and-white) regions. This method is tested in lab settings with both artificial weeds and grass. However, practical limitations include a fragile performance in real-world applications in the field and a heavy reliance on parameter calibration. Next, a machine-learning approach is developed to address the shortcomings of the prior approach

as well as to deal with the challenges specific to weeding in a domestic setting. A five-step process involving CNN structures proves successful at accurately detecting dandelion weeds within grass and other lawn vegetation. Extensive tests have been carried out on a wide array of real work images and the results demonstrate that the developed algorithm can detect and recognize dandelions in the grass within a reasonable range of natural lighting conditions.

Acknowledgements

This research project was conducted in the department of Mechanical, Industrial, and Aerospace Engineering at Concordia University under the supervision of Dr. Wen-Fang Xie. Many thanks are due for her keenness in the development of my research credentials and her guidance in terms of navigating my master degree. Furthermore, this project would not have existed without her help and financial support.

Additionally, the indispensable help from colleagues Mr. Mikail Arani, Mr. Tzvi Filler, and Mr. Daniel Lofeodo cannot be neglected and it is with much warmth that I offer them my gratitude. Further guidance and collaboration from Dr. Guangyi Chen are met with equally warm thanks.

Most importantly, I would like to thank my tireless parents, Mrs. Tahani Abbas and Mr. Fathelrahman Babiker, for their unwavering support in my studies and in life as a whole. Without them, none of this would be possible.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Overview

The following sections provide a brief but necessary overview of topics relevant to this thesis. The project work encompasses elements from the fields of robotics, computer vision, and artificial intelligence.

1.1.1 Relevant Context

A short introduction of the mobile robot this work intends to equip is necessary. Firstly, it must be noted that the scope of the robot's involvement in this project is limited to its requirement of a sensor. The building, programming, and control of this robot are not explored in this thesis. However, the robot's general functionality will be briefly summarized insofar as to explain the necessity of the sensor.

A mobile robot was initially designed for the mechanical engineering capstone project of the 2017-2018 academic year. The objective of this robot was to autonomously locate and subsequently remove dandelion weeds in grass. The robot possessed a webcam as its principal sensor and a drill with the intention of extracting and destroying the weed. The robot was successfully built. However, it did not achieve full functionality because the weed sensing component was not completed.

The work in this thesis has, as goal, to complete the sensing component of the aforementioned robot and enable future projects to build upon the control derived from the proposed sensor. As will be seen by later content, the sensing attribute constitutes an extensive research project in its own right.

Finally, under the direction of this project's author, another capstone team has redesigned the robot by moving its drill-arm assembly to the chassis center. This decision was made for future integration of the proposed weed detecting algorithm into a control scheme where the robot must simply align its own center with the dandelion weed center before mechanical extraction. **Fig. 1** displays the original and current robots. The colour image on the left is the mobile robot from the original capstone project. The drawings on the right belong to the current iteration. Note the drilling arm has been moved to the center of the chassis.

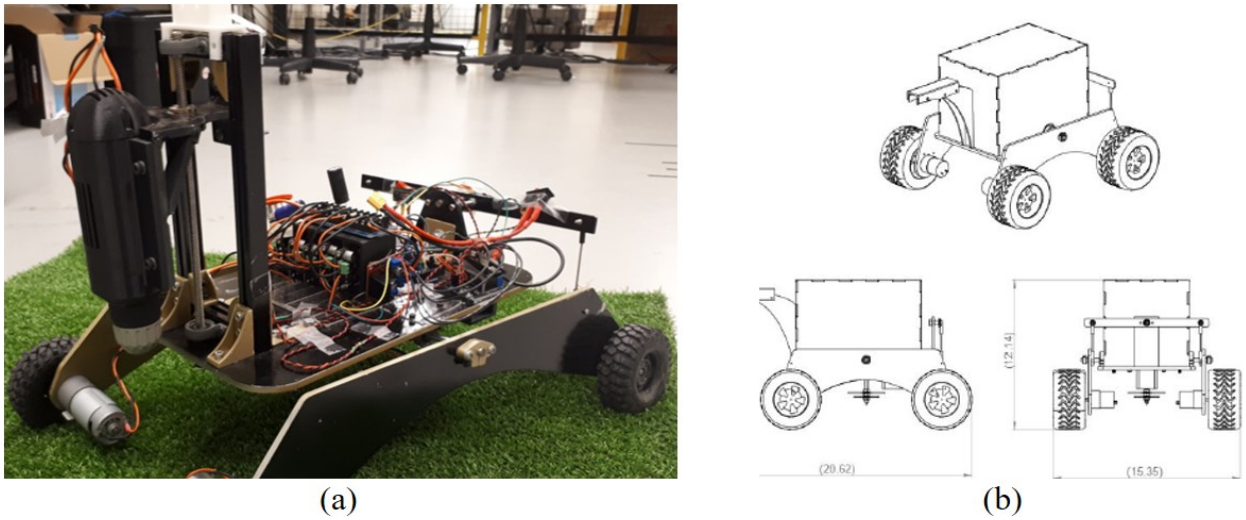


Fig. 1. (a) Previous weed removing robot. (b) Current weed removing robot [1]

1.1.2 Robot Sensing

Robotics and automation have become increasingly important in modern industry practices associated with manufacturing, production, aerospace, automotive vehicles, and agriculture. Traditionally, robots can be placed into two distinct categories: industrial robots and mobile robots.

Industrial robots are often comprised of articulated arms of up to 6 degrees of freedom (DOF) mounted at fixed reference frames whereas mobile robots are of a generally wheeled configuration allowing them to move and navigate untethered to a fixed point in space. Both types require some form of sensing in order to interact with their respective environments.

Improving robot sensory capabilities is the goal of a significant portion of research in the field of mechatronics and automation. Sensing encompasses the fields of vision, hearing, touch, and movement amongst other things. Currently, machine vision is an important area of sensory development research.

1.1.3 Machine Vision

Machine vision provides robots with powerful human-like sensing at a relatively low cost. Generally, some type of camera or combination of cameras is sufficient to gather relevant data for a variety of automated tasks. Much of machine vision research focuses on pattern recognition algorithms, precision measurement in 2D and 3D, path and trajectory planning in 2D and 3D, object detection, and so on.

Machine-learning's renewed popularity within the domain of artificial intelligence has injected much energy into generating intelligent object classification and detection frameworks for machine vision. In particular, research in deep learning networks has made leaps and bounds in terms of developing powerful architectures that perform in some cases better than humans in object classification tasks [2].

1.1.4 Convolutional Neural Networks

A notable element of intersection between machine vision and deep learning algorithms is the class of convolutional neural networks (CNN) which have a variety of applications in pattern recognition, linguistics, medicine, and even finance. CNNs possess properties that enable them to

overcome common difficulties associated with object recognition. These properties include a large and adjustable learning capacity and an ability to acquire and use prior knowledge obtained from a training set [3, 4]. They use far less parameters than common forward neural networks which results in easier training and similar performance to their computationally heavier counterparts [3]. Their high level of applicability in images is due in part to their use of 2D convolutions in strides along the whole image connected to non-linear activation functions while down sampling the output to subsequent convolutional layers. The network eventually processes the low-level features of the original input image to high-level features that take on more and more of an abstraction [5].

In 2012, the advent of AlexNet [3] dominated the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [6]. Subsequently, better performing CNN architectures were developed. Visual Geometry Group (VGG) [7] introduced smaller convolutional kernel sizes, deeper networks and the stacking of convolutional layers uninterrupted by pooling layers; GoogleNet [8] created the inception module architecture that involved simultaneously concatenating convolutions of varying sizes; and residual network (ResNet) [9] conceived pathways that skip immediately subsequent layers maintaining residual information along the network. All these models successively reduced the ILSVRC classification error rate to negligible amounts [10].

1.2 Motivation

The motivation behind this work is to develop a low-cost sensor capable of accurately detecting dandelion weed centroids within grass for eventual mechanical removal. Currently, several governments around the world have banned pesticides such as glyphosate. This move seems to follow a global trend of making environmentally conscious choices that aim to reduce and outdate the release of potentially harmful chemicals into our ecosphere.

Furthermore, the accessibility of machine-learning frameworks, their operation under open source licenses, and their relative ease of prototyping and fast implementation in recent years encourages this research in the direction of smart systems. The aim is to reduce and perhaps even eliminate the load on human operators involved in tasks that are at best relatively tedious and at worst possibly strenuous.

However, before any of what is hoped for can be accomplished, a proper sensor is most definitely required.

1.3 Contribution

The contribution of this thesis consists of tackling a specific problem that has not yet been exhaustively explored. As illustrated in the next chapter, current research largely concentrates on weed removal in agricultural settings where weeds must only be distinguished from crops. This project presents the development of novel methods that begin as heuristic pattern recognition approaches and progress to the semi-intelligent. It is hoped that the contents of this thesis will prove that the presented low-cost system can accurately locate dandelion weeds in the midst of grass and other lawn vegetation. Eventually, this will pave the way for its implementation in a weed removal robot.

1.4 Thesis Outline

Henceforth, the thesis is divided as follows. Firstly, a literature review pertaining to relevant studies on matters of shared interest is conducted. They are object detection in images in general and weed or plant identification and detection in agriculture and grass in particular. Secondly, an in-depth explanation of both proposed methods and their development is made, that is, the pattern recognition approach followed by the machine-learning approach in dandelion weed detection in grass. Thirdly, the experimental setup for both methods is demonstrated and key points such as

data collection methodology and data subdivision are highlighted. Fourthly, the results for each approach are presented, analyzed, and system limitations are discussed. Finally, a concluding note is offered and relevant future work is suggested.

Chapter 2: Literature Review

2.1 Object Detection with CNNs

A brief overview of object detection using CNNs is necessary to highlight the applicability of existing schemes with respect to the current problem of detecting dandelion weeds in grass.

As mentioned in the introduction, CNNs have proven themselves great tools for object classification tasks. A rudimentary interpretation is that they are good at determining whether images contain X or Y object. However, detecting multiple instances of the same or different objects in a single image and providing their locations is a different problem. This distinction has made necessary the development of several different CNN-involved object localization algorithms.

Generally viewed as first to the table, regions with CNN features (R-CNN) [11] debuted learned region proposals. The underlying principal of the innovation was employing the "recognition using regions" paradigm in the context of CNN architectures. This was done by producing a robust bag of overlaid regions and subsequently learning their weights using a CNN classifier.

R-CNN performed exceedingly well on the PASCAL VOC (Pattern Analysis, Statistical Modelling and Computational Learning Visual Object Classes) [12] dataset. Consequently, further iterations Fast R-CNN and Faster R-CNN [13, 14] came about. They were followed quickly by

other schemes: You Only Look Once (YOLO) [15] and Single Shot MultiBox Detector (SSD) [16]. The former two performed as their names suggest whereas the latter two compromised on accuracy to further reduce computational costs and increase prediction speed. The superior performance of these methods on the PASCAL VOC as well as other datasets quasi-eliminated the need for the traditional and more expensive sliding window method [10].

Now, the PASCAL VOC and similar object detection datasets such as COCO [17], SUN [18], KITTI [19], and INRIA [20], generally consist of scenes that are viewed from natural perspectives. A brief survey by the average human eye will quickly distinguish target objects from their background based on a variety of factors such as contrast with the foreground, texture, shape, contours, etc. Techniques like introducing labelled data for semantic segmentation (i.e. the attributing of each pixel with a class label) are more straightforward. However, the data relevant to this project consist of a birds-eye-view perspective of green dandelion plants within green grass amongst other generally green lawn vegetation.

Inherently, the problem of ours is far more specific and presents a different set of challenges in terms of object detection. Dandelion weed plants are not always fully visible –in fact, they are often hidden in the grass while only their leaves are visible in part or in full. Generally, these leaves protrude radially outwards from the roots. Furthermore, the view of these leaves is also often broken by overlap from neighbouring weed leaves, grass blades, and other vegetation. This complicates tasks such as the aforementioned semantic segmentation. **Fig. 2** will be left as a conclusion to this section.

Fig. 2. (Left) PASCAL VOC dataset. (Right) Dandelion Weed dataset

2.2 Weed Sensing in Agriculture

In agriculture, weed detection has been the topic of much research with the aim of developing systems that discriminate between crops and weeds while effectively locating the latter. Lottes et al. [21] created an advanced end-to-end CNN architecture that was able to both semantically segment crop and weed plants and segment and locate weed stems. To achieve this result, an encoder was paired with two decoders. The original image, captured in the red-green-blue (RGB) colour space and the near infrared (NIR), was preprocessed with Gaussian smoothing and then standardized between -5 and +5. This was input into the encoding CNN that extracted high-level features. The encoder employed a dense building block similar in structure to the Fully Convolutional DenseNet (FC-DenseNet) [22]. Subsequently, both decoders, also built using dense

units, decoded and up-sampled the high level features to generate, in one case, labelled plant segmentation maps, and in the other, plant stem maps and subsequently their locations.

Table 1: Comparison of the BoniRob and the Dandelion Weed datasets

BoniRob dataset [23]	**Dandelion Weed dataset**
Captured in RGB + NIR under artificial lighting conditions.	Captured in RGB under natural lighting conditions, often in full sunlight.
Plants are in stark contrast with dark brown soil background with sporadic and sparse amounts of grass.	Plants are in a near homogenous background of similarly coloured grass and other vegetation.
Substantial amount of inter-plant overlap [21].	Substantial to extreme amount of inter-plant overlap between weeds of different plants, grass blades and other lawn vegetation.

The advantages of this system are many; first, in that it is a completely end-to-end trainable CNN, which quasi-eliminates the need for heuristic parameter tuning; second, the encoder/dual-decoder architecture reduces computational costs significantly as features from one network suffice as the input for two distinct networks that accomplish, in turn, different functions.

At first glance, the above work may seem like the perfect solution to our problem. However, similar distinctions as those made in the previous section can be established between the BoniRob dataset [23] used for their study and the one used for this particular project. Table 1 above illustrates a few of the key differences while **Fig. 3** shows a few examples.

Fig. 3. (Top row) BoniRob dataset. (Bottom row) Dandelion Weed dataset

In [24], research was carried out with the similar goal of semantic segmentation and classification of weeds and crops. The presented method is in the form of an image-processing pipeline that innovates on classical mathematical morphology operations.

Mathematical morphology operations rely on structuring elements that perform operations such as erosion and dilation alone or successively in some type of sequence to remove noise and retain only the important binary (black and white) image features. When an object is eroded and then dilated, it is called 'opening', when the order is reversed, 'closing'. See **Fig. 4** for a brief example. A problem with morphology operations is that boundary details from the original shape are sometimes lost. A common solution is opening and closing by reconstruction [25, 26]. However, the requirement for structuring elements means that they do not solve problems associated with rotational invariance and decoupling of shape and size [24].

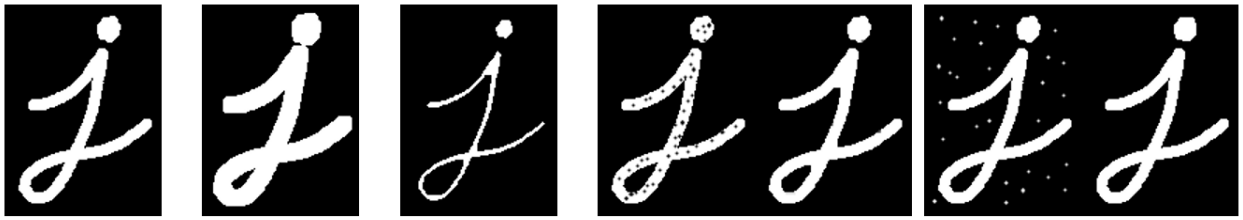


Fig. 4. (a) Original Binary Image (b) Dilated image (c) Eroded image (d) closing of image on left results in image on right (e) opening of image on left results in image on right [27]

Attribute morphology solves these problems by discriminating between desirable and undesirable regions based on an attribute criterion [28, 29]. For example, if the area of the input region is larger greater than a certain value, it will be kept. Bosilj et al. take this a step further by applying attribute morphology measurements to a range of greyscale thresholds on the input image thereby generating a hierarchical max-tree representation [29] of the image via its decomposed states (i.e. at successive image threshold values). Amongst other things, a key detail is their monitoring of the growth factor attribute, or more simply, the growth in area of a region within its decomposition range. Other attributes such as solidity, eccentricity and circularity are computed as well. Their method proves effective in segmentation and classification when paired with a Support Vector Machine (SVM) classifier [30]. They do encounter problems with overlap regions which contain both crop and weed pixels and assign this to a third class of mixed vegetation. A solution they suggest is the addition of a pixel-based classifier within their pipeline as Lottes et al. used in [31]. Datasets used in the above research consisted of the same sugar beet field data from the BoniRob set and other similar plant in soil imagery.

In our project, a similar approach is taken concerning the area attribute of our region proposal method. Furthermore, leaf orientation is viewed as a highly informative attribute that indicates where the plant center tends to be. This idea was largely developed in our purely pattern recognition method based on the binary region associated with the leaf. Later in the text, a distinction will be made between leaf orientation and region orientation.

However, a drawback of depending too heavily on attribute morphology in the binary image is that false but well-formed region proposals can mimic weed leaf region proposals. The variability in our data encountered under natural lighting conditions such as excessive glare created many such instances. In fact, these are noticeably absent from the agricultural weeding datasets even when taken under natural lighting as the soil offers little to no reflectivity. In turn, this led to the necessity of incorporating machine-learning elements into our pipeline. The next subsection will explore literature more specifically pertaining to our problem of weed detection in grass.

2.3 Weed Sensing in Grass

As has been explained earlier, the gathering of data under artificial conditions often leads to different challenges than that done under natural conditions. In [32], a comparative study of several proposed methods was carried out on a standardized dataset of grassland weeds (i.e. Rumex and Urtica). These methods included but were not limited to linear binary patterns (LBP), nearest neighbour, and support vector machines (SVM) classifiers, and it was found that the most accurate method consisted of LBP with SVM classifiers.

