Developing a Seed Segmentation Algorithm which Incorporates a Deep-Learning Architecture

Peter Leonard Hart

HAR12421031

MComp Computer Science

Supervisor: Dr. Wenting Duan

26th April 2018

ACKNOWLEDGEMENTS

The author would like to thank Professor David WH Rankin from the University of Edinburgh for supplying the large seed sample image dataset which played a crucial role within this project, the images were provided in a high resolution which proved to be very useful for the various algorithms explored in this project.

The author would also like to take this opportunity to provide a significant mention to the project supervisor, Dr. Wenting Duan, for the extensive support and guidance provided throughout the entire project lifecycle. The eloquent advice and excellent supervision was inspiring and played a fundamental role toward the motivations behind this project. Thank you for the dedication and enthusiasm towards the successful completion of this project.

ABSTRACT

The following dissertation reviews the development of an image segmentation algorithm using deep-learning strategies and its possible application within the field of botany image processing. Ultimately, this project followed the development of a superpixel convolutional neural network image segmentation algorithm, an algorithm which theoretically should combine the benefits of both techniques. This algorithm was applied to a series of different samples of seed image data, whereby the algorithm performance was assessed based upon the estimated confusion matrix and subsequent classification accuracy from a test dataset of images. Therefore, it is hoped that this project will provide botanists with means toward effective seed object extraction within seed samples, such that this useful data can be used by various botanists for further photometric analysis. In conclusion, the results of this project demonstrated clear capability toward accurate segmentation of the seed objects with the convolutional neural network classifying the seed and background superpixels with a 95.84% classification accuracy, which in turn allowed accurate segmentation to be procured on the seed sample images. However, the output segmentation accuracy was estimated to be an underwhelming 73.18%, therefore further work on model refinement and post-processing are evident.

CONTENTS

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND TO THE PROJECT

Photometric analysis of seeds entails the concept of identifying and analysing key characteristics of each seed, identifying descriptive traits such as seed colour, seed texture or the angularity of the seed. However, with this, an imperative issue introduced is the consideration for the countless array of different plant species which the various seeds may originate from, and therefore the identification and analysis of each seed can be considered as an expensive operation in regard to time complexity if carried out manually.

The basis of this project explores this concern and aspires to develop an algorithm which is capable of automatically identifying the plant species for each individual seed that may be present in each seed image sample in the dataset, performing accurate image segmentation for the extraction of each seed object based on the data which the algorithm has learned from a given training dataset of seed images; hence a deep-learning approach. Imperatively, the automated system for the analysis of the seed sample images should provide an interface for seed experts and non-experts to be able to perform necessary analytical operations on a large dataset of seed images in a timely manner.

However, a technical challenge that will be faced is the development processes involved with deep-learning strategies and techniques, this is largely due to an initial lack of experience in regard to the development of this type of system architecture. On the other hand, this challenge contributes towards the motivations behind this project, as this project presents a great opportunity to learn more about the field of deep-learning and its applications within computer science.

Previous research in this field of study indicates that an approach for accommodating this concern would be to develop a system which is inspired by the cognitive functions of a biological brain, otherwise referred to as an artificial neural network (ANN) (Egmont-Petersen et al, 2001). However, a convolutional neural network (CNN) is a form of ANN which focuses solely on the functional replication of the visual cortex of a biological brain. Furthermore, research was also indicative of the fact that the performance of the system architecture is largely dependent on the data which is passed through as a parameter for the CNN, for example larger patches of seed image training data would likely improve the accuracy but decrease speed of the algorithm (Mori, 2017). In addition, the complexity of the developed deep-learning architecture will also contribute to the overall performance of the algorithm, for example more layers of convolutional tasks could lead to a higher time complexity for the system performance. To this end, research also suggested that this could be amended by developing the algorithm through the utilisation of the graphics processing unit (GPU) device, which would allow some of the deep-learning tasks to be parallelised and operate more efficiently (Mori, 2017).

To surmise, the primary deliverable of this project will be a feature extraction and seed segmentation algorithm using a deep-learning CNN system architecture, the success of which should provide seed experts and non-experts with the opportunity to analyse key characters of various seed samples with a high performance. The core of this project envelopes the concept of testing how accurate the system will be able to correctly estimate seed characteristics such that accurate seed segmentation, while considering the speed of the system.

1.2 PROJECT AIM AND OBJECTIVES

Predominantly, the aim of this project is to develop a seed segmentation and feature extraction algorithm while utilising a deep-learning architecture. This algorithm was designed to perform accurate feature extraction by learning characteristics from a training dataset of seed sample images, this data will then be utilised for the extraction of all seed objects that may exist within an image to enable further in-depth image analysis for any botanists that may subsequently use this data when analysing additional characteristics of each seed.

Explained below are the objectives, which upon completion, will help accomplish this aim:

1. Experiment with existing deep convolutional neural network architectures and evaluate their effectiveness for the problem at hand.
2. Develop a customised deep convolutional neural network architecture such that the algorithm is capable of learning key characteristics from a given training dataset of images.
3. Develop the algorithm with the extraction of photometric features. The system should be capable of identifying key features of each seed foreground object that can be observed in the image to provide further analysis.
4. Test and evaluate the developed architecture and ensure that the system performance is sufficient for achieving a satisfactory performance with the final segmentation output.
5. Adapt the algorithm using parallel programming and the graphics processing unit (GPU). Parallelising some segments of the developed algorithm should theoretically enable a significant performance improvement in regard to the processing capabilities of the deep convolutional neural network that was developed. However, it should be noted that this objective is being considered as a STRETCH objective, and therefore development towards the completion of this objective will only begin when the prior objectives have been completed to a satisfactory standard.
6. Design and develop an easy-to-use graphical user interface (GUI) for any non-experts which may use this architecture. This is considered as a STRETCH objective and will only enter development upon prior objectives of the project being completed to a satisfactory standard.