The main issue that this paper attempted to resolve was the custom nature of each dataset originally tested on. Most sets were taken under constant or artificial natural lighting conditions [32], subdivision between training and test sets were unclear in some, and not all datasets were available. The accuracies reported were all found to be lower than those tested on the standardized set.

Unlike our data and the original data, the dataset collected in [32] is viewed at a downward angled perspective and not from the birds-eye-view (top view). This created different challenges for the authors in terms of perspective operations such as normalization and contributed to the lower scores. The general aim was to facilitate spraying type treatment of the weeds. However, for

our goal of mechanical weeding needs, it was judged that the top view would facilitate control aspects for centering a robot's end effectors on weed centers especially when aiming to reduce the degrees-of-freedom (DOF) and, consequently, project complexity.

Another key difference is that the spraying treatment does not necessarily entail precise location knowledge of the plant root. The target spraying operation is generally applied indiscriminately to weed leaves and root alike, which is not applicable in the case of target mechanical weeding where knowledge of the plant root's location is necessary. **Fig. 5** displays two example pictures from [33] of Urtica, one of the targeted grassland weeds. It can be quickly observed that treatments dealing with Urtica weeds are far different in application from those treating dandelion weeds.

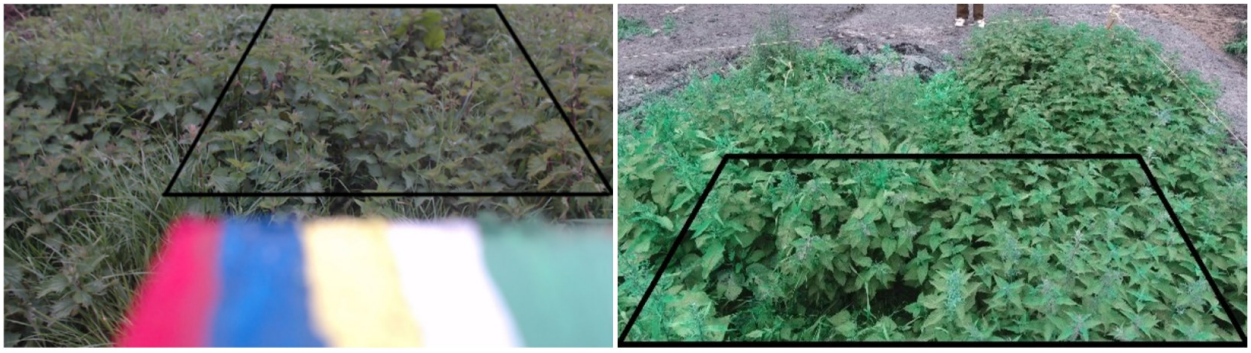


Fig. 5. Examples of the perspective view used in [33]

Another distinction in our goal is, in part, to successfully identify and locate dandelion weed centers using a CNN architecture for both classification and regression. As opposed to generating our features manually, CNN features are generated by the network itself during training. This can save preprocessing time when applied in the real-time.

An important detail made apparent in [32] was that the further the viewing frame was from the ground, the less accurate the classifiers were. This issue was in fact encountered during our initial data collection and the camera height was adjusted to be lower. The goal was to increase weed resolution while reducing the overall amount of weeds per image.

Further research involved in distinguishing the Rumex (dock weed) from grass was carried out by Smith et al. [34]. 3D images were captured with, in one case, an RGB-D camera and, in the other, distinct two-light and four-light photometric stereo (PS) setups. The former method generated point-cloud data, which can be defined as maps of depth values per individual pixels or per regions of pixels depending on the resolution. The latter two methods were capable of capturing high-resolution surface normal data that translates into 3D surface texture information. It was argued by the authors that PS could even work in direct sunlight when used in the NIR.

The PS+NIR methods appear very promising, though the authors only demonstrate their performance for leaf texture recovery in a controlled environment (i.e. it is not done within grass.) The four-light PS is used statically while the two-light method is applied dynamically on moving plant gradient maps where the stems have been previously labelled. It is not indicated whether the gradient maps are taken of dock weeds within grass or under the same artificial conditions.

The RGB-D camera is readily available off the shelf. However, its performance deteriorates when changing viewing distance and its hardware (i.e. USB camera interface) limits higher resolution image capture. For these methods, it is posited that the data gathered in the 3D can enable combinations of the depth threshold with the existing 2D image data analysis. In conclusion, Smith et al. write that their work has the potential to locate plant meristem (i.e. tissue that can develop into all other plant tissues.)

In fact, this is highly encouraging for our research team to improve the 2D image data analysis for the specific task of locating dandelion weeds under a variety of natural lighting conditions. Combining our methods with 3D image data would render them more accurate and precise. However, the scope of the methods presented in this thesis remain in the 2D. Further speculation

on the potential of incorporating 3D information will be discussed in the conclusion and future works section.

2.4 Summary

This section has explored literature relevant to our project from the more general to the more specific. Firstly, a brief survey of current object detection algorithms and the datasets on which they were tested is made. Secondly, recent innovative research on vision techniques developed for agricultural weeding is explained while highlighting both similar and distinct attributes to our own project. Finally, literature on weed detection in grass is critically regarded with respect to the project requirements of this thesis and certain points of divergence are identified.

In summary, the main distinction between current research and the contents of this thesis is the specificity of the problem at hand. Undeniably, there are increasingly shared components as we navigate the literature review from beginning to end. However, there are key points where our presented work branches itself off from current literature and makes its own, perhaps small contribution to the academic community.

Chapter 3: Pattern Recognition Methods and Testing

3.1 Pattern Recognition Methods

Two main schemes are explored in the development of the dandelion weed sensor. The first scheme, developed during the beginning stage of the master degree, is designed using a computer vision and patter recognition approach. This effort in itself is more heuristic, depends on the fine-tuning of parameters, and is not robust to environmental variability.

This approach employs concepts from pattern recognition to accomplish the task of detecting the centers of dandelion weed plant in grass. It is separated into two primary steps; first of which, the weed is distinguished from grass based on its leaves. Techniques of colour scheme conversion and thresholds, frequency filtering, and attribute morphology are used to accomplish this task. The second of the two steps is to identify the plant center. Two methods attempt to do so; the first uses the morphological attribute of region orientation, the second is a simple pooling of neighbouring regions.

3.1.1 Sensing Based on Colour and Shape

Sensing the weed leaf began by discriminating between desirable and undesirable regions within the image based on colour. It was assumed that in an image of weeds, grass, and lawn vegetation,

it would be safe to neglect anything that was not green when looking for viable regions. However, in order to distinguish between colours a brief explanation of relevant colour-spaces in digital images is required.

Generally, each image is subdivided into a grid where each individual cell is called a pixel. The higher the resolution, the more pixels there are in the image, and, therefore, the finer the detail of what can be perceived. Usually, colour images are stored per pixel in a set of Red-Green-Blue (RGB) or Blue-Green-Red (BGR) values. When working with 24-bit images, each value occupies 8-bits ranging between 0 and 255. The ratio of these elements within the pixel is sufficient to indicate the colour in some cases. For example, if an RGB pixel has values [255, 0, 0] then it is clearly red, [0, 255, 0], it is green, and [0, 0, 255], blue. However, when the ratios are closer together, it is difficult to assess the colour. This is partly due to the human perception of luminance being more sensitive than that of chrominance (i.e. colour.) Therefore, it is recommended to change colour-spaces when seeking to threshold for a specific colour.

To solve this problem, a non-linear transform of RGB colours into the Hue-Saturation-Value (HSV) colour-space was made. HSV is a subclass of Hue-Saturation-Lightness (HSL) schemes that are carried out by a deformation of the RGB colour cube. **Fig. 6** represents this transformation. The Travis method [35] was used to calculate HSV from the RGB values and permitted the application of a threshold based on hue as opposed to an RGB combination.

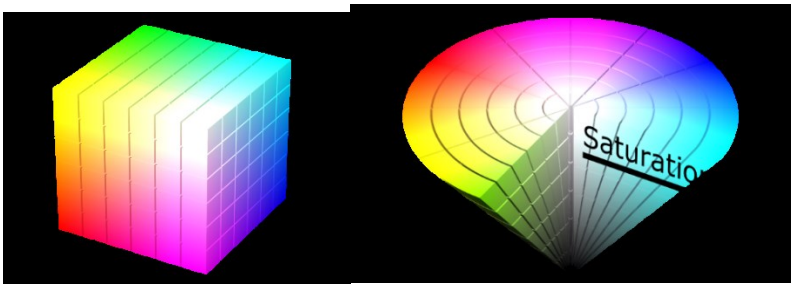


Fig. 6. (Left) RGB colour-space. (Right) HSV colour-space

Below, **Fig. 7** displays an example of the application of a threshold based on the green hue. Hue values between 70 and 170 on a scale of 0 to 360 were generally found to give satisfactory results.

Fig. 7. Image before and after threshold

Subsequently, the image was processed by first converting it to gray scale and then applying a low-pass filter repeatedly until the grass was effectively blurred out. The image was then converted to strictly binary (i.e. black and white) and any remaining regions became of interest for further methods. In **Fig. 8**, we see the sequential application of the greyscale, the low-pass filter, the binary threshold, and the labelled leaves (blue dots) in the original image. Once the regions are identified, two methods are developed to approximate the weed centroid location.

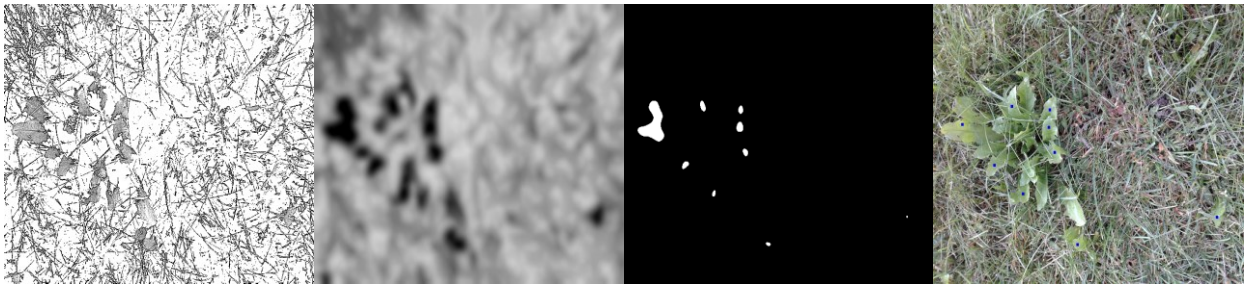


Fig. 8. Image processing steps to detect weed leaves.

3.1.2 Leaf-Region Intersection Method

The leaf-region intersection method employed a form of attribute morphology to propose potential plant centroid locations based on longitudinal intersection points of neighbouring regions. The intersection points where attributed with scores that are based on the number of regions about them. The highest score was proposed as a potential weed centroid. Briefly, the procedure consists of three steps: calculating region elongation vectors, interpolating intersection points between region vectors, and attributing them with scores.

The first step of calculating the region elongation vectors is derived from the following two assumptions: the region shape is similar to the actual leaf shape and the center of the plant can be estimated from the leaf orientation. The original image on the left in **Fig. 9** shows the red lines that approximate the leaves' longitudinal axes while the red circle is where the plant center appears to be. The binary region map on the right shows the longitudinal axes of the regions representing these leaves. The orientations of the lines on the right match rather well with those of the lines on the right.

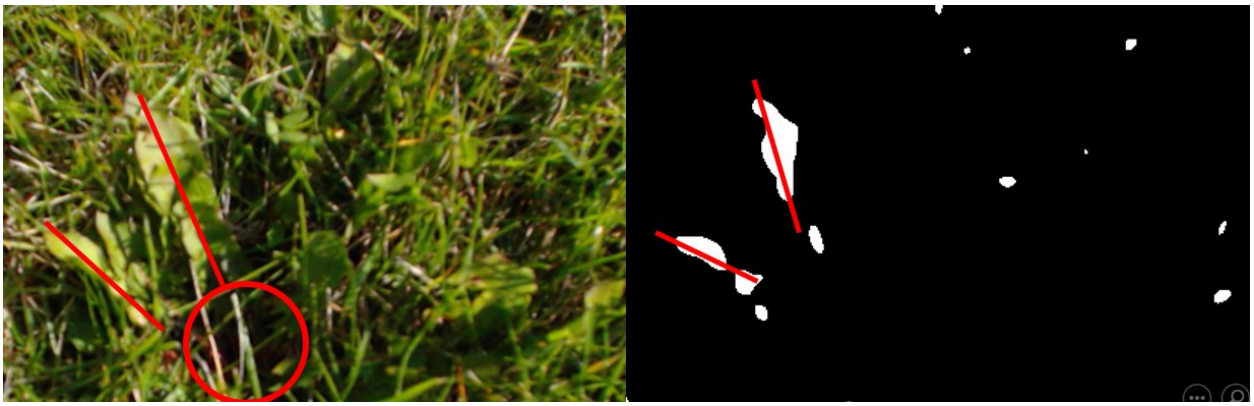


Fig. 9. (Left) Original image. (Right) Binary image.

Since weed leaves can be bent and are not always straight, two directions of elongation are calculated for each region shape from its centroid. We can refer to these directions as vectors with arbitrary non-zero magnitudes. Since each potential leaf region has two vectors, only the vectors

of neighbouring regions that converge are considered for intersections. To decide what a converging pair was, the vectors are linked at their origins (i.e. respective region centroid) by a straight line. This connection and the vectors form three sides of an open trapezoid. If the sum of the interior angles of this open trapezoid is less than 180 degrees, then the parametric lines defining each respective vector will eventually meet and form a triangle.

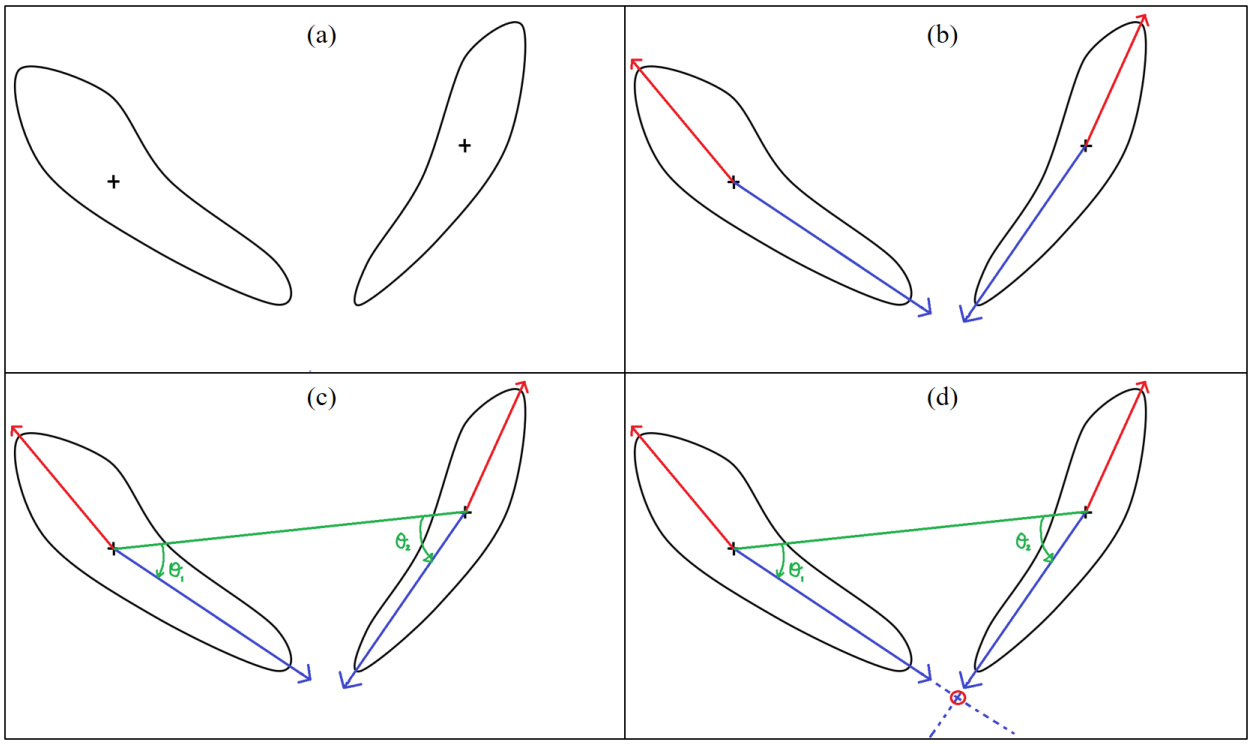


Fig. 10. Determining intersection point based from region orientation

In **Fig. 10**, the process is illustrated. Image (a) is composed of the proposed regions; (b) establishes the vector pairs (red and blue) that indicate the directions of max elongation from the region centroid; (c) shows that the blue vector pair is converging; and (d) displays the intersection being extrapolated from the parametric lines. Once potential intersection points are calculated, many options were possible for the final proposition. They included but were not limited to average pooling with neighbouring intersection points, selective pooling with at least one common intersect, scoring each proposition strength with respect to its surrounding regions, combination of

any of the previous in a weighted scheme, etc. The third option was adopted for simplicity's sake. The following is a brief description of how points were scored.

Each proposal was considered the center of a Cartesian frame. A score of one was given for each quadrant in which a region existed within a certain radius to the origin. A cut-off score was determined based of the highest scores. If there existed a proposal where all 4 quadrants had regions within the radius of the origin, then it was allocated four. Then all other proposals within its neighbourhood would have to be a four as well to be considered for further operations such as pooling. Otherwise, it would be the only accepted proposal. The same goes if the highest score was three or two and so on. **Fig. 11** illustrates two example scores for a system of four total proposals. In image (a), the proposal (dark red) has a score of 3 because its radius is touching binary regions in three quadrants. In (b), the proposal has a score of 4 because its radius is touching regions in four quadrants.

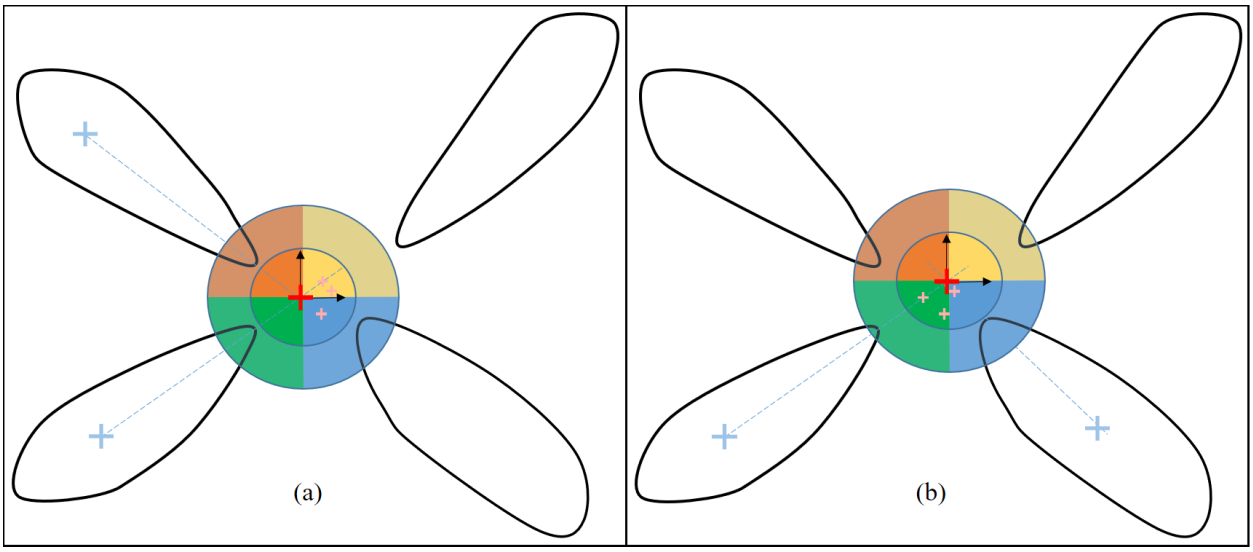


Fig. 11. Proposal scoring method.

The images in **Fig. 12** demonstrate these final two steps of the intersection method on the original image presented in **Fig. 7** and its binary region map. The red dots in **Fig. 12** are the

proposals on the binary map and the blue dot is the highest scoring one. The blue dot is superposed on the original image.

Fig. 12 Proposals from the leaf-region intersection method

3.1.3 Mean Centroid Method

The neighbourhood pooling of region centroids (MCM) was a simple averaging scheme designed as a baseline of sorts for comparison with the intersection method. The coordinates of the centroids of the binary regions were averaged with each other based on their proximity. The amount of clusters of regions determined the amount of neighbourhoods. To simplify the method, once a centroid was assigned to a neighbourhood, it could not be shared with another neighbourhood.

3.1.4 Software

The pattern recognition approach was written in the C++ language using a vision library for image processing provided by Dr. Brandon Gordon in his course MECH 472/6631: Mechatronics and Automation at Concordia University. This library included all the basic functions for image capture, data allocation, centroid calculation, operations such as the low pass filter, etc. The author of this thesis wrote all specific functions using elements of this library as building blocks. They consist of the HSV colour threshold, the complete preprocessing, the region orientation

computation, the intersection point proposals and their criteria, and the neighbourhood pooling method amongst others.

3.2 Testing of Pattern Recognition Methods

This section describes the setup, experiments, results, and analysis for the pattern recognition approach. The experimentation in this section aims to reveal issues that will be addressed in later chapters.

3.2.1 Artificial Weeding Setup

The artificial weeding setup consisted of a rather rudimentary approximation of real life conditions. Three 'weeds' were fashioned out of plastic plant leaves, each with a different shape. They were a two-leaf, three-leaf, and four-leaf weed. They were placed on 4ft by 4ft mat of artificial grass under the indoor lighting conditions of lab EV-S2.355 in the Concordia University building. The camera was placed 40cm in height facing directly downwards. The weed leaves and grass are shown in **Fig. 13**.

Fig. 13. The three "weeds" in grass.

3.2.2 System Testing

The system testing was carried out by recording video combinations of different weed types placed together. They were placed in different image quadrants as shown in **Fig. 14**. The computations for both the leaf-region intersection method (IM) and mean centroid method (MCM) were carried

out for each image frame and their results were superposed directly on the images. This enabled us to reveal system strengths and flaws simultaneously.

In **Fig. 14**, the process is applied on two different examples. In (a), the original images are labelled with the IM in red and with the MCM in blue. In (b), the IM labels are superposed on the binary region map. In (c), the MCM labels are superposed on the binary map.

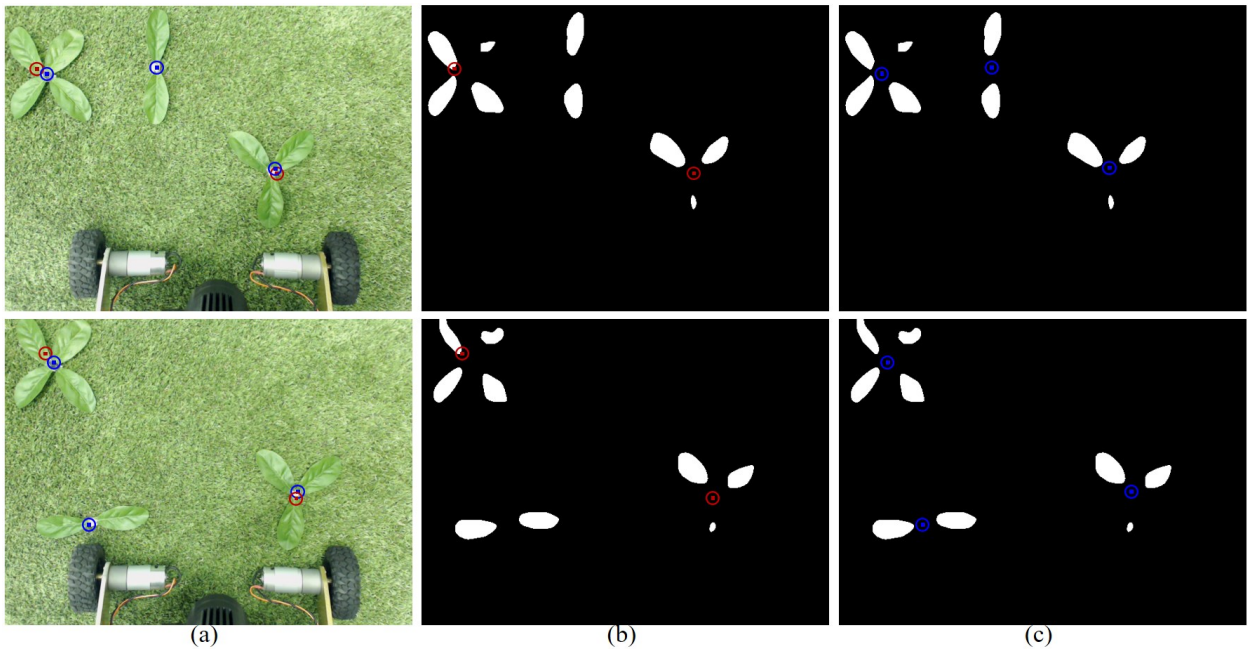


Fig. 14. Two examples of the testing from Table 4

The following tables from [36] summarize the performance for several configurations. They also indicate whether the methods generally agree or not (i.e. *Cmb*. column). A notable detail that can be observed from both the tables is that the IM is not consistently capable of locating the two-leaf weed. Otherwise, the methods generate similar predictions.

Table 2: One Weed per Image in Different Quadrants

Method			Quadrant		
IM	**MCM**	**Cmb.**	**4-leaf**	**3-leaf**	**2-leaf**
YES	YES	YES	1	-	-
YES	YES	YES	2	-	-
YES	YES	YES	3	-	-
YES	YES	YES	4	-	-
YES	YES	YES	-	1	-
YES	YES	YES	-	2	-
YES	YES	YES	-	3	-
YES	YES	YES	-	4	-
NO	YES	NO	-	-	1
NO	YES	NO	-	-	2
NO	YES	NO	-	-	3
NO	YES	NO	-	-	4

Table 3: Two Weeds per Image in Different Quadrants

Method			Quadrant		
IM	**MCM**	**Cmb.**	**4-leaf**	**3-leaf**	**2-leaf**
2/2	2/2	2/2	1	1	-
2/2	2/2	2/2	1	2	-
2/2	2/2	2/2	1	3	-
2/2	2/2	2/2	1	4	-
1/2	2/2	1/2	1	-	1
1/2	2/2	1/2	1	-	2
1/2	2/2	1/2	1	-	3
1/2	2/2	1/2	1	-	4
1/2	2/2	1/2	-	1	1
1/2	2/2	1/2	-	1	2
1/2	2/2	1/2	-	1	3
1/2	2/2	1/2	-	1	4

Table 4: Three Weeds per Image in Different Quadrants

Method			Quadrant		
IM	**MCM**	**Cmb.**	**4-leaf**	**3-leaf**	**2-leaf**
2/3	3/3	2/3	1	1	1
2/3	3/3	2/3	1	2	2
3/3	3/3	3/3	1	3	3
2/3	3/3	2/3	1	4	4
3/3	3/3	3/3	1	1	2
2/3	3/3	2/3	1	2	3
2/3	3/3	2/3	1	3	4
2/3	3/3	2/3	1	4	1
2/3	3/3	2/3	1	1	3
2/3	3/3	2/3	1	2	4
2/3	3/3	2/3	1	3	1
2/3	3/3	2/3	1	4	2
2/3	3/3	2/3	1	1	4
2/3	3/3	2/3	1	2	1
2/3	3/3	2/3	1	3	2
2/3	3/3	2/3	1	4	3

Furthermore, the IM method was tested on real world data. The method was able to perform rather well when the image presented conditions that were similar to those in the artificial testing environment. These conditions include uniform lighting, plants that are not hidden within the grass, and radial plant symmetry about the root. On the other hand, when the field conditions diverged from the controlled testing environment, performance rapidly declined. To correct this, a complete recalibration of parameters was necessary each time a different situation was encountered.

Fig. 15 (a) demonstrates a relatively successful example where one weed center is accurately located by the IM while two other plants are roughly detected. A fourth erroneous detection is made at the bottom left corner. **Fig. 15** (b) gives an example where the excess light leads incorrect predictions. In this case, the system suffered from its lack of robustness; recalibration of parameters would have been necessary.

(a) (b)

Fig. 15. Two examples of the intersection method being tested on real weed pictures

3.2.3 Analysis of Pattern Recognition Pipeline Testing

Generally, the testing revealed that in favourable conditions both methods worked reasonably well together in locating weed plant centroids. However, the IM could not consistently identify the two-leaf weed center due to the orientation of its binary regions. Furthermore, the MCM worked best when the leaf-regions were symmetrical about the plant center; the three-leaf weed often shifted the MCM prediction towards the two leaves that were closer together.

It can be said that the underlying assumptions for both methods were naïve; any region successfully filtered through the colour isolation and thresholding operation was considered a dandelion weed leaf. Furthermore, the artificial grass was useful in determining the general filtering process, however, it did not really mimic the more difficult and recurring scenarios of longer grass blades hiding large parts of the plant. Unfortunately, this pipeline alone would not withstand more rigorous testing.

When set upon a few examples of images of real weeds gathered under natural lighting, the performance deteriorated rapidly with the smallest divergence from the controlled settings. This drawback would require a retuning of parameters for each different case, which is not at all a satisfactory solution. In conclusion, the system was not robust.

3.3 Summary of Pattern Recognition Methods and Testing

The two methods that constitute the pattern recognition approach are described in this chapter. The first is the leaf-region intersection method (IM) and the second is the mean centroid method (MCM). The former uses the orientation of the binary regions associated with the leaves (i.e. leaf-regions) to propose potential plant centers. It then scores these proposals and chooses the highest one. The latter method simply creates neighbourhoods of nearby leaf-region centroids and averages out their coordinates per neighbourhood. It is applied as a baseline of comparison with the IM.

The testing is first carried out in artificial settings. The capabilities of either method are demonstrated; generally, they agree. The IM is especially fragile in cases where there are few leaves in a plant (i.e. the two-leaved weed) and the MCM lacks accuracy when dealing with plant asymmetry. Testing on real world data further exposes system limitations. In fact, there is a general lack of robustness to environment variability where predictions are only accurate on images that resemble the artificial test images.

In conclusion, the performance of this method illustrates the need for a more robust method that is adaptable to the variability of real-world data. In a sense, this pattern recognition approach is a precursor to its machine-learning counterpart.

Chapter 4: Machine-learning Methods

The development of the dandelion weed sensor led to the machine-learning approach, which was designed to address and overcome the shortcomings found in methods from the previous chapter. This section will address the algorithms and theory behind this approach.

From the prior chapter, it is clear by their inherent structure that the pattern recognition methods require much fine-tuning and calibration. Region proposals that survived the preprocessing were all assumed leaves without discrimination and region orientation was treated as synonymous with leaf orientation. The methods presented in this section seek to alleviate this overdependence on heuristic parameter tuning and naïve assumptions that can lead to fragile performance during real-world testing.

For this section, the use of "weed" or "plant" and "weed leaf" or simply "leaf" will all indicate "dandelion weed" and "dandelion weed leaf" respectively unless otherwise specified.

4.1 Classification and Regression

The system in this section makes use of the dual functionality that the fully connected (FC) layers at the end of a CNN can have. Generally, the output of the convolutional and subsampling layers is flattened (becomes one-dimensional) and passed through a FC layer which is analogous to a multi-layer perceptron (MLP) or feed forward neural network. The fully connected layer can

output a probability for different classes when using a soft-max activation function. This is done for classification tasks. On the other hand, when the output is used without a soft-max activation, the MLP generates continuous values. These values can be used for regression tasks in the same way that a least-squares regression fits a function about a set of points. The use of CNNs for both classification and regression problems will be made clear in the following section.

4.2 Pipeline Scheme

The scheme proposed in this section consists of a pipeline approach where the raw image is first input to a binary (i.e. two-class) CNN classifier to determine whether there are any dandelion weeds or not. The two classes are ‘weed’ and ‘not-weed’. The latter category is comprised of grass and other lawn vegetation (i.e. weeds other than dandelions.)

Next, the image is preprocessed and region proposals are identified on a binary (i.e. black-and-white) image. These region proposals are extracted from the original colour image and input into another CNN designed to determine whether the proposal contains a weed leaf or not. The two classes for this classifier are ‘weed leaf’ and ‘not-weed-leaf’ where the latter class encompasses grass and other types of weeds.

Finally, the identified weed leaves are passed through a third CNN that outputs Cartesian coordinates indicating where the plant center is tending to based on leaf orientation. Neighbouring regions are then paired together in the original image and another prediction for both regions is made by the same CNN. They are scored based on the agreement of their individual predictions with their paired prediction. Neighbouring predictions can later be averaged together with respect to their scores to give a final proposal. **Fig. 16** on page 32 demonstrates the complete pipeline architecture graphically.

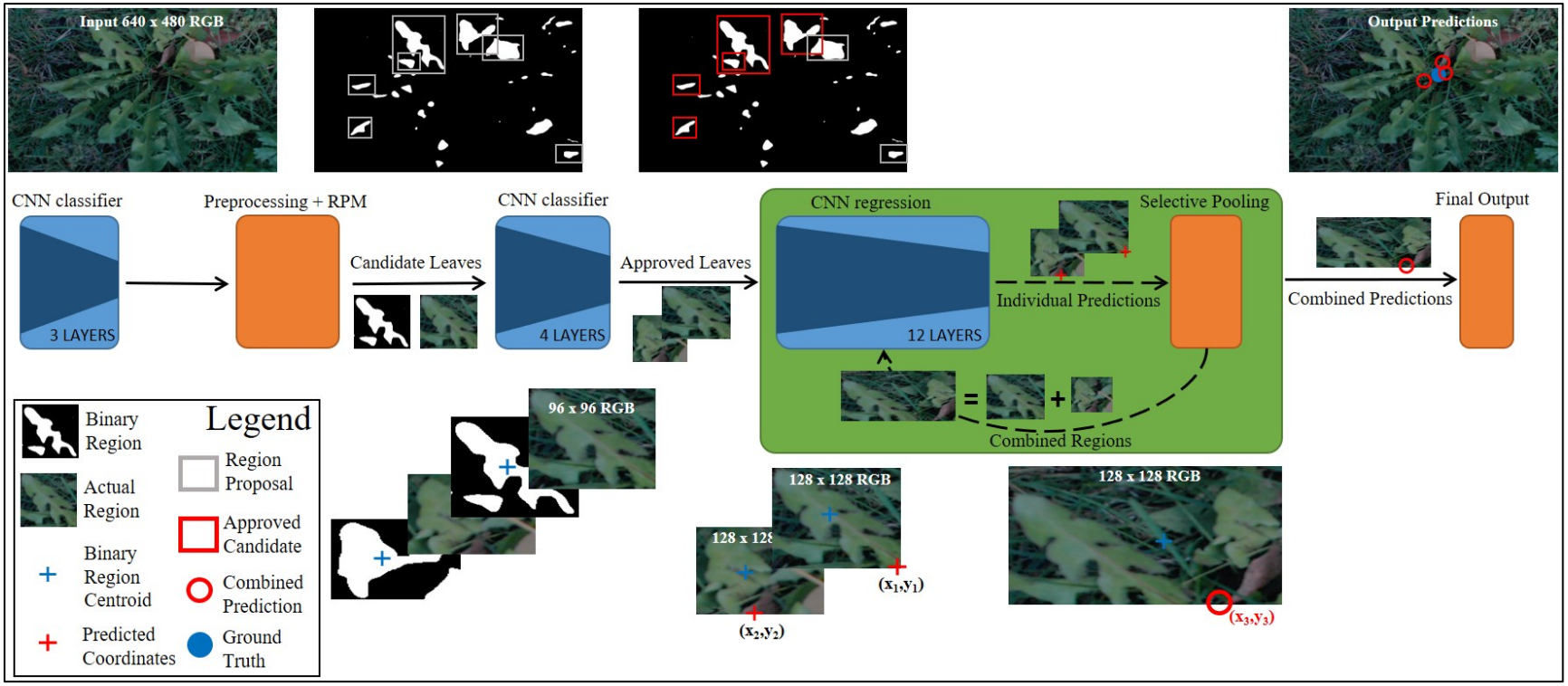


Fig. 16. The pipeline scheme.