CHAPTER 2: LITERATURE REVIEW

2.1 TRADITIONAL IMAGE SEGMENTATION TECHNIQUES

Within the field of image processing and computer vision, image segmentation can be considered as the approach undertaken to partition segments of a particular image into multiple segments of pixels. As defined by Shapiro and Stockman (2001), the fundamental goal of image segmentation techniques is to simplify the representation of an image for a particular problem which is to be solved by the algorithm, for example segmenting any seed objects contained within the seed image samples involved in this project. Furthermore, one approach toward accomplishing image segmentation is through the application of traditional image segmentation techniques, which are hand-crafted algorithms tailored a particular segmentation problem such as simple linear iterative clustering.

Ren and Malik (2003) define the principle of intra-region similarity as elements in a region which may have a similar brightness, texture and a low contour energy inside the region, whereas intra-region dissimilarity can be described as the elements in a region which may have a contrasting brightness, texture and a high contour energy inside the region. superpixel segmentation can be derived as a popular image processing technique which operates by dividing the pixels of an image with similar visual characteristics into an atomic region, effectively replacing the original structure of the pixel grid.

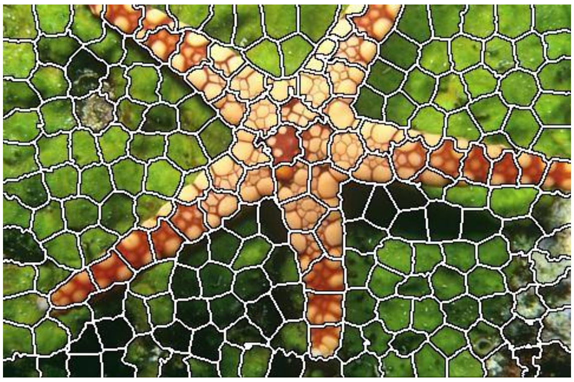

Figure 1. Example of SLIC being applied on an image of a starfish. (École Polytechnique Fédérale de Lausanne, 2017).

Furthermore, a study was conducted which investigated the development of a superpixel segmentation algorithm referred to as simple linear iterative clustering (SLIC), a superpixel segmentation approach which adapts the k-means clustering approach to efficiently generate superpixels. Achanta et al (2012) describes superpixel segmentation as an algorithm capable of capturing image redundancy and provide an effective means for computing image features. This in turn can reduce the complexity of any subsequence image processing tasks that may be applied in later phases of a system. For example, this algorithm could be partitioned into a deep-learning model to enhance the feature extraction capabilities, such as extracting any features from the seed sample images involved in this project.

In a similar study, Wang et al (2017) conduct an investigation into a superpixel segmentation algorithm which was based on multiple seed growth, advocating a solution which was high in segmentation accuracy and performed at a linear execution time compared to other superpixel segmentation algorithms. Furthermore, this study advocated that superpixel segmentation supplements the advantage of ensuring that the superpixels are generated at a more relative segmentation size distribution in comparison to rivalling methodologies.

One requirement for deep-learning architectures such as convolutional neural networks (CNN) is that the input images must be sized to a normalised size, such that all of the images are the same size and dimensions. Therefore, the studies conducted by Achanta et al (2012); Wang et al (2017) could suggest that the similar size attribute of the generated superpixels could be advantageously used by deep-learning models, where theoretically it could utilise the high segmentation accuracy provided through the superpixel segmentation algorithms such as SLIC and integrate the superpixel outputs as a possible input for a deep-learning model.

Chitra et al (2016) conducted a comparative study for alternative, classical image analysis techniques. Overall, this study found that image processing techniques such as histogram thresholding present the advantage of not requiring any previous image information and emphasises on strong edges within the image, however this technique cannot be applied to multi-channel images and foreground objects can distort the image histogram. Furthermore, the watershed segmentation technique can efficiently merge essentials from both discontinuity and relationship-based systems, yet this technique is known to create over-segmentation due to local irregularity within the image.

Furthermore, existing state-of-the-art image segmentation techniques such as mean-shift and graph-cut are able to procure a lower time-cost segmentation solution, however these techniques suffer from the weakness of allowing the superpixel size and shape to be inconsistent and have a high variance. To this end, Wang et al (2017) perform a study entailing the development of a superpixel algorithm which minimises the cost function through a minimum spanning tree cost function based on graph theory. This algorithm divides the superpixels by region growth based on the seed points that were pre-determined, using an energy function which considers both the colour difference and a distance factor. Overall, the conclusion of this study found that the algorithm was capable of generating superpixels with the same colour and size, the solution of which was found to be robust and capable of operating with different totals of superpixels and is capable of running consistently with either a small or large number of seed points.

In conclusion, segmentation techniques such as automatic histogram thresholding could automatically segment the seed objects away from other entities in the image without being required to learn commonalities in seed features from a large dataset of seed image samples. On the other hand, deep-learning architectural approaches such as the LeafNet CNN approach established by Barré et al (2017) demonstrate potential of deep-learning strategies by achieving a high performance on a complex dataset. In conclusion, this research could be argued to show that a deep-learning approach for the system architecture could be a more accurate means for processing the sample seed images involved with this project, despite a higher time complexity. However, the region segmentation simplicity of superpixel segmentation approaches such as the SLIC algorithm suggest a possible methodology toward segmenting an input image into regions of almost equal size, therefore inferring a possible integration between superpixel techniques and deep-learning strategies.

2.2 APPLICABILITY OF DEEP-LEARNING ARCHITECTURES

Artificial Neural Network (ANN) is a technique largely founded upon an attempt to replicate the methodologies performed by a biological brain to process data. However, a study was conducted which involved a comprehensive review of more than two hundred applications of neural networks being incorporated into modern day system architecture. Egmont-Petersen et al (2001) argue that while ANNs are not automatically seen as the best solution towards most classification problems, they have been largely incorporated into various applications which inspired its creation: pattern recognition, psychology and neurophysiology. This study also suggests that a challenge for ANNs is the issue of the feature complexity and subtle discrepancies between images, such as variations in position, orientation and scale.

The term "big data" refers to extremely large datasets that have been procured as a result of advancements in data storage capability, increase of computational power and more data volume accessibility. As such, Sutton and Barlow (1998) suggested that deep-learning architectures provide the capability for effective analysis and abstraction of useful knowledge from both large amounts of data and data collected from difference sources. To this end, the utilisation of big-data provides the opportunity for more patterns and correlations to be detected and analysed by the deep-learning model, hence; more data could provide a higher level of data abstraction.

Furthermore, a research survey investigates the availability and applicability of big-data in regard to any deep-learning strategies that could be employed, Gheisari et al (2017) suggests that most of the current technologies are striving to investigate six prominent characteristics of big-data: volume, velocity, variety, veracity, validity and volatility. To this end, the application of deep-learning methodologies and frameworks are described to primarily concern two of the previously described characteristics of big-data: volume and variety. As such, Gheisari et al (2017) argue that the application of deep-learning strategies is effective when attempting analyse and learn useful knowledge from large amounts data as well as data collected from a variety of different sources, such that a high performance can be performed by a deep-learning model when a sufficient amount of data is provided.

A research study explores the concept of computer-aided image analysis techniques which are considered as contributing towards improving the insight of seed analysis, such as image acquisition and pattern recognition. Varma et al (2013) suggest that the combination of image analysis and human intelligence emulation is an important technology that will aspire to many applications in modern varietal identification and seed certification. It is proposed that a current major research area is the encapsulation of human intelligence, suggesting that development in image acquisition, pattern recognition and decision-making techniques will help to improve existing systems while providing a better return on investment and reduced costs.

In a research study conducted by Zhong et al (2018), deep-learning algorithms are described as presenting a major breakthrough within the field of artificial intelligence and can be considered as the current hotspot of recent research, particularly as deep-learning strategies are being incorporated into many real-world applications at an impressive rate. For example, deep-learning algorithms are applied for image classification, object detection and speech recognition, whereby traditional or state-of-the-art approaches toward these problems can be argued to be vastly improved through deep learning.

Furthermore, taxonomic identification is the recognition of the identity of an organism, taxon identification is an imperative step within plant ecological studies the efficiency and reproductivity of which may benefit from automation of this task. The process of manual plant species identification and analysis can be argued to be difficult, time-consuming and erroneous for non-experts. While a deep learning approach may require a large input dataset, it could allow reduced costs for any plant experts and non-experts that may use this seed segmentation and feature extraction algorithm.

2.3 EXISTING DEEP-LEARNING NEURAL NETWORK SYSTEMS

Similar research followed the development of an automatic segmentation algorithm for fluorescence micrographs which incorporated a fast-learning neural network for the detection of fluorescent cells. Nattkemper et al (2002) suggest that the evaluation of a large number of micrographs by human experts is time consuming and nearly impossible due to the observer's concentration naturally declining rapidly during the visual inspection of the noisy intensity images. A pre-trained neural network is utilised for the initial position detections of the various fluorescent cells in the image, this consists of a neural network of local linear map type (LLM) which is trained through a set of image patches that contain fluorescent cells. This output is subsequently incorporated into the second neural network: a recurrent neural network. The initial positions that were detected in the previous phase of the system architecture are considered as "focus points" and are used to guide the recurrent neural network when attempting to extract the cell contours in each image. This approach was proposed as the lymphocyte objects in the image show significant variation in shape, so state-of-the-art approaches would make it difficult to define a single common contour model. Overall, the system was evaluated to be highly accurate with reproducible results and shows a possible methodology for automatic evaluation of high-throughput topological screening of lymphocytes.

Zhao et al (2009) developed an automatic mass peanut seed detection algorithm through appearance characteristics of each peanut seed. This entailed the process of performing component analysis for evaluating the contribution rate of each variable, which would subsequently be input as component parameters for the neural network model of the algorithm. Zhao et al (2009) found that the automatic mass peanut seed detection algorithm achieved a peanut seed variety recognition rate reached 91.2% and a quality recognition rate of 93.0%. This study surmised that the developed peanut detection algorithm through a machine vision approach possessed cost and speed advantages and could be considered for the identification of peanut cultivars and quality.

However, convolutional neural networks (CNN) are a form of ANN which are largely inspired by the biological processes and organisation of the visual cortex of a biological brain. A CNN structure is largely comprised of an input and an output layer, with a variable number of hidden layers depending on the desired complexity of the system. The hidden layers can include convolutional, pooling or fully-connected layers.