As shown, there are two CNN classifiers between which are the preprocessing and region proposal algorithms. Classified weed leaf regions pass through a third CNN that outputs a regression to predict the Cartesian point towards which the centroid tends. Finally, leaf regions with converging prediction vectors are grouped together and returned to the same regression for a second prediction. This process occurs for several region groups and the outputs can be further pooled.

4.2.1 Weed Classifier

The weed classifier is a simple 3-layer convolutional neural network that receives inputs of size 320 x 240 x 3 (i.e. half of the original image's size). It uses 3 x 3 convolutional kernels with single strides for the first two layers. A subsampling using the max-pool argument with a 2 x 2 kernel and a stride of two occurs after each convolutional layer to reduce dimensionality. The third layer is a FC layer (i.e. also MLP) that receives the flattened output of the previous layer as input and outputs the classification for either class 1 or class 2 using a logistic regression activation function. This maps the two outputs as probabilities between zero and one. See the standard and simple CNN structure in the small diagram of **Fig. 17**.

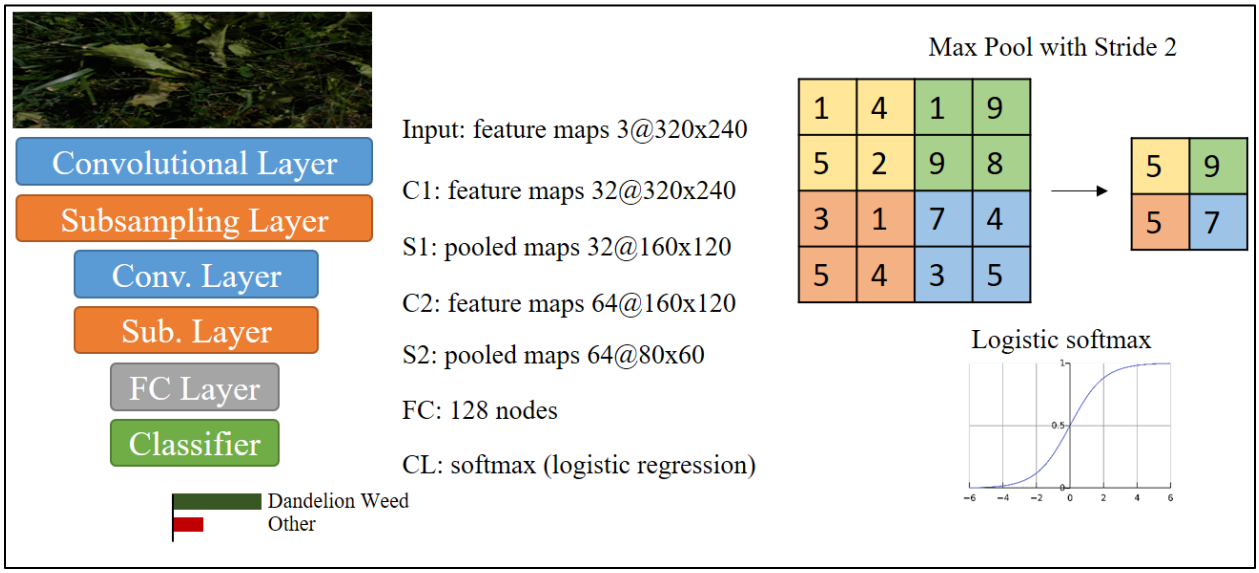


Fig. 17. Architecture of Weed Classifier

4.2.1.1 Image Augmentation for Classification

It was necessary to carry out image augmentation on the dataset to avoid over-fitting. Over-fitting leads to bad model generalization. This means that the model will not perform well on data it has never seen like the test set. Furthermore, it was not possible to load the whole dataset onto the cloud-computing framework that was used to train the data. Therefore, only a subset of the data

was trained, validated, and tested on. It was unknown as to whether the amount of the subset would be sufficient to obtain reasonable accuracies. Hence, data augmentation was used.

Data augmentation consisted of applying a set of random transformations on the input images once per epoch. An epoch represents one full pass of the training data by the CNN and, thus, one complete weight adjustment to minimize the model's loss function. The loss function represents some form of computation of the error (i.e. difference) between the predicted output and the true output. The L2 loss function in (1) is defined as the sum of the squared differences.

$$L2 = \sum_{i=1}^{n} \left(y_{true} - y_{predicted}\right)^2 \tag{1}$$

Data augmentation also serves as a form of regularization. Regularization is required to avoid that certain training examples cause the model weights to grow unevenly in size. For example, if one weight has a magnitude in the 10^6 due to a certain data point while all the others range between 10^2 and 10^3, then it can cause the model to over-fit that specific training example. When input images are augmented by random transformations, the model does not ever see the exact same training set per epoch. This makes it very difficult for a model to over-fit a single data point or, in this case, the whole training set. There are other popular forms of regularization such as dropout regularization where each hidden unit's activity is independently set to zero with probability of usually 50%.

Fig. 18 are a few examples of random transforms applied to the training data. They included combinations of affine transformations such as rotation, shearing, translation, and scaling. Data augmentation was applied similarly to all the architectures in the pipeline though it was applied differently for the regression CNN. The top row consists of the original images and the bottom row consists of the augmented images.

Fig. 18. Image augmentation transformations

4.2.2 Preprocessing and Region Proposal Method

The preprocessing consisted of overall steps similar to those used in section 3.1.1. Colour isolation was not used and histogram equalization was applied. A Gaussian filter (2) replaced the standard low-pass filter for removing grass, and the image binarization (i.e. to black and white) was tuned for a range of parameters. The reason for the first change was that during data collection, many of the weed leaves took on shades of red and pink, as it was late in the fall. To avoid disregarding these leaves, the images were not thresholded for green. Contrast Limited Adaptive histogram Equalization (CLAHE) was used to improve image contrast. In the case of grass removal, applying a Gaussian filter the same amount of times as the low-pass filter preserved small contour detail better when thresholding. Lastly, the variability of properties such as lighting in the dataset led to a fine-tuning of thresholds to accommodate these ranges.

$$g(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

An image histogram graphically represents the frequency of pixel values in an image. The majority of values of a bright image will be on the right side. Darker images will have more values on the left. Roughly, equalization involves a transformation of the image that more evenly distributes the pixel values over the range of possible values. This improves image contrast. CLAHE is the application of this equalization on individual subdivisions of the image to achieve a more robust effect (i.e. to account for images with regions of starkly varying contrasts). **Fig. 19** is an example of a histogram before and after equalization. In (a) the pixel values (horizontal axis) that are low are frequent (vertical axis.) That is why the area of the graph is principally to the left. In (b) the pixels equalized over range of possible values.

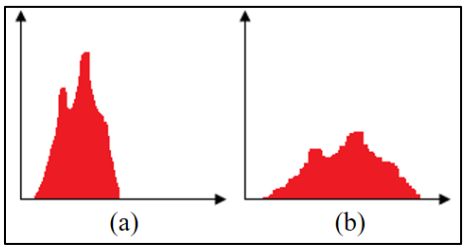


Fig. 19. Histogram before and after equalization

Fig. 20 demonstrates its effect on two example dandelion images from our dataset. The benefits on the region proposal method are significant when observing the binary images in the bottom row of the figure.

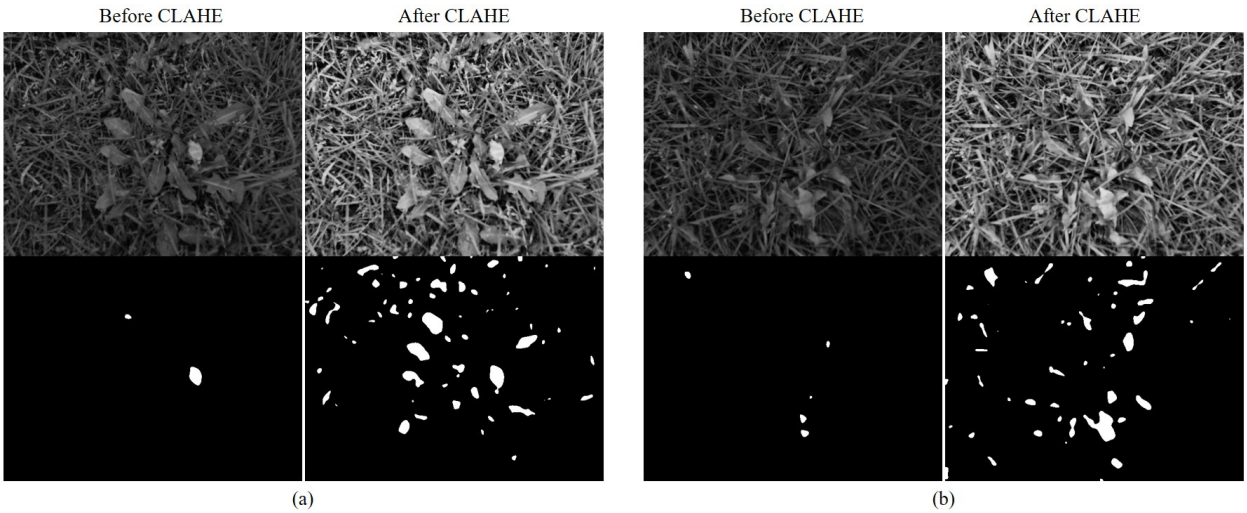


Fig. 20. Using CLAHE

4.2.3 Weed Leaf Classifier

Two classifiers were designed for the leaf classification task. Initially, a four-layer classifier similar in structure to the weed plant classifier in the previous section was trained. Afterwards, a deeper 5-layer structure was used in the actual pipeline scheme. Performance testing will be covered in the later sections. The inputs to the latter classifier were resized to 96 x 96 x 3.

4.2.4 Weed Pair Classifier

An attempt at training a four-class classifier to distinguish between pairs of region proposals was made. The first class consisted of two regions belonging to a single leaf; the second, of two regions belonging to two or more leaves of the same plant; the third, of two regions belonging to two or more leaves of different plants; and the final, of two regions belonging to the 'not-weed' (i.e. grass, other weeds) class. For the last class, if a weed leaf region was paired with a 'not-weed' region, the pairing was also considered a 'not-weed'. The goal was to replace the classifier in the previous section that identified single region proposals by a classifier that was capable of receiving pairs of regions and discriminating between them. In order to do so, the input to this classifier consisted of the RGB image and the binary region map. Figures **Fig. 21**- **Fig. 24** illustrate each of the four classes.

Furthermore, **Fig. 25** and **Fig. 26** demonstrates a few of the different architectures attempted for this particular classifier. In one case, the 3-channel RGB input of size 96 x 96 is concatenated immediately with the 1-channel binary input of the same height and width of the region map. This forms a 4-channel input. In another case, they are each independently passed through parallel convolutions until they are concatenated before being flattened and passed through the classifying MLP. In the third case, the original image, the binary region map, and the product of the original image and the binary region map are input in parallel until they are concatenated before being flattened. The product of the region map (with a slight gradient) and the original image can be seen in **Fig. 27**. The goal of adding the product was to introduce a means of 'attention' to the CNN. In other words, the CNN focused only on the regions whose pixels were turned on by the region map, which functioned as a mask.

The independent pathway scheme in the latter two architectures was an attempt at giving the 1-channel binary input equal or more weight than the 3-channel RGB input of the original image. In conclusion, these architectures were designed to help the classifier discriminate between region relationships.

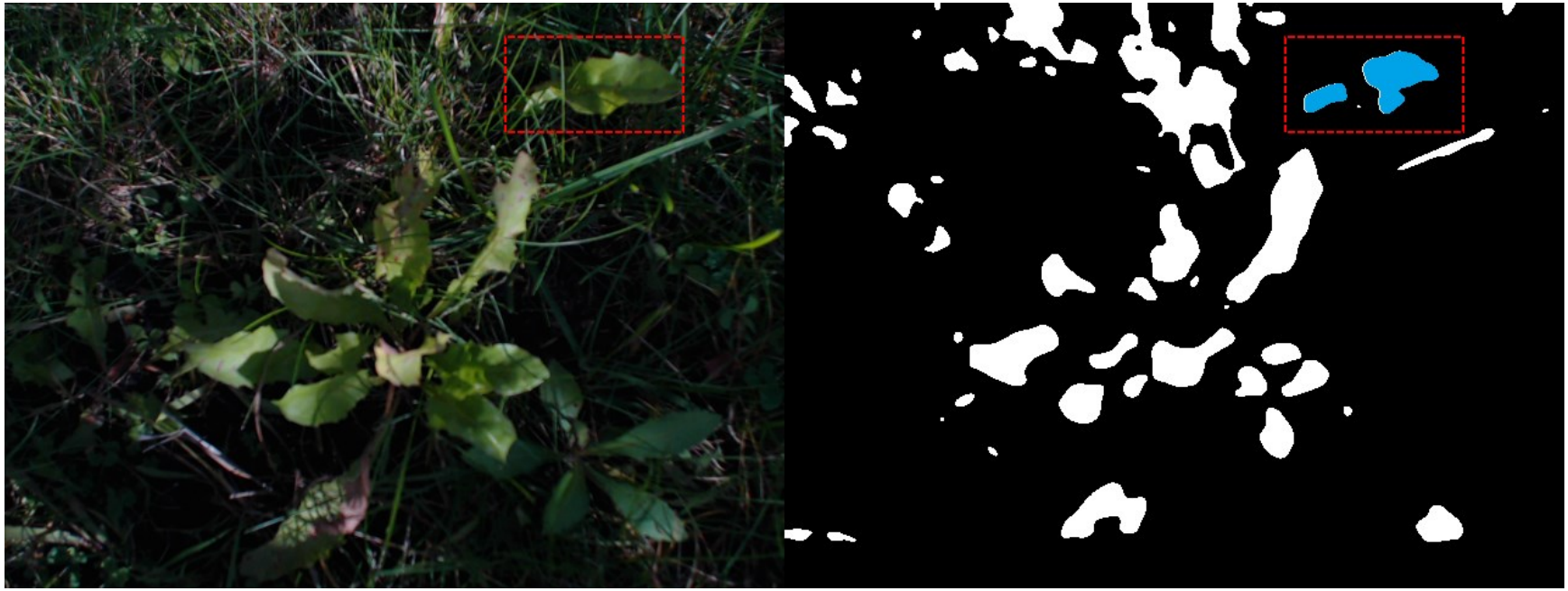


Fig. 21. Example of Class 1

In this example, the original image and the region map are presented side by side. The pair of regions proposed is inside the dashed red rectangle. As can be observed, the pair of regions in the binary map, highlighted in blue, belong to the same leaf on the actual image. Therefore, they belong to class 1.

Fig. 22. Example of class 2

In this case, the paired binary regions, highlighted in red, belong to two distinct weed leaves of the same plant. Therefore, this pair of proposed regions belongs to class 2.

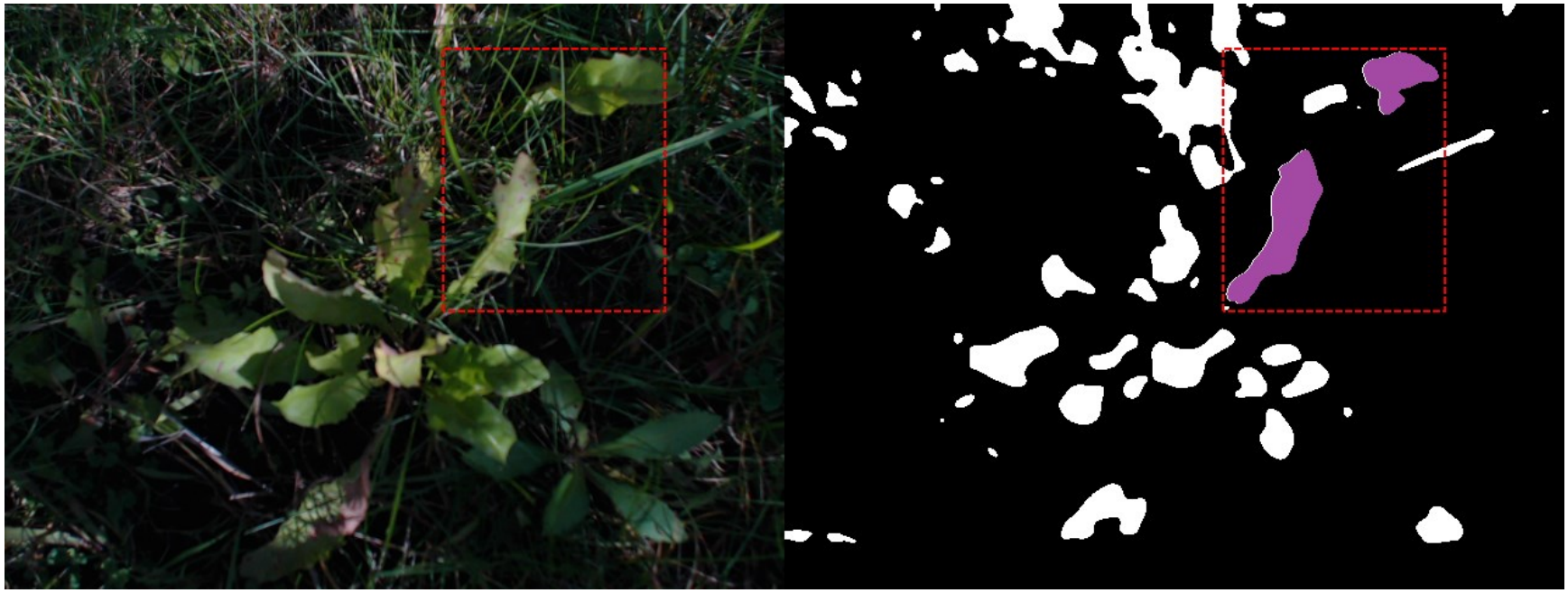


Fig. 23. Example of class 3

In this case, the paired binary regions, highlighted in purple, belong to two distinct weed leaves of the two different plants. Therefore, this pair of proposed regions belongs to class 3.