A recognised issue with current customised, specialised and hand-crafted identification systems is the expense of these approaches, whereas a deep CNN approach provides an alternative system architecture which permits adaption to different taxa by training the algorithm with different training data. Barré et al (2017) explains that CNNs learn automatically to extract sets of features by repeatedly presenting the algorithm with training data: the original image data and the respective classes that any features may be identified as. As the CNN is rigorously trained with a sufficient amount of data, the internal parameters are adapted as the algorithm begins to learn more about how to process the dataset it is provided. Thus, a deep CNN system entitled “LeafNet” was developed and was designed to learn discriminative features from leaf images taken from plants and establish plant species identification for each sample, the neural network architecture of which is comprised of twenty-six different layers (see figure 2).

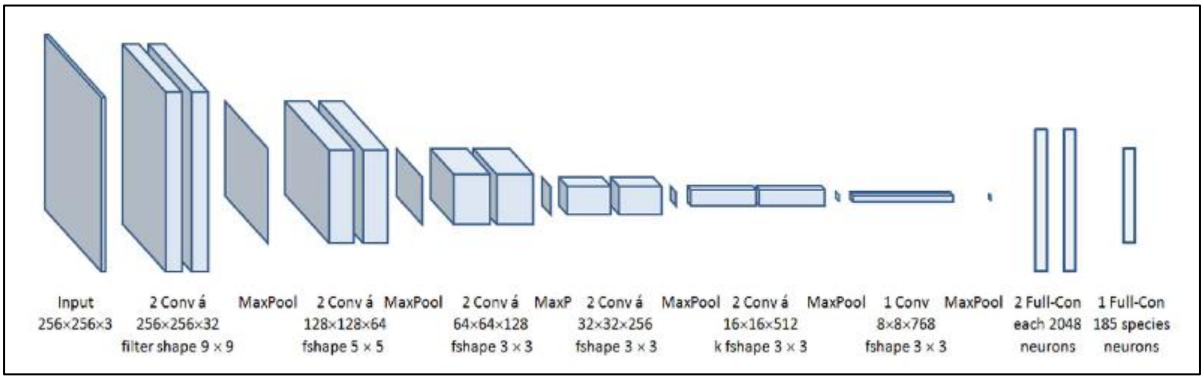

Figure 2. Image showing the overall network structure of the LeafNet model. (Barré et al, 2017).

Furthermore, Barré et al (2017) compared the performance of the LeafNet CNN model with current state-of-the-art approaches when operating on several image datasets containing a various amount of different plant species. Specifically, the LeafSnap dataset comprised of 184 different classifications, the Foliage dataset comprised of 60 different classifications and the Flavia dataset comprised of 32 different classifications. Overall, it was found that the LeafNet CNN model demonstrated superior accuracy performance, the results of which concluded that learning features through CNN provided better feature representations for leaf image datasets when compared to hand-crafted feature approaches for plant species identification.

A similar study investigates automated plant species identification through the utilisation of machine learning algorithms, specifically investigating the overall effects sustained when alterations are made with various pre-trained neural networks. However, Ghazi et al (2017) infer that while deep learning architectures have demonstrated success, training deep neural networks which contain millions of parameters require extremely large amounts of data in the order of millions of samples to ensure optimisation.

As such, this study explores the concept of continuing to abstract high amounts of knowledge from big-data through an approach referred to as transfer learning, a technique which amasses previously learned from a different problem and applies it to a new problem. To this end, Ghazi et al (2017) advocates that this could be achieved through the utilisation of a pre-trained neural network by using pre-existing weights to extract features in a new problem, specifically by utilising the output from all layers prior to the final fully-connected layer of a neural network model. On the other hand, it is also advocated that this could be accomplished through fine-tuned weights where the pre-trained neural network weights would undergo retraining with the new dataset, although a requirement enforced by this approach is that the number of output layer nodes must be equal to the total number of classifications in the new problem.

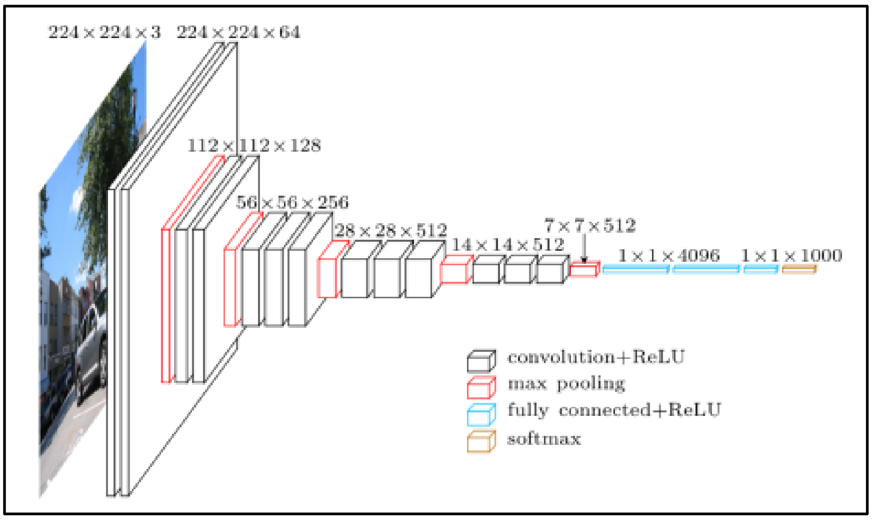

Figure 3. Image depicting the neural network architecture of VGGnet16. (Frossard, D., 2016).

Upon further experimentation with different pre-trained neural network models and the application of transfer learning techniques, Ghazi et al (2017) discovered that fine-tuning the VGGNet model obtained an underwhelming performance of 78.44% accuracy which contrasted against the performance of the AlexNet model which was trained from scratch. In conclusion, this study found that while there are benefits to be gained through transfer learning, training simpler neural networks from scratch could provide more novel and computationally efficient neural networks due to the lower complexity of the model. Therefore, this research could be indicative that while transfer learning is a possible technique which could be utilised within this project to ensure that the weights of the model have been sufficiently trained, the efficiency of such a technique is to be questioned depending on the desired complexity of the neural network model that is involved with this project.

In conclusion, this research suggests that deep-learning strategies, such as the various CNN models which have arisen over recent years, can provide impressive classification performance on datasets of various complexities. Therefore, this could suggest that applying a deep-learning approach for the classification of seed images could achieve a high performance, whereby a deep-learning approach could utilise a training dataset of images to estimate unknown features on a new dataset of images. For example, deep-learning could provide the opportunity to identify new seed species that may be contained a more recent seed sample image, possibly providing a basis for an effective and robust system architecture for this project.

2.4 DEEP-LEARNING WITH A GRAPHICS PROCESSING UNIT (GPU)

A study researched a deep convolutional neural network for the classification 1.2 million high-resolution images from the ImageNet dataset into a thousand different classifications as part of the ImageNet LSVRC-2010 contest. ImageNet is an extremely large dataset comprising of over fifteen million labelled images belonging to approximately 22,000 different classes.

The GPU-enabled CNN architecture was designed with five convolutional layers accompanied with max-pooling layers and three fully connected layers. Krizhevsky et al (2012) argued that the utilisation of a single GPU device would severely limit the maximum potential size of the network which could be train on it, as a training dataset comprising of approximately 1.2 million image samples were found to be too big to fit onto a single GPU device. Thus, the architectural design of this CNN model has been implemented such that the delineation of the model operations has been delegated between a total of two Nvidia GeForce GTX 580 GPU devices (see figure 4), thereby operating under a parallelisation framework. As such, the parallelisation strategies employed emplace half of the neurons between both of the GPU devices, where the GPUs communicate and synchronise only in specific layers of the CNN model. For example, Krizhevsky et al (2012) elaborated as an example that all filter maps within hidden layer 3 take input from all filter maps that reside within hidden layer 2, whereas filters within hidden layer 4 only take filter maps from hidden layer 3 which exist on the same GPU device. This study found that the CNN model performance with the test dataset achieved an error rate of 37.5% and 17.0%, top-1 and top-5 respectively.

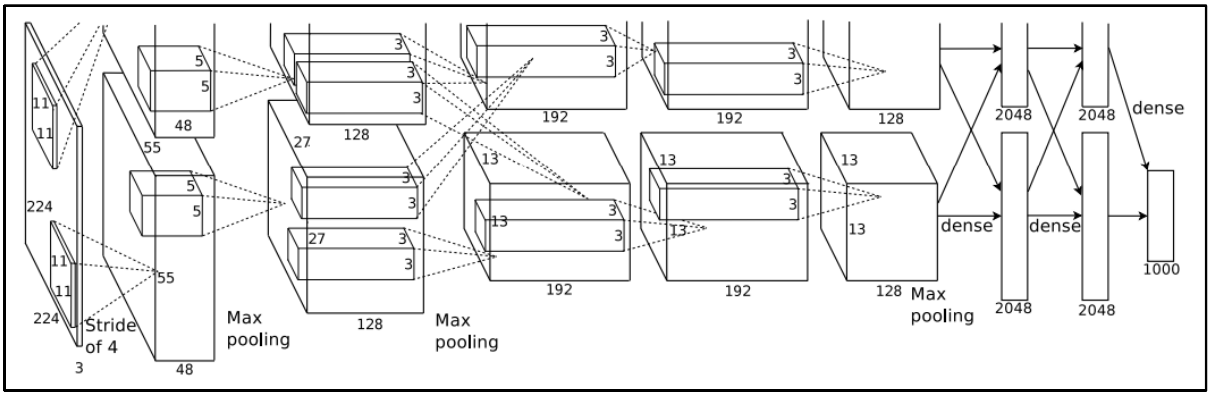

Figure 4. Image showing a network structure for a GPU-enabled CNN model. (Krizhevsky et al, 2012).

Overall, this study concluded that a large, deep CNN provide the capability toward record breaking results on a highly challenging dataset using supervised learning methodologies. However, Krizhevsky et al (2012) conclude that while the results were shown to have been improved, further development is still required to match the infero-temporal pathway of the human visual system.

Similarly, further research indicates that while ANNs are becoming more popular in acoustic model training, the speed of the model performance can be improved by utilising the GPU device. To this end, a back-propagation (BP) neural network acoustic model for speech recognition is developed and utilised on the GPU device, therefore the application of an asynchronous implementation between the CPU and GPU alongside parallel reduction could be applied on some of the operations involved within the BP neural network. As such, Liu et al (2012) concluded that the training of the BP neural network was accelerated 26 times faster than using a single thread Intel Math Kernel Library implementation, this as a result enabled the opportunity for more data and more complex neural networks to be developed. Thus, this could suggest a performance improvement in regard to the recognition accuracy and speed of the algorithm that is going to be developed in this project if GPU-enabled processes are considered.