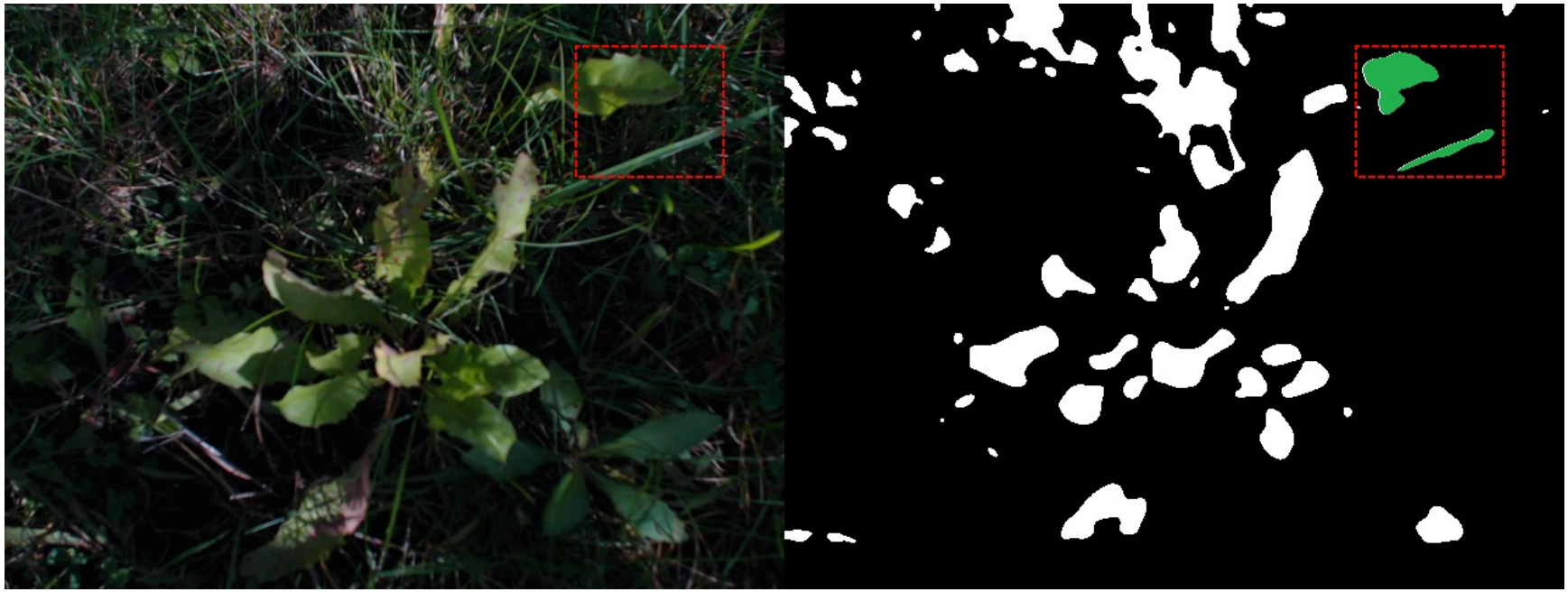


Fig. 24. Example of class 4

In this example, the paired binary regions, highlighted in green, consist of a dandelion weed leaf and a thick grass blade. Since the latter region belongs to the ‘not-weed’ class, this pairing belongs to class 4. The same label would apply if both regions in the pair were from the ‘not-weed’ class.

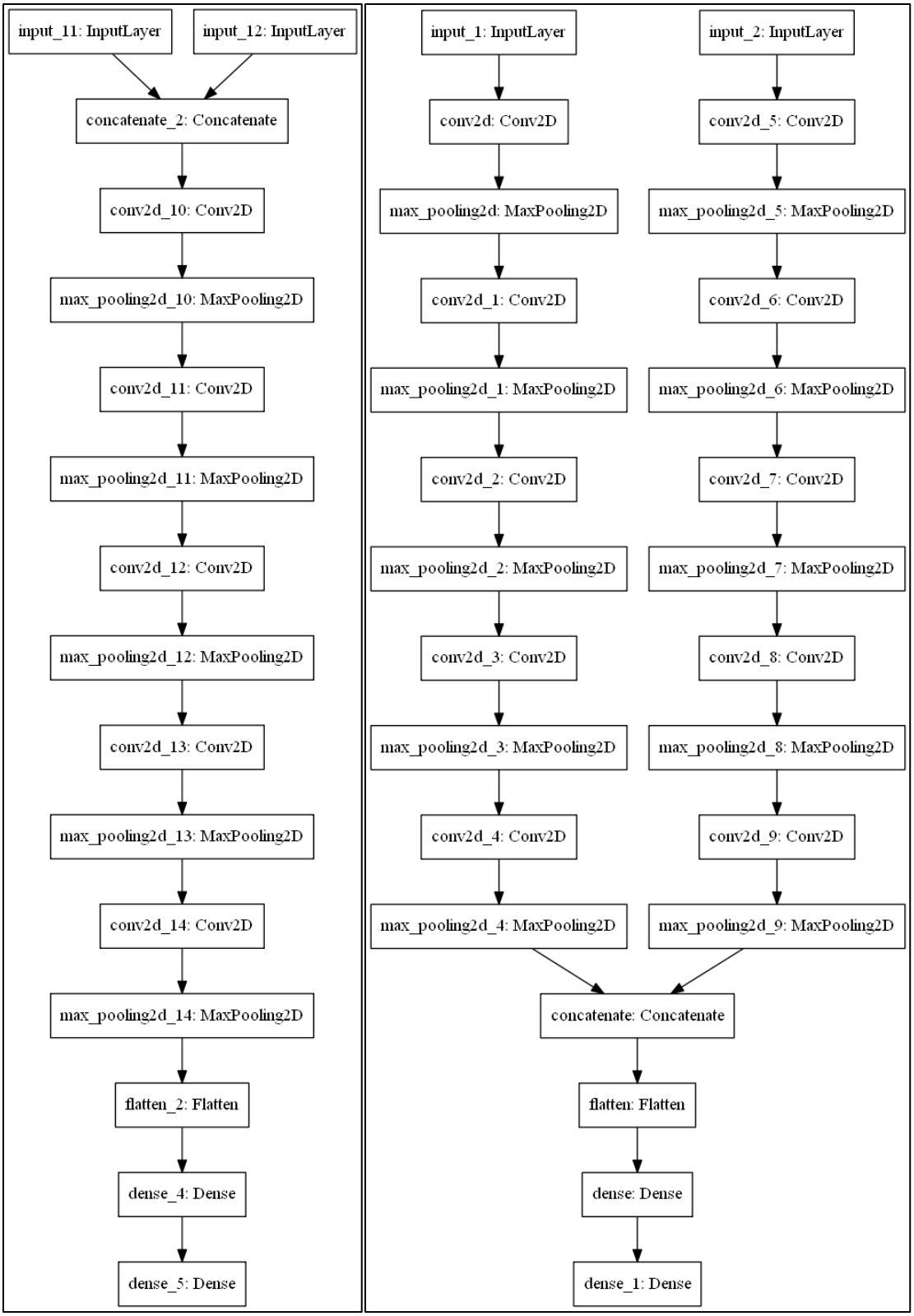


Fig. 25. Single and dual-path convolutional architectures

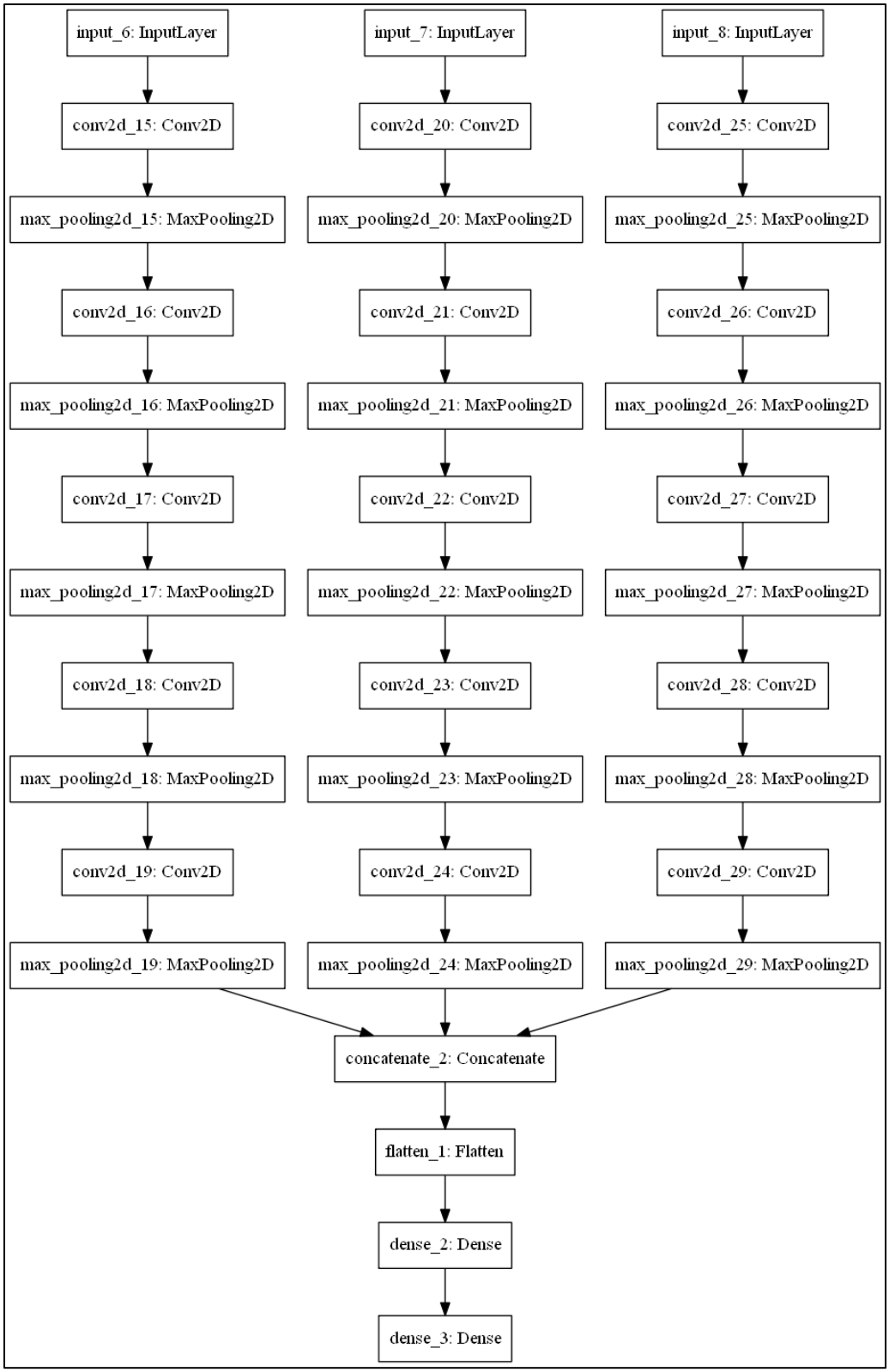


Fig. 26. Three path convolutional architecture

(a) (b)

Fig. 27. Introducing “attention” to the input

The original image and region map can be seen in (a). The product of original image and region map with small gradient on region contours can be seen in (b). The product introduces “attention” by selectively turning image pixels ‘on’ and ‘off’.

4.2.5 Weed Leaf Regression

A CNN was trained to output Cartesian coordinates that indicated where proposed leaves originated in the image. This was based on the leaf orientation. See **Fig. 28** (b) below, which uses red lines to indicate the orientation of a leaf wedge. The assumption was similar to that used for the region orientation in the region intersection method in section 3.1.2. However, the previous method naïvely equated region orientation with leaf orientation. This is not always true as it depends largely on the region map's accurate portrayal of leaf contours. Lighting conditions can often adversely affect the binarization process and lead to region maps that identify the leaf-like objects' locations but do faithfully preserve leaf shape. Furthermore, even when the black-and-white region accurately portrays the leaf shape, sometimes the inclination of that particular leaf (e.g. it is rather upright and not laid-out along its length) means the region may be long along the actual leaf's width, etc. For this reason, a method of determining actual leaf orientation as opposed to binary leaf-region orientation is developed in this subsection.

Fig. 28. (a) Weed-leaf wedge. (b) Wedge direction

A 12-layer CNN, similar in structure to the first CNN described in section 4.2.1, served to predict these Cartesian coordinates. The main difference, other than the overall depth, is that the fully connected layer that classified the features extracted by the preceding convolutional layers

was itself several layers deep. It was composed of seven convolutional layers with pooling and five FC layers. The data was labelled so that the weed point was usually situated at the extreme edge of image. This posed challenges for the image augmentation that will be addressed in below the section.

4.2.5.1 Image Augmentation for Regression

The challenges created by imposing the condition of maintaining the weed point at the extreme edge of the augmented image were encountered during the affine operations of rotation and shearing. Two separate protocols were established for each case in order to control augmented data and keep it consistent with the original data and their labelled coordinates. For sake of brevity, only rotation adjustment will be explained. Shearing adjustment followed similar steps.

Rotation generally presented three cases: the labelled coordinates were initially near an edge and they were rotated away from the edge into the image; they were initially near an edge and they were rotated away from the edge, outside of the image; and, finally, they were initially within the image (i.e. not near an edge) and were rotated anywhere. There is overlap in all these cases; however, this delineation makes for a clear explanation of the adjustment algorithm.

Rotated points were only projected back onto the edge if they were outside the radius of an ellipse within the image dimensions minus a tolerance. This ellipse established whether the rotated points were still “near” the edge or not.

The equations (3) and (4), governing the above relationships, were derived from the radius of an ellipse contained in the input image for the inner bounds and that of the radius intersecting Cartesian horizontal and vertical image boundaries for the outer bounds as shown in equations.

$$r_{inner}(\theta) = \left[\frac{\cos(\theta)^2}{(a-tol)^2} + \frac{\sin(\theta)^2}{(b-tol)^2}\right]^{-1} \quad (3)$$

$$r_{outerX}(\theta, x, y) = \frac{a.sign(x)}{\cos(\theta)}, \qquad r_{outerY}(\theta) = \frac{b.sign(x)}{\sin(\theta)} \quad (4)$$

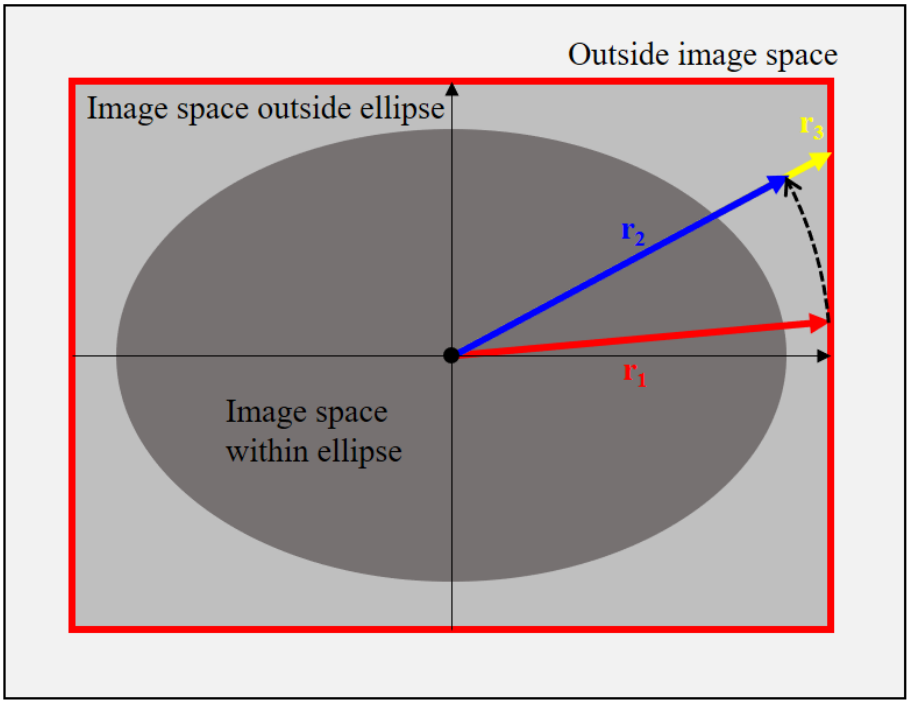
'r' is the polar radius of point, 'θ' is the polar angle of point, and '*x'* and '*y'* are Cartesian horizontal and vertical coordinates of point respectively. '*a'* and '*b'* are half of the image width and height respectively, and '*tol'* is an arbitrary tolerance by which the ellipse is scaled within the image.

If the radius of the rotated point was smaller than the inner bound radius (i.e. that of the ellipse in equation (3)), it was not projected anywhere. If it was larger than the inner bound radius whether within or outside of image space, it was projected to the smaller outer bound radius in equation (4). Naturally, the larger of the two outer bound radii would be situated outside of the image space. Figures in the next page illustrate examples from each case in schematic form.

In **Fig. 29**, the initial vector (red) is on the edge of the image. The rotated vector (blue) is within the image space but outside of the ellipse. The final vector (yellow) has been projected outward onto the image edge. This represents Case 1.

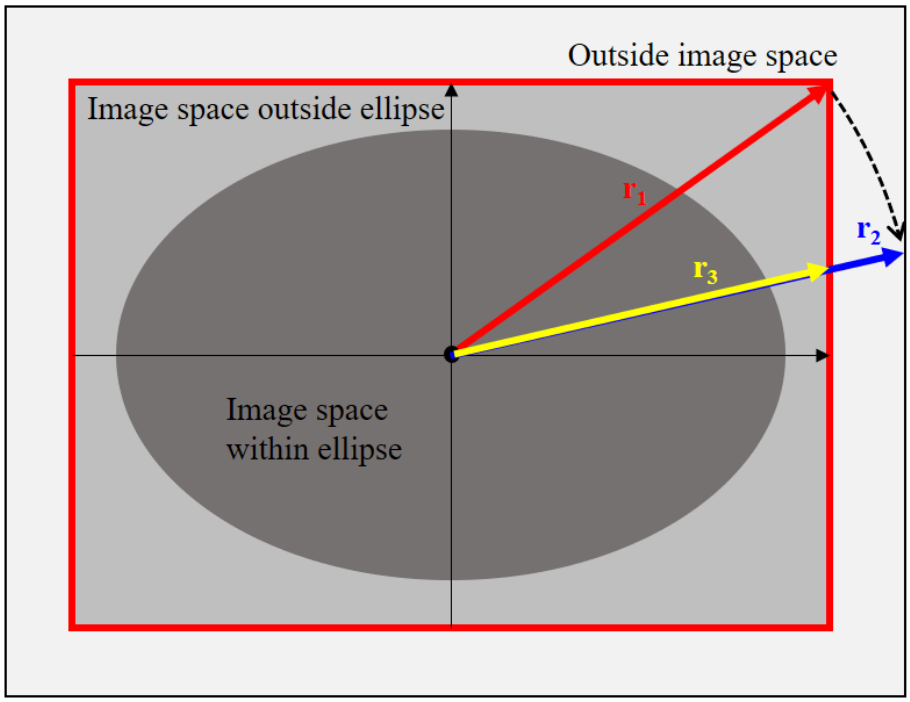
In **Fig. 30**, the initial vector (red) is still on the edge of the image. However, the rotated vector (blue) is now outside the image space. The final vector (yellow) has been projected inward onto the image edge. This represents Case 2.

In **Fig. 31**, the initial vector (red) is within the image. The rotated vector (blue) is within the image space and inside of the ellipse. The final vector (yellow) remains in place because the point is inside the ellipse. This represents Case 3. Were the final vector outside of the ellipse, it would have been projected outwards onto the image edge.



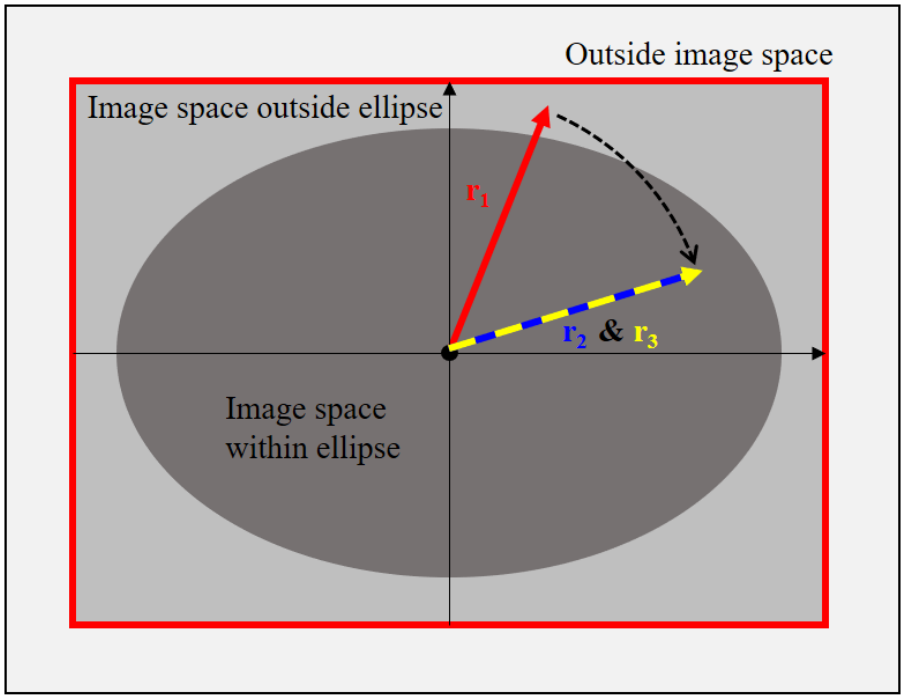
Case 1

Fig. 29. Rotated point projected outwards onto image edge



Case 2

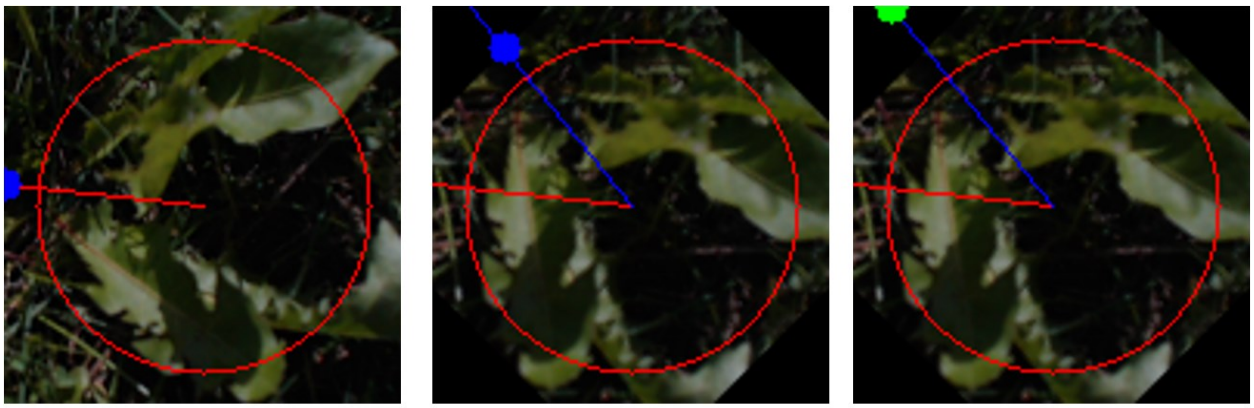
Fig. 30. Rotated point projected inwards onto image edge



Case 3

Fig. 31. Rotate point remains in place.

Fig. 32, **Fig. 33**, and **Fig. 34** illustrate examples on the actual augmented data. In each of the figures, (a) is the original labelled image; (b) is the rotated image and label; and (c) is the label projected back to the edge.

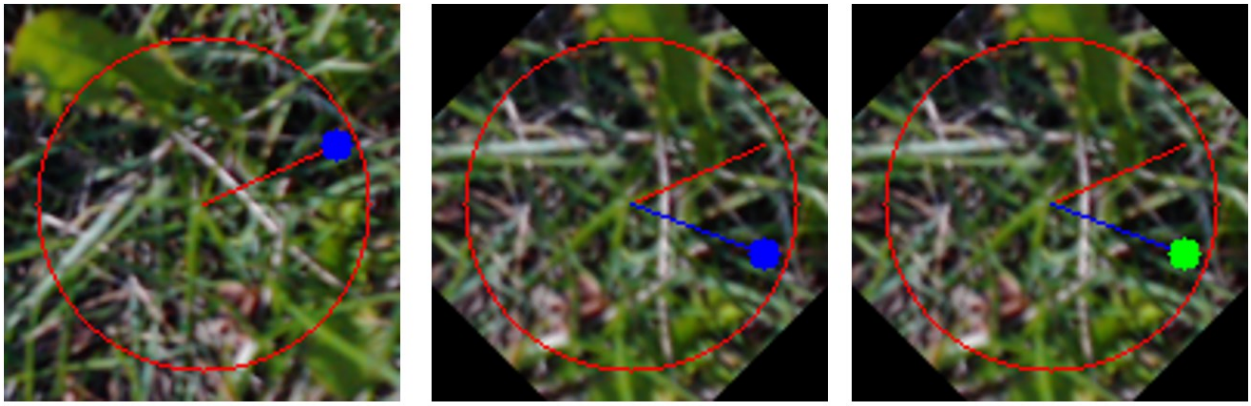


(a) (b) (c)

Fig. 32. Case 1

(a) (b) (c)

Fig. 33. Case 2



(a) (b) (c)

Fig. 34. Case 3

4.2.6 Iterative Prediction Pairing

Individual leaf predictions were paired with each other and sent back to the CNN so that it could make a new (third) prediction on the conjoined pair. For each leaf, the prediction provided by the CNN was taken as the second point of a vector originating at the binary region's centroid. As opposed to the region intersection method, the vector was produced by hybridizing the binary region with the actual leaf as can be seen in **Fig. 35**. Each row represents an individual leaf proposal and its binary region. The hybrid vector (yellow) is generated by using the centroid (blue) from the binary region and the individual prediction (red) from the CNN. For the example on the second row, it is obvious that the binary region elongation does not match the leaf proposal elongation.

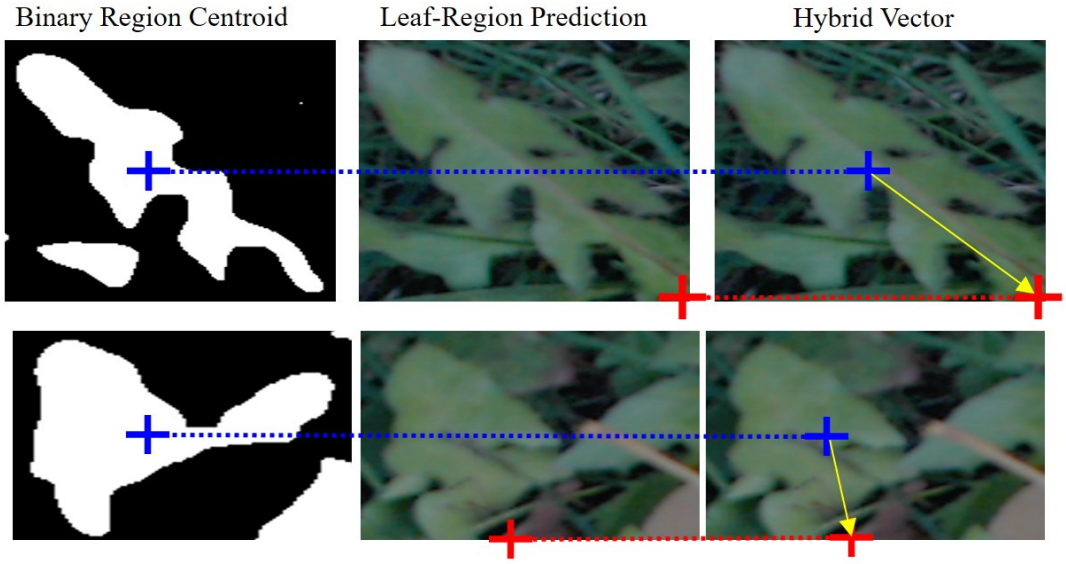


Fig. 35. Hybrid vectors on individual leaves

Once the hybrid vectors were generated and paired with neighbouring vectors, they were predicted upon before being judged for convergence similarly to **Fig. 10** in section 3.1.2. However, another key difference was that there was a total of three independent vectors linked to the pair: a vector from each individual leaf-region and the vector generated by the prediction of the combined leaf proposals originating midway between their binary centroids. Therefore, the convergence criteria was determined in two steps.

Firstly, the convergence of the vectors from each leaf proposal was tested. Secondly, the convergence of each leaf proposal with the combined prediction was tested. If the first test and the second test both converged, then the paired prediction was in full agreement (FA). If the first test did not converge, but there was at least one convergence in the second test, the combined prediction was considered in partial agreement (PA). The figures below illustrate this concept where (a) is the first individual prediction (green), (b) is the second (green), and (c) is the combined prediction (purple).

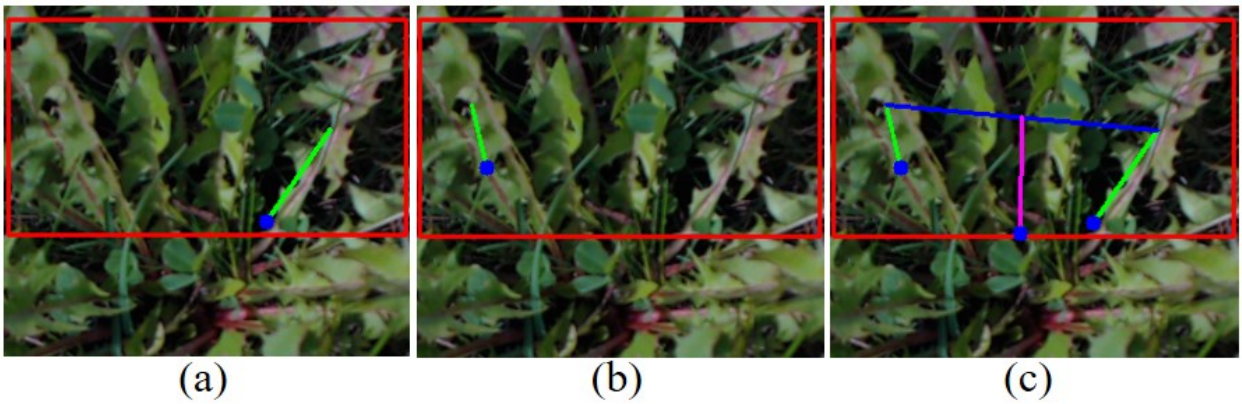


Fig. 36. Combined prediction in FA (purple).

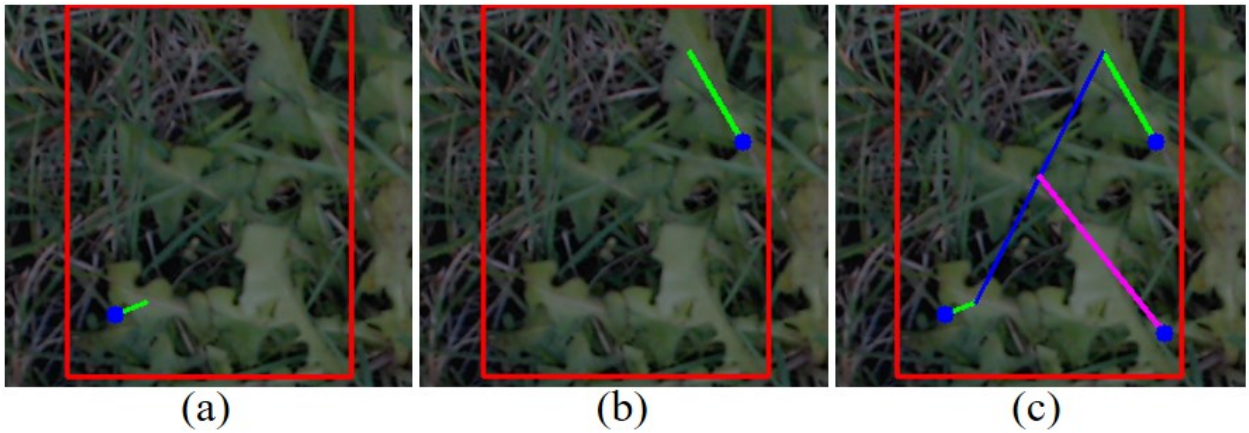


Fig. 37. Combined prediction in PA (purple)

In theory, this method can be iterated by pairing converging leaf-region predictions, then pairing converging combined predictions, and so on, until a prediction is made for the whole plant or many leaves from the same plant. However, this was not attempted in this pipeline.

4.3 Software

Python was used to program the complete machine-learning approach. The "Tensorflow" implementation of the "Keras" API was used for all the CNNs and their training. Image augmentation was done using Image Data Generator class from Keras when training the Weed Classifier and the "imgaug" library was used for both the Weed Leaf Classifier and the Weed Leaf Regression structures. The image augmentation for the latter structure was particularly challenging, as the procedure described in section 4.2.5.1 required much tinkering with the imgaug API.

4.4 Summary

This section offered a rather comprehensive explanation of the methodology used in developing the dandelion weed sensing and recognition. Firstly, a pattern recognition approach was used to interpolate weed centers based on binary region orientation. This method made the naïve assumption that binary region orientation would faithfully preserve the shape of the actual weed leaf in order to make an accurate prediction. A simple baseline of neighbouring binary region centroids was also developed for comparison. Secondly, a pipeline approach firmly rooted in machine learning and the use of convolutional neural networks was developed. A five-step process of weed classification, binary region proposal, weed leaf identification, weed center predictions, and prediction pairing. Furthermore, image augmentation methods specific to our problem were developed. They include, amongst other things, the combination of RGB and binary inputs to 'focus' (see **Fig. 27**) the four-class classifier in section 4.2.4 as well as the edge-placement condition imposed on the regression problem. The CNN structures and their specific pipeline functions were illustrated while the theory behind them was lightly touched upon.

Chapter 5: Machine-learning Testing and Analysis

This section presents an account of the data collection, subdivision, and testing for each sub-element of the machine-learning pipeline as well as the testing of the integrated system as a whole. Finally, a summary of the results and their analysis is given at the end of this chapter.

5.1 Data Collection

The data collection took place in June, September, and October 2019. A total of 8,424 colour images were collected in 640p resolution. There were 4,414 images of the dandelion weed class and 4,010 of the 'not dandelion' class. The images were taken under natural lighting conditions during different periods of the day. Some were taken under clear skies while other were taken during overcast. Of the total set of 8,424 images, around only half was used for the experiments in this work. Dandelion weed leaves generally bore three distinct features; some were broad, angular, and full; some were very narrow; and some were very jagged (or toothed.) Combinations of these traits were also found. They were all considered to be of the dandelion weed class and were not differentiated between. **Fig. 38** presents an example of these distinguishing features.