Mori (2017) developed a real-time image processing algorithm for image-guided radiotherapy, the purpose of this study was to explore the implementation of different neural network models with different imaging modalities and consider the possibility of a real-time neural network architecture. In this study, a residual convolutional neural network (rCNN) is trained which is comprised of multiple sets of convolutional, BN and ReLU layers, with the last layer being a convolutional layer with a single feature map and an input image. Mori (2017) found that while it was possible to develop the rCNN model in real-time image processing, it was also established that the image quality needs to be adjusted in consideration of the image size and the fluoroscopic frame rate, as in some cases the results suggested a performance which was under 30fps and therefore deemed unsuitable for real-time processing.

In summary, this research suggests that developing a system architecture which utilises deep-learning strategies, for example a CNN, could demonstrate a possible approach which helps to neglect the strenuous time complexities enforced when training a large neural network architecture. For example, Krizhevsky et al (2012) developed a CNN which operated intermittently based on two GPU devices, which not only enabled a speedup with the training time of the model but also demonstrated the possibility of loading a much larger neural network between the two devices. Therefore, this research suggests that developing a GPU-enabled CNN architecture for the seed sample images involved in this project could allow for the development of a solution with increased efficiency, as the CNN operations could be delegated between two or more GPU devices.

CHAPTER 3: METHODOLOGY

The following chapter will entail a discussion regarding the strategies employed when attempting to establish a suitable methodological approach towards managing and completing the project throughout the entirety of the project lifecycle. Specifically, this will involve a discussion regarding the devised approaches undertaken for subjugating the various priorities of the project, such as strategies regarding project management, software development, toolsets and machine environments and research methods.

3.1 PROJECT MANAGEMENT AND SOFTWARE DEVELOPMENT

Allocation of project resources was managed and pre-planned for the project through a project plan framework referred to as a Gantt chart. Wilson (2003) argues that Gantt charts provide an effective means for displaying important information and useful for implementing interactive approaches to scheduling in a project. As such, a project plan document in the form of a Gantt chart was contrived to schedule an appropriate amount of project resources, namely project time, relative to each major task involved in the project. The completion of this document was designed to ensure that the project aim, and objectives can be satisfied within a realistic timeframe, constant reference to which provided an effective means a constant monitoring of project progression.

The Gantt chart involved within this project was designed under a one-week granularity, whereby project resources were delegated based upon the estimated task difficulty and in reference to the preliminary risk assessment, for example this lead to a larger amount of project resources being allocated toward tasks which propose a higher risk of failure in comparison to other tasks, for example it was suspected that more time may be required for the development of a CNN architecture due to a lack of experience and knowledge within this field of interest.

As previously discussed, the basis of this project was to deliver a software solution which could provide seed experts and non-experts with the opportunity to efficiently segment images of seeds originating from various plant species, which in turn should aid any further analysis that may be performed on subsequently extracted seed objects (see chapter 1). An effective direction toward the project management strategies would advocate the quality of the developmental processes that are employed throughout the entirety of the project, the outcome of which would determine the overall quality of the final outcome accomplished with the project.

A waterfall project management paradigm was considered. As such, the project management methodology could follow this paradigm by designing the project lifecycle as a single cycle of developmental processes, which are enlisted based on an initial set of unmodifiable requirements that were elicited by a client during the initial establishment of the project.

This paradigm was ultimately rejected for the project management strategies to be employed during this project. Disadvantageously, this paradigm would introduce a significant element of risk into the project management of this particular project, as the software development lifecycle operating under a single cycle of developmental processes would in turn lead to limitations in regard to the possibility of regressing to previous phases of the software development lifecycle. As such, this can be considered as an imperative project management prerequisite for this project, fundamental components of the developed software solution, such as the development of the convolutional neural network (CNN) architecture, required rigorous testing for the necessity of refinement of the neural network parameters and hyperparameters. Furthermore, the development of the CNN under a waterfall project management paradigm would infer that the testing phase would not commence until the final phases of the software development lifecycle, the neglection of which could have led to a far less robust solution which may have produced a much lower performance as a result.

In contrast, an agile project management paradigm advocates the developmental processes of the project to be delegated amongst multiple small software development lifecycles and consequently deliver the final outcome of the project in small iterations (Cockburn and Highsmith, 2007). This approach adapts well to change during the software development lifecycle, such that the project management could design, develop, evaluate and continue to iterate through an extensive amount of project lifecycles until a desirable outcome is achieved with the project (Meso et al, 2006). As such, the software solution could be tested and evaluated at the conclusion of each lifecycle, the feedback of which would be communicated during the initialisation of the subsequent lifecycle in the event of any amendments being required to be made to the project requirements elicitation.

Despite this, an acknowledged weakness which is imposed on this project as a result of the agile paradigm is the consideration of the impact on the excess project resource expenditure, namely project time. Therefore, the continuous iterations of the small project lifecycles could in turn lead to an overall exponential increment in project resource usage if the project requirements are intermittently amended until a more desirable outcome is attained, an issue which can be argued to become increasingly prominent depending on the amount of small software development lifecycles that are employed. In comparison, a single software development lifecycle approach advocated through a waterfall paradigm would ensure a strict management on the project resources, as this approach would not allow for multiple software development lifecycles to be established.

However, as this project was completed as an individual project, grasping the full array of advantages presented from this project management methodology will be limited by being unable to take advantage of multiple team members contributing to the project. As such, the software development methodological approach utilised during this project was largely a theoretical approach which was derived from the fundamental principles defined within the agile software development paradigm. To this end, this entailed adhering to the principles by ensuring that each version of the solution would be tested using black-box and white-box testing techniques on the software solution upon the completion of each small software development lifecycle. For example, figure 5 demonstrates the advantage enabled through an agile-inspired software development paradigm as an issue arose during development where the image reconstruction phase of the project was encountering quality issues, however the final output segmentation was drastically improved as a result of copious amounts of testing cycles which in turn allowed a higher quality seed extraction at the outcome of the project.

Figure 5. A comparison showing the development improvements which resulted in a cleaner output segmentation due to testing phases in numerous project lifecycles.

Furthermore, an agile-inspired project management paradigm allowed a high amount of flexibility toward any requirements amendments that may need to be implemented as a result of testing, an aspect which would not have been possible through a waterfall inspired paradigm approach. For example, shortly after milestone 2 of the project, it was asserted that while an effective dataset of superpixels were being generated by the SLIC superpixel algorithm, the image depiction of each superpixel that was input into the subsequent CNN phase of the software solution could be observed as incorporating a significant amount of noise as a result of extracting the superpixel as a full block of pixels as shown in the leftmost example within figure 5. Consequently, this lead to the output mask generated by the superpixel CNN segmentation algorithm to have a rigid structure, irrespective to the true mask value of each seed object contained within each image when reconstructing the segmented output image. Imperatively, this can be considered as vital issue for the project due to the necessity to maximise the overall image segmentation accuracy performance, thus the inclusion of neighbouring pixels in the mask would lead to an undesirable impact on the algorithm's ability to provide a clean segmentation.

However, an agile inspired project management paradigm allowed this issue to be amended in the subsequent software development lifecycle by ensuring that only the pixels contained within each superpixel are extracted and setting all other pixels to a pixel intensity value of 0, effectively excluding other unnecessary pixels. Thus, as a result of copious amounts of testing being employed through an agile-inspired project management paradigm, a higher quality segmentation output can be observed with the segmentation structure appearing more fluent and respective to the true segmented mask output as the algorithm was continuously tested during multiple lifecycles (see figure 5).

In conclusion, while it can be acknowledged that the employment of an agile inspired project management paradigm would lead to an increased amount of project resources in comparison to a waterfall inspired project management paradigm, this issue was mitigated within the project plan by prioritising the tasks essential toward the accomplishment of the fundamental objectives featured in this project as opposed to the tasks completing any of the STRETCH objectives (see chapter 1.2). For example, an allotment of project resources that could have been delegated to tasks contributing toward the development of a graphical user interface (GUI) were instead allocated to tasks with a higher priority which contribute to the essential objectives of the project.

3.2 TOOLSETS AND MACHINE ENVIRONMENTS

The following section will entail a discussion regarding the methodological approaches enforced concerning the computational environments which were applied throughout the course of this project, the effectiveness of which can be argued to contribute to the quality of the solution delivered at the outcome of this project.

A vital aspect of the software solution which was developed throughout the course of this project is the development of the convolutional neural network (CNN) phase of the system architecture, whereby the CNN was tasked with the objective of learning discriminative features based on all of the input seed sample images that were collected to ensure effective classification. To this end, classification was applied using the open-source machine learning computational framework TensorFlow throughout the course of this project for the purpose of the numerical computations involved with the neural network.

Furthermore, deep learning software frameworks which were considered included an open source library referred to as Torch, which can be described as a machine learning framework based on the Lua programming language. Fonnegra et al (2017) advocate that the Torch deep learning framework acquired a vast amount of popularity within the scientific community upon initial release due to the simplicity of the code when designing various machine learning models, the popularity of which lead to the deep learning framework acquiring interest from prominent information technology companies such as Facebook and IBM. On the other hand, Giannini et al (2017) suggested that that the Torch deep learning framework could invoke an amount of difficulty in comparison to alternative frameworks, specifically due an issue with the framework documentation being spread across the torch GitHub repository. Consequently, it could be inferred that this could in turn provoke further issues when attempting to resolve any debugging issues when developing the CNN seed classification algorithm involved with this project, potentially causing unnecessary delays within the project developmental processes.

However, the machine learning framework that was largely followed throughout the course of this project was TensorFlow. Ertam and Aydin (2017) describe TensorFlow as an open-source software library developed by Google and is widely used by many large organisations for the purposes of providing an interface for expressing various machine learning algorithms, which as noted by Fonnegra et al (2017) was originally named DistBelief for conducting machine learning and deep neural network research. Abadi et al (2016) suggest that despite the TensorFlow interface being a work in progress library, the flexible graph-structure and dataflow representation enabled excellent performance to be achieved by the neural network. As such, Giannini et al (2017) suggested that the flexibility provided by a TensorFlow machine learning framework enabled very fast development for users from any level of expertise, which in turn ensured that the CNN developed during this project could be developed effectively and efficiently.

Additionally, Ertam and Aydin (2017) also suggested that a calculation expressed using the TensorFlow interface can in turn also be compatible for implementation with a wide range of heterogeneous systems with minimal modification required, such as CPU and GPU devices. Therefore, the widescale applicability of TensorFlow advantageously allows the deployment of numerical computation to one or more CPU or GPU devices in a wide variety of different system architectures through a single application programming interface (API). Which as a consequence, could enable flexible development with the CNN model which operated on seed sample images in this project, such as ensuring that the STRETCH objective of developing the CNN model with a GPU device enabled (see chapter 1.2) could be accomplished to further enhance the CNN classification performance.