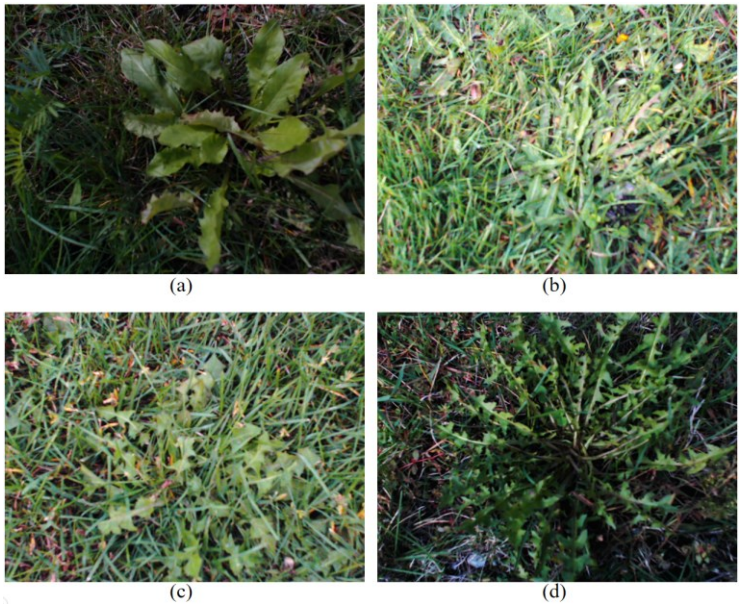


Fig. 38. (a)A broad-leafed dandelion. (b) A dense narrow-leaved dandelion. (c)A sharp-leaved toothed plant. (d) A combination of b and c.

The ‘not-weed’ class consisted of not only grass but also other weeds like clover, broadleaf plantain, and anything else such as fallen tree leaves. **Fig. 39** gives an example of each. Challenges that arose from this blanket classification was that some instances of these weeds were quite difficult to distinguish from dandelion weeds. In fact, broadleaf plantain can seem almost identical to a broadleaved dandelion when immersed in thick grass. Furthermore, due to autumn coloration, some red tree leaves could even resemble the pinkish dandelion weed leaves found around that time.

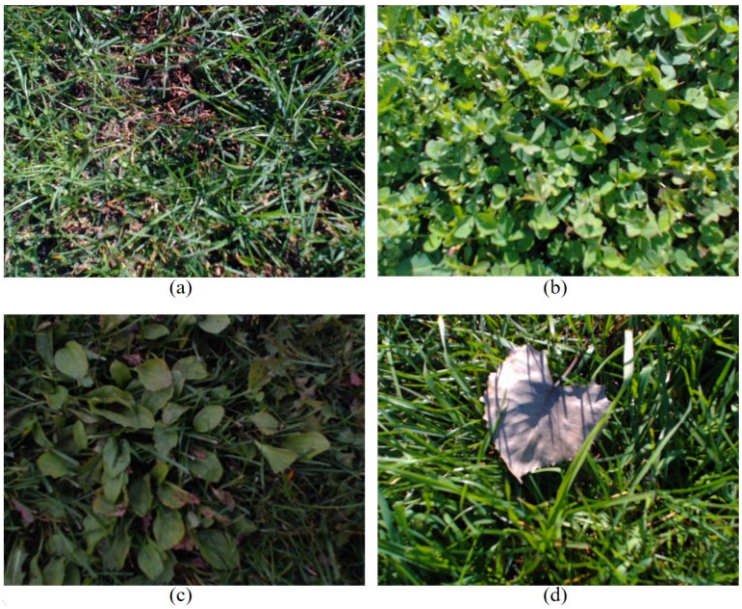


Fig. 39. (a) Grass. (b) Clovers. (c)Broadleaf plantain. (d) Fallen tree leaf.

The images were all taken from top view at a height of 23.5 cm. **Fig. 40** is a similar rig to that used for data collection. The camera used was a "Logitech c920" webcam.

Fig. 40. (Left) Diagram of (Right) Rig used for Data Capture

5.2 Testing of Weed Classifier

The data used for the training, validation, and testing of the weed classifier consisted of 4,440 raw images divided in half between weed and not-weed class. They were further subdivided into a 60, 20, and 20% split for training, validation, and testing respectively. A note on data splits for training

neural networks –or any predictive model for that matter: once per training epoch (i.e. single full pass of the training data), the system learns features and weights from only the training set and then computes loss and accuracy from both the training and validation set. However, only the loss gathered from the training set is used to update weights per epoch. This entails that 'learning' is solely done from the training data. Computations on the validation set are done to compare different models' performances. Naturally, for each of two consecutive epochs, a model will have a different set of weights. Thus, it is considered two distinct models. After the necessary number of epochs is completed, model performance is evaluated on the test set.

Without augmented data, the network tended to over-fit the training data and performed poorly on the test set. Augmenting the data resulted in a 96% accuracy on the test set.

5.3 Testing of Leaf Classifier

The weed-leaf classifier began as a four-layer CNN. It was trained on a subset of the training data used for the previous classifier. The weed class consisted of 3,041 manually cropped images of weed-leaf wedges of at least two leaves of the same plant. These were called leaf-wedges (see **Fig. 28**). The other class consisted of cropped images from the 'not-weed' class that were generated by a code that subdivided images from the original 'not-weed' set. This amounted to 1,925 cropped images to yield a total data subset of 4,966. With image augmentation, the model achieved an accuracy of 96% on the test set.

However, when combined with the region proposal method (RPM), it did not perform well in identifying regions that consisted of weed leaves. For that reason, a second iteration of this classifier was created by adding a convolutional layer and creating a new data subset. This new set consisted of 1,914 and 1,942 images of the weed and not-weed class respectively. It was created by labelling images that were generated by the region proposal method. Consequently, the CNN

was tailored to fit the RPM's output. This new CNN ended up achieving an accuracy of 89% on its test set. The lower accuracy can be attribute in part to the lower resolution data. The reason is that a proposal from the RPM is derived from a single binary (black-and-white) region. Though it is possible for the binary region to represent multiple leaves on the actual image, it often only represents a single leaf. This means that the majority image proposals were small enough to contain single leaves whereas the manually cropped images from the previous classifier consisted of two or more leaves. Therefore, the images for this classifier were naturally smaller and of lower resolution. In fact, the original mean image area of the wedges was 20,000 square pixels whereas the mean area of images proposed by the RPM was only 5,000 square pixels. This is a significant drop in original image resolution.

The reason was due to the following assumption: the leaf-wedge classifier should be able to identify single leaves just as well as it can identify the leaf-wedges on which it was trained. This assumption proved to be false.

Table 5: Accuracy of Leaf Classifiers

No. of Layers	No. of Data Points (i.e. images)	Generation (manual/auto)	Mean Resolution before Resize (pixels-square)	Test Accuracy
4	4,966	manual	20, 000	0.96
5	3,856	auto	5, 000	0.89

5.4 Testing of Leaf-Pair Classifier

The leaf-pair classifier was initially meant to serve as the third CNN classifier after the leaf-classifier. This was meant to compare paired leaf proposals and determine whether they belonged to the same leaf, to different leaves of the same plant, to different leaves of different plants, or to the 'not-weed' class. An in-depth look of the four classes is given in section 4.2.4.

The data collected to train this CNN consisted of leaf-region pairs that exited the previous classifier coupled with their region maps. The subdivision consisted of 1,441; 1, 232; 996; and 878 image/map couples for classes 1 to 4 respectively. The three architectures performed better than random (25%).

However, the single, dual, and tri-branched structures seen in **Fig. 25** and **Fig. 26** plateaued at 66, 67, and 69% accuracy respectively. Though this is significantly better than random, issues regarding the ambiguity of the data with respect to the classes (that it is often difficult to determine when two leaves belong to the same plant) require resolution before applying it. The maximum accuracy is not sufficient for a beneficial integration into the pipeline. An important finding of these results is that there is a capacity for learning the four classes, however, more data points per class and perhaps a higher original image resolution are both required. In addition, Bosilj et al. [24] faced a similar problem regarding leaf overlap of distinct plants and recommended pursuing a pixel based classifier as in [31].

Table 6: Accuracy of Leaf-Pair Classifiers

Structure	**Special Colour Image/ Binary Region Map Operations on Model Input (Yes/No)**		**Test Accuracy**
	Concatenate	**Product**	
Single Convolutional Branch CNN	Yes	No	0.66
Dual Convolutional Branch CNN	No	No	0.67
Tri Convolutional Branch CNN	No	Yes	0.69

5.5 Testing of Leaf Regression

The data subset for the leaf regression was the same wedge class from the initial dataset for the leaf classifier (see section 5.3) –that is, the set that was manually cropped from the original raw images. The labelled images numbered 3,016 and they were once again split into the 60-20-20 train-validate-test ratio.

The 12-layer structure was achieved by incrementally adding convolutional layers and, later, fully connected layers to a base three-layer model. Each trained model's feature extractor (i.e. convolutional layers) was frozen at peak accuracy, the classifying layers were truncated (i.e. FC layers), and further convolutional layers were added. Once seven convolutional layers were reached, the same process was applied to the FC layers.

The labelled coordinates were normalized to range from zero to one using the top left corner of the image as the standard origin for images using the OpenCV library. Two parameters were monitored to compute loss and monitor system performance. They were based on the normalized Cartesian coordinates. The first was the loss computed by the mean Euclidean distance between the labelled coordinates and their predicted counterparts or the Root-mean-square error (RMSE). The RMSE was converted to an accuracy by dividing the distance by the larges possible Euclidean distance between two points in the normalized Cartesian frame (i.e. $\sqrt{2}$) and subtracting from one. The other value was simply the mean absolute error (MAE) of the sum of the differences between the X and Y components of the true and predicted coordinates. **Fig. 41** and equations (5) and (6) illustrate these measurements and their computation.

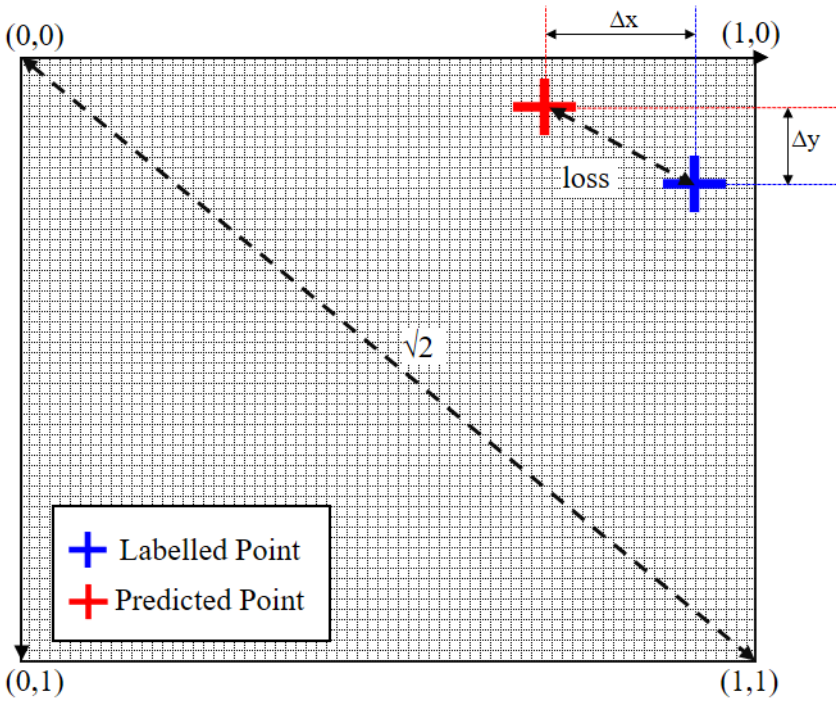


Fig. 41. Loss Measurements for Regression

$$RMSE = \frac{1}{N}\sum_{i=1}^{N}\left(\Delta x_i^2 + \Delta y_i^2\right)^{\frac{1}{2}},\ MAE = \frac{1}{N}\sum_{i=1}^{N}|\Delta x + \Delta y| \tag{5}$$

'N' is the number of images and 'Δx' and 'Δy' are horizontal and vertical component differences between the truth and the prediction as shown in the figure.

$$Overall\ Accuracy = 1 - \frac{RMSE}{\sqrt{2}}, Accuracy\ per\ dimension = 1 - MAE \tag{6}$$

Now, an interesting note on the performance of the leaf-regression CNN is that images ranging in floating point pixel value between 0 an 1 (normalized by dividing the original 24-bit integer image by 255.0) consistently increased in overall accuracy undergoing a Gaussian brightening using equation (7) of 0.8. Brightening any further decreased the accuracy.

$$g(x) = p[i,j]^x \tag{7}$$

Though brightening was explored initially during the testing phase only, it was decided to introduce brightening to the training and validation processes, which resulted in a testing increase of 0.2% in **Fig. 42**. When this behaviour was observed, it was decided to brighten data during training and validation. This resulted in an even greater improvement of 0.8% on the test set. This

may have to do with the property of Gaussian brightening in that it can maintain or enhance image contrast unlike brightening by scalar multiplication of the image pixels.

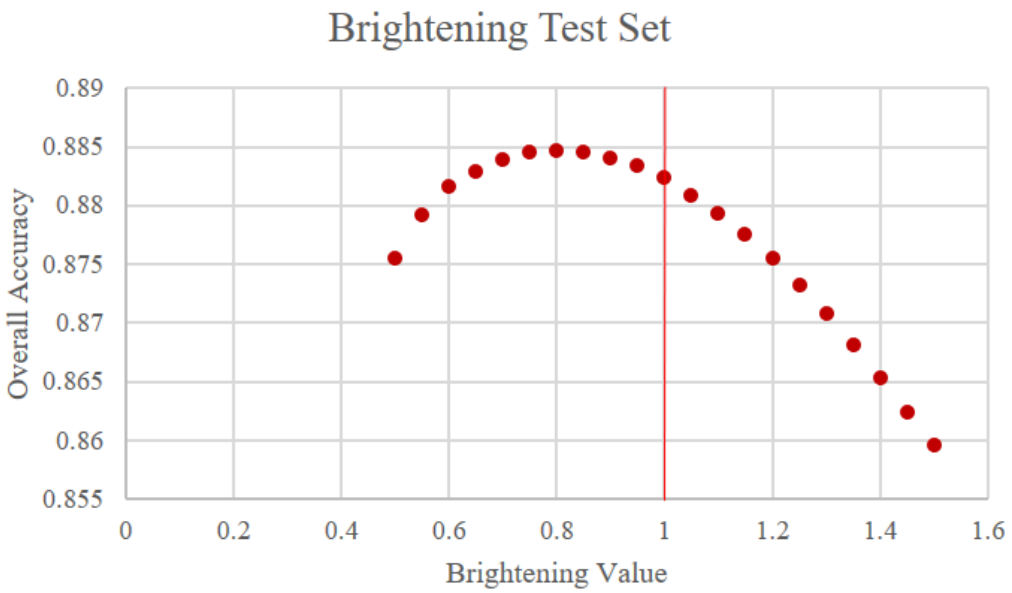


Fig. 42. Brightening vs Overall Accuracy: the red vertical line indicates no brightening,

As with the leaf classifier, the leaf regression CNN was initially trained using manually cropped data. To avoid poor performance on the leaf regions provided by the RPM, it was decided to add newly labelled images of single leaves and not wedges. The 1,320 additional images led performance on the test set to fall from 89% to 86%. Normally, an increase in data tends to improve performance. However, this can be explained by the poorer resolution of the smaller single-leaf images not contributing positively to model generalization. Although manually cropped, they have an average resolution of 11,000 pixels-square, which is around half the 20,000 of the original wedge images.

A summary of the leaf regression results can be seen in Table 7 and example pictures of test predictions versus the ground truth are in **Fig. 43** using the final regression model (green highlight.)

Table 7: Leaf Regression Models and Accuracy

Model	Brightening	Data Used (Yes/No)		Total images	Overall Accuracy on Test Set
		Wedges	Single Leaves		
12-Layer	Yes	Yes	Yes	4336	0.86
12-Layer	No	Yes	No	3,016	0.88
12-Layer	Yes	Yes	No	3,016	0.89
5-Layer	No	Yes	No	3,016	0.79
6-Layer	No	Yes	No	3,016	0.80
7-Layer	No	Yes	No	3,016	0.84
8-Layer	No	Yes	No	3,016	0.86

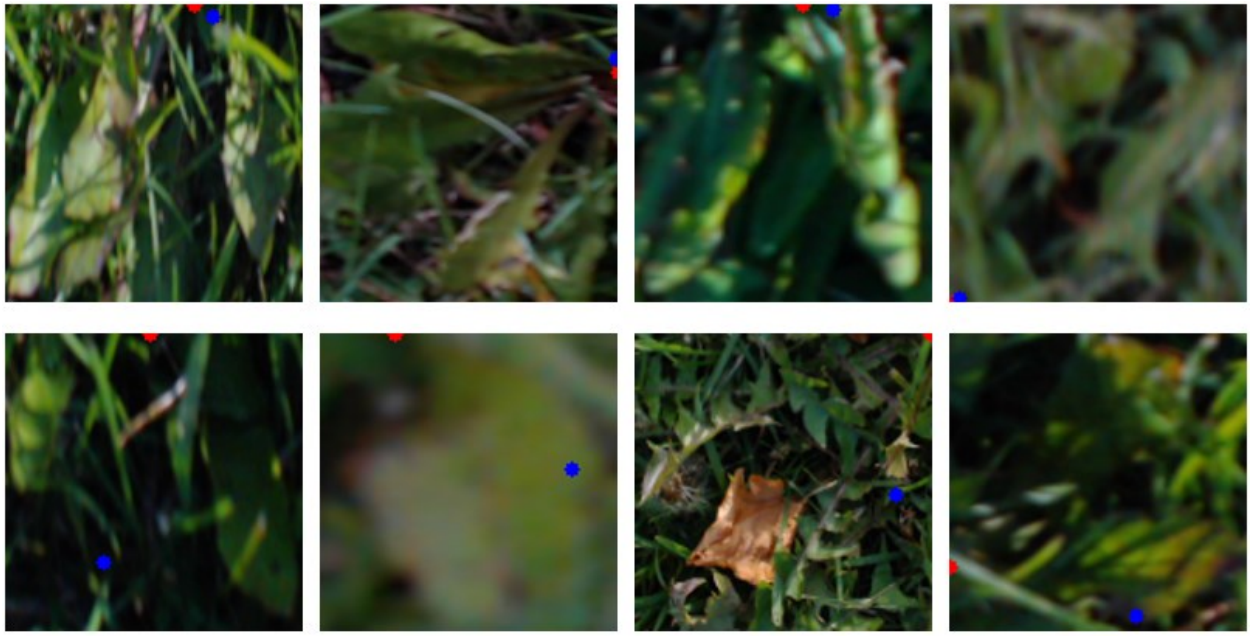


Fig. 43. Final model on test set.