3.3 RESEARCH METHODS

As previously discussed, the nature of this project entails the development of a superpixel convolutional neural network seed image segmentation algorithm, capable of providing the capability for extracting seed objects from seed sample images to assist botanists with a means of performing analysis of the plant species to which the seeds may belong to (see chapter 1). To this end, an issue of paramount importance which has been considered throughout this project is the consideration of the accuracy performance of the developed software solution, and the accuracy to which the algorithm is able to coherently discriminate the seed objects away from the background and any noisy pixels that may reside in the seed sample images. The quantitative assessment methodology for this is twofold; the classification performance of the CNN and the image reconstruction performance.

Table 1. Example of the expected confusion matrix structure for the developed CNN model.

	Predicted: Seed	Predicted: Background
Actual: Seed	TP	FN
Actual: Background	FP	TN

As such, quantitative data was conformed in the form of a confusion matrix, whereby the predicted values are immediately compared with the true classification value of each superpixel and subsequently providing four vital statistical values describing the classifier performance. As elaborated by Kuhn and Johnson (2013), true positives (TP) and true negatives (TN) describe the cases where the CNN model correctly predicted the superpixel classification in correlation with the true classification value, whereas false positives (FP) and false negatives (FN) describe the cases where a different classification value was predicted in comparison to the true classification value of a superpixel.

As explained by Yaram (2016), the production of this confusion matrix was used to produce a set of quantitative data which in turn could be used to provide an overview in regard to the percentage of the test data that was correctly classified by the CNN model, the quantitative data of which could be subsequently utilised in further statistical calculations to further evaluate the performance of the neural network.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100 \quad \text{(Eq. 3.1)}$$

In reference to formula 1, the classification accuracy would be calculated by identifying the total number of correct predictions, namely true positive (TP) and true negative (TN) predictions and dividing this summation by the total number of all predictions that were made; including the false positive (FP) and false negative (FN) predictions. As a result, this quantitative percentile accuracy metric was utilised as an overall performance metric which could be argued to be representative of the CNN model performance when evaluating the classification of superpixel images contained within the training, validation and test datasets.

$$Precision = \frac{TP}{TP + FP} \quad \text{(Eq. 3.2)}$$

Additionally, a further quantitative precision metric was calculated by estimating the ratio of correctly predicted positive seed classifications against the total number of predicted positive observations. Effectively, this metric was used to measure the CNN's ability to accurately predict only the relevant instances, namely whether a particular superpixel was correctly classified as a seed superpixel.

$$Recall = \frac{TP}{TP + FN} \quad \text{(Eq. 3.3)}$$

On the other hand, a recall quantitative metric was calculated by estimating the ratio of correctly predicted positive seed superpixel classifications against the total number of all of the seed superpixels that could have been labelled. This metric was utilised to assess the capability of the CNN in regard to correctly classifying all relevant instances, for example measuring the accuracy of the CNN in terms of correctly predicting a seed superpixel against all possible seed superpixel instances that reside within the ground truth mask.

An imperative characteristic to be considered in regard to the overall performance of the algorithm is whether the generated mask is justifiably representative of the true seed objects that reside within each seed sample image, thus additional quantitative data was collected for the assessment of the final image segmentation accuracy during the image reconstruction phase of the software solution.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

$$mean\ IoU = \frac{\sum_1^N IoU}{N} \qquad \text{(Eq. 3.4)}$$

This entailed a intersect over union (IoU) accuracy metric to be estimated, where the generated segmentation mask from the developed algorithm would be compared to the desired true values depicted in the ground truth mask to produce an estimate of the statistical overlay ratio, a process of which would be repeated until an IoU value has been estimated for all of the output masks predicted within the test dataset. Thus, these IoU values are subsequently summed to a total IoU value and divided by the total number of image samples contained within the test dataset, which in turn will provide an estimate on a mean IoU segmentation accuracy value. As a result, this would be used to assess the overall accuracy of the final output seed mask segmentation, where a higher overlay ratio would be indicative of a higher segmentation accuracy performance of the algorithm.

CHAPTER 4: DESIGN, DEVELOPMENT AND EVALUATION

The fundamental aim of this project was to design an automatic seed segmentation algorithm and develop a robust software solution which could be utilised by botanists to assist with the detection of various plant species to which the seeds may originate from. The software solution developed during this project was designed to combine the benefits of the superpixel segmentation technique with the benefits of a convolutional neural network (CNN), the outcome of which should provide an effective, robust software solution which is capable of automatically segmenting images of seeds to a high standard of accuracy. The successful delivery of this software solution is hoped to assist botanists by providing an effective solution which can deliver an accurate segmentation depiction of each seed efficiently.

The following chapter will follow an in-depth discussion regarding the prominent phases of the established software development lifecycle for the project, further exploring how the solution was tailored toward a satisfiable delivery outcome for the previously elaborated project aim and objectives.

4.1 DATA ACQUISITION

As previously described in a subset of the literature review (see chapter 2), Krizhevsky et al (2012) and Gheisari et al (2017) suggest that deep neural network architectures provide the capacity to perform more effectively if a larger input training dataset is learned upon by the CNN, further simulating a big-data learning environment to which Sutton and Barlow (1998) suggest could lead to a higher abstraction of knowledge being sought by the neural network. To this end, an initial small dataset comprising of approximately 161 different seed sample images were supplied, where each seed sample image contains a random assortment of seeds originating from different plant species. While this dataset could be considered as small for deep-learning classification, this size of this dataset was increased exponentially