The red dots in the above figure indicate the ground truth and the blue dots are the model output. The top row demonstrates very close agreement while the bottom row shows the opposite case.

5.6 Integrated System Testing

The integrated system that was tested consisted of the following elements: the RPM, the leaf classifier, the leaf point predictor, and the two-iteration pooling (see **Fig. 16**.) The first step in the scheme (i.e. determining whether there was a dandelion or not in the photo) was neglected as its component achieved a high enough accuracy of 96% on the test set. In fact, recall the images it was tested on were half the resolution (320p) of the actual data. This indicates that there was no real need to test that component any further. On the other hand, as previously mentioned, the four-class leaf-pair classifier was also neglected, as its final accuracy (69%) was just too low.

The integrated system was tested on a subset of the data that was never before seen by any of the networks. This consisted of 262 images with at least one labelled weed by an undergraduate research student. Briefly, using the same measurements of Euclidean (RMSE) and Cartesian (MAE) component distance of the pixels, the minimum value was computed between the label coordinates and the predictions. These parameters were normalized as in **Fig. 41**. That is, the RMSE was divided by the largest Euclidean distance possible (i.e. from one corner of the image to the other) while the vertical and horizontal MAE components were divided by the respective image height and width.

Collectively, these computations on the minimums were seen as the simplest means of determining whether any of the weeds were accurately located. Although admittedly an optimistic method, taking the minimum error between prediction and ground truth marginalized the problem of incorrect or unclear labels. Furthermore, it simplified performance evaluation for eventual comparison to other methods and eliminated the need for pooling densely packed clusters. Finally, the results from using the minimum error readings still provided a strong proof for the success of the pipeline and its potential for improvement with higher resolution data.

In fact, the results on the test set alone were very promising. The minimum and normalized prediction for the RMSE was of 9% while that of the MAE for the horizontal and vertical components were each 7%. A few examples of this performance can be seen in the figures **Fig. 44** through **Fig. 51**. They are presented in close to full scale 640p to provide the reader with observation of the image in detail similar to actual system input. Table 8 presents key points posited throughout this paper that are addressed by the figures.

Table 8: Image Guide for Test Results

Project Key Points with respect to the Integrated Testing	**Figures**
The minimum MAE and RMSE are sufficient in demonstrating system capabilities of accurately locating weed centers.	**Fig. 44** and **Fig. 45**
Multiple weeds at a time can be detected with accuracy competitive to or better than the average human operator (i.e. the person who labelled the data)	**Fig. 45**, **Fig. 47**, and **Fig. 50**
Weed centers can be detected even when hidden within the grass, a challenge not significantly addressed by agricultural weeding literature.	**Fig. 45**, **Fig. 47**, and **Fig. 51**
It is possible to join (in an iterative matter) the leaf-region pairs that have generated the predictions in both FA and PA. This is potentially valid for tight and spread out prediction groups within tolerances.	**Fig. 46**, **Fig. 49** and **Fig. 50**
The minimum MAE and RMSE are not sufficient for practical in-field application; a pooling scheme is definitely required for real world implementation.	**Fig. 44**, **Fig. 46**, **Fig. 49** and **Fig. 50**
Though robust to some lighting in natural settings variability, oversaturation and low image resolution can lead to significant loss of detail that deteriorates performance.	**Fig. 48**

A note on the images is that the blue dots represent the ground truth, the hollow red dots are predictions that are in full agreement (FA), and the hollow green dots are in partial agreement (PA) in section 4.2.6, Iterative Prediction Pairing. The figure captions provide remarks that highlight important qualitative issues such as the ambiguity of the truth label, the advantages/disadvantages of using the minimum RMSE and MAE, prediction strengths, prediction weaknesses, etc.

Fig. 44. Test example 1

In the case above taking the minimum values for the RMSE and MAE seem just as good as pooling the points in whatever weighted scheme. The system was able to detect one weed center just as accurately as the human eye. The other weed was not detected at all in FA nor in PA. This may be due to the RPM; a look at the region map is required. Note: these are examples of the weed type in **Fig. 38** (b).

Fig. 45. Test example 2

Above, the system accurately detected a labelled weed center in FA. Once again, using minimum error values is sufficient to prove accuracy. Furthermore, one label is not detected. However, an additional unlabelled weed center is rather accurately detected even though it is hidden in grass.

Fig. 46. Test example 3

Although the predictions are largely concentrated near the weed center in this example, they are rather far apart. This is an example where the use of minimum RMSE ad MAE values instead of pooling prediction clusters shows its practical limitations. It is understood that in real-time and real world settings, no ground truth information is readily available. Therefore, a pooling (or averaging) method would be necessary. In this example, averaging the predictions will vary in accuracy depending on the tightness of the grouping radius. By visual inspection, a reasonably tight grouping radius would provide up to three individual weeds where it is obvious there is only one. Any larger radius would propose one weed center yet the accuracy would be significantly diminished. That being said, even in this example, the minimum error measurements still undeniably demonstrate the effectiveness of the overall pipeline scheme. The relative

improvement from choosing an arbitrary pooling method is beside the point. On a positive note, the leaf-regions associated with each paired prediction can be further paired together to identify the plant as a whole and perhaps iterate another prediction.

Fig. 47. Test example 4

This example illustrates a recurring theme to this report: inherently, the labelled data tends to increase in ambiguity as the weed concentration increases within grass. By close inspection of the image, one can observe at least three distinct dandelion plants –and that is being conservative. Case in point, the leftmost red prediction (FA) matches well with the blue truth label, while the prediction slightly to its right is arguably well centered on another plant. The rightmost red prediction may belong to yet another plant though it is not located on its center. The rightmost green (PA) predictions are very close to their truth label but are clearly not centered.

Fig. 48. Test example 5

The saturation of light in the above image makes it difficult for one to distinguish the plant center. However, the blue truth label can be assumed accurate. The predictions are spread widely across what appears to be a single or possibly two plants. The oversaturation of natural light floods image pixels considerably diminishes detail within the image. Thus, the limitations of the 640p resolution are made apparent. Higher resolution should provide more robustness to environment variability.

Fig. 49. Test example 6

The above example presents a favourable cluster of predictions near the plant center. Compared to the ground truth in blue, a simple average of the predictions that are grouped together (neglecting the offshoot at the bottom) would generate a point that is comparable to the label, if not just as good. The plant in this example is of type (c) from **Fig. 38**.

Fig. 50. Test example 7

A good overall example of system performance in uniform lighting. What appears to be two overlapping weeds, of which one is labelled, are accurately represented by groups of points around their respective centers.

Fig. 51. Test example 8

Finally, the above example of the system performs just as well as the human operator. The sole prediction matches well with a weed plant that is largely hidden by grass. In fact, this example may be the most important in that it demonstrates the potential of the system to locate the weed center in a near homogeneous background. As aforementioned in the literature review, this project sought to address challenges largely absent from agricultural datasets such as the BoniRob [23] set.

5.7 Testing Analysis for the Machine-learning Approach

A brief summary of the dataset used for each pipeline component was given. Image subsets extracted from the original set tended to perform worse as they became smaller (e.g. leaf-wedge to single leaf.) The different types of dandelion plants belonging to the weed class were brought

explained, as were the distinct plants that inhabited the ‘not-weed’ class. The data capture method and hardware were presented.

The testing process represented the individual testing of the pipeline elements followed by the integrated system testing. The individual components were each trained and tested with different configurations. Firstly, the initial weed plant classifier required basic data augmentation to achieve high generalization capabilities. Subsequently, the leaf classifier was trained on two distinct subsets of the original images, one of which was manually extracted and performed well due to higher overall resolution. However, when coupled with the RPM component, it failed to yield consistent results. This made necessary the alternate dataset that was generated by the RPM and that was significantly lower in pixel resolution. Thankfully, it achieved acceptable results. Additionally, several structures involving a four-class leaf-pair classifier were intended to discriminate between types of joined leaf-regions. A capacity for learning was demonstrated yet performance plateaued at accuracies insufficient for system integration. Next, a CNN designed to output coordinates was built to include 12-layers in total. Furthermore, data resolution played a familiar role in training as test accuracy dropped when introduction lower resolution data though performance remained within acceptable limits. Finally, the satisfactory pipeline elements were integrated and tested as a system whole.

Overall, the integrated system successfully addressed many of the challenges of locating dandelion weeds in grass under natural lighting conditions such as the problem of a homogenous background, the locating of multiple weed instances in the same image, the weed centers that are partially hidden from view, etc. In addition, integrated testing has proved that leaf orientation is very important in locating dandelion plant centers. System drawbacks included sensitivity to an oversaturation of light, the ambiguous nature of the data, the low overall image resolution, and

other limiting factors. To summarize further, the system is only as good as its weakest component. Since the original resolution of input image was low, subsequent steps that depend on extracting subsections of that input were negatively affected. That being said, testing has proved that the overall system is definitely a viable option for detecting dandelion weeds in grass and is open to many possible improvements.

5.8 Summary of Testing

In this section, each component of the machine-learning pipeline was tested individually and then the system was tested as a whole. The individual component tests were all successful save for the weed-pair classifier. For that reason, it was not included in the integrated system testing. The successful component tests were comprised of the weed classifier, the weed-leaf classifier, and the weed-leaf point predictor.

The final system testing proved successful in accurately identifying and detecting weed centroids for a range of conditions. However, testing also revealed system drawbacks that can still be improved upon.

Chapter 6: Conclusion and Future Works

6.1 Conclusion

The project work aims to develop a mobile robot sensor capable of accurately detecting dandelion weed centers in plants for eventual mechanical removal by a mobile robot. Two general approaches, developed one after the other, are presented. The first is a pattern recognition methodology that uses colour isolation, frequency filtering, and attribute morphology amongst other things to determine weed plant locations in grass. It proposes a novel method using the binary region orientation of presumed leaves to interpolate intersection points that serve as candidates for the plant center. This method performs well in a controlled environment; however, when put to the test against real world data, performance rapidly deteriorates with image variability. This introduces the need to develop methods that are robust to real-life situations where a large range of variability naturally occurs.

The machine learning methodology addressed many of the issues facing the naïve assumptions and heuristic nature of the pattern recognition approach. In fact, the pipeline, tested on a set of 263 images, achieved mean maximum accuracies of 92% and 93%. These values are based off minimum RMSE and MAE computations respectively. Although promising, there are still considerable system limitations that need to be addressed.

The CNN based system performed for a far wider range of data than the pattern recognition approach; however, it still was not completely robust to oversaturation of light, low-resolution inputs, etc. This led to the conclusion that the CNN pipeline's performance suffered from being only as good as that of its weakest link. For example, misclassified items by the leaf-classifier (second CNN) would propagate the error along the centroid-point predictor (third CNN) and reduce system accuracy. Furthermore, CNNs in the system are dependent on extracted samples from the original 640p input image. These are drastically reduced in size and heavily compromise pipeline performance. In an attempt to deal with this, a four-class classifier was created but it could not properly deal with the ambiguity of its respective classes.

In conclusion, the project work appears to address initial issues discussed with reasonable success. By treating a problem distinct in nature from current agricultural weeding research, it presents novel solutions that have not yet been explored in a domestic weeding setting.

6.2 Future work

Essentially, there are countless points for potential improvement where future work should be explored. First and foremost, a new set of high-resolution data should be gathered of at minimum 1080p. That being said, current options of 4K (near 4000p) image acquisition are being explored. These present huge possibilities of improvement for system performance. Additionally, acquiring data in the near infrared (NIR) is a concurring avenue of research that would potentially diminish the problems caused by data variability under natural lighting. A final point for the matter of the data, 3D vision data should be attempted using photometric stereo (PS) or at least RGB-D cameras. These will potentially offer additional depth vector gradients that would constitute a useful attribute in the morphology of leaves

In terms of the inherent structure of the system, developing an end-to-end CNN structure that is fully trainable using an encoder-decoder structure is definitely an attractive avenue of exploration. This would reduce and possibly eliminate the disadvantages of propagated error in the current step-by-step pipeline model. Furthermore, the successful implementation of a pixel-wise classifier could possibly replace the attempted four-class classifier and yield even more accurate and robust results.

Finally, it would be very interesting to further pool predicted pairs together (as was done from single leaves to leaf pairs) to identify as much of the plant as possible before making yet another predictions. A procedure of this type would result in the current iteration of predictions generally receiving larger input images than the previous iteration and, consequently, make less but more accurate predictions. That is, a single prediction would be generated from a higher resolution input image. This diminishes the need for a pooling scheme that simply averages out clusters of prediction points or any such weighted scheme.

References

[1] L. Boubouleix, M.-D. Aghazadeh-Tabrizi, M. Mazumder, H. Ijaz, M. De Celis, D. Shehata and D. Fernandez, "Mech 490: Team 21 Weeder Bot 2, Final Project Report," Montreal, QC, 2020.

[2] K. He, X. Zhange, S. Ren and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *ICCV,* 2015.

[3] A. Krizhevsky, I. Sutskever and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Proceedings of the Advances in Neural Information Processing Systems 25,* pp. 1097-1105, 2012.

[4] Y. Le Cun, F. Huang and L. Bottou, "Learning methods for generic object recognition with invariance to pose and ligting," *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on,* vol. 2, pp. 11-97, 2004.

[5] R. E. Turner, *Lecture 14: Convolutional neural networks for Computer Vision,* Cambridge: Cambridge University of Engineering, 2014.

[6] J. Deng, W. Dong, R. Socher, L. Li, K. Li and F. Li, "Imagenet: A large-scale hierarchical image database," *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition,* pp. 248-255, 2009.

[7] K. Simonyan and Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv,* no. arXiv:1409.1556., 2014.

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

[9] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015.

[10] U. Alganci, M. Soydas and E. Sertel, "Comparative research on deep learning approaches for airplane detection from very high-resolution satellite images," *Remote Sensing,* no. 12, p. 458, 2020.

[11] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colombus, OH, USA, 2014.

[12] M. Everingham, L. Gool, C. Williams, J. Winn and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *International Journal of Computer Vision,* no. 88, pp. 303-338, 2010.

[13] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 2015.

[14] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015)*, Montreal, QC, Canada, 2015.

[15] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu and A. Berg, "SSD: Single Shot MultiBox Detector," in *Proceedings of the European Conference on Computer Vision*, Amsterdam, The Netherlands, 2016.

[17] T. Lin, M. Marie, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. Zitnick, "Microsoft COCO: Common Objects in Context," in *Proceedings of the European Conference on Computer Vision*, Zurich, Switzerland, 2014.

[18] X. J., J. Hays, K. A. Ehinger, A. Oliva and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[19] A. Geiger, P. Lenz and U. R., "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[20] D. N. and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[21] P. Lottes, J. Behley, N. Chebrolu, A. Milioto and C. Stachniss, "Joint stem detection and crop-weed classification for plant-specific treatment in precision farming," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[22] S. J´egou, M. Drozdzal, D. V´azquez, A. Romero and B. Y., "The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation," *arXiv preprint,* no. arXiv:1611.09326, 2016.

[23] N. Chebrolu, P. lottes, A. Schaefer, W. Winterhalter, W. Burgard and C. Stachniss, "Agricultural Robot Dataset for Plant Classification Localization and Mapping on Sugar Beet Fields," *Intl. Journal of Robotics Research (IJRR),* 2017.

[24] P. Bosilj, T. Duckett and C. G., "Connected attribute morphology for unified vegetation segmentation and classification in precision agriculture," *Computers in Industry, Special Issue on Machine Vision for Outdoor Environments,* vol. 98, pp. 226-240, 2018.

[25] J.-C. Klein, *Conception et réalisation d'une unité logique pour l'analyse quantitative d'images (Ph.D, thesis),* 1976.

[26] L. Vincent, "Morpholohical grayscale reconstruction in image analysis: applications and efficient algorithms," *IEEE Trans. Image Process,* vol. 2, no. 2, pp. 176-201, 1993.

[27] O. 4.40-dev, "OpenCV python Tutorials: Morphological Transformations," [Online]. Available: https://docs.opencv.org/trunk/d9/d61/tutorial_py_morphological_ops.html. [Accessed 20 07 2020].

[28] E. J. Breen and R. Jones, "Attribute openings, thinnings, and granulometries," *Comput. Vision Image Understand.,* vol. 64, no. 3, pp. 377-389, 1996.

[29] P. Salembier, A. Oliveras and L. Garrido, "Antiextensive connected operators for image and sequence processing," *IEEE Trans. Image Process,* vol. 7, no. 4, pp. 555-570, 1998.

[30] C. Cortes, P. Raghavan and H. Schutze, "Support-vector networks," *Machine Learning,* vol. 20, no. 3, pp. 273-297, 1995.

[31] P. Lottes, M. Hörferlin, S. Sander and C. Stachniss, "Effective vision-based classification for separating sugar beets and weeds for precision farming," *Journal of Field Robotics,* vol. 34, no. 6, pp. 1160-1178, 2016.

[32] A. Binch and C. Fox, "Controlled comparison of machine vision algorithms for Rumex and Urtica detection in grassland," *Comput. Electron. Agric.,* no. 140, pp. 123-138, 2017.

[33] A. Binch, N. Cooke and C. Fox, "Rumex and Urtica Detection in Grassland by UAV," in *Proceedings of the 14th International Conference on Precision Agriculture*, Montreal, 2018.

[34] L. Smith, W. Zhang, M. Hansen and e. al., "Innovative 3D and 2D machine vision methods for analysis of plants and crops in the field," *Comput Ind,* no. 97, pp. 122-131, 2018.

[35] D. Travis, Effective Color Displays. Theory and Practice., Academic Press, 1991.

[36] I. Babiker, W. F. Xie and G. Chen, "Recognition of Dandlelion Weed via computer Vision for Weed Removal Robot," in *The 1st International Conference on Industrial Artificial Intelligence (IAI 2019)*, Shenyang, China, 2019.

[37] R. Woodham, "Photometric method for determining surface orientation from multiple images," *Opt Eng,* no. 19, pp. 139-144, 1980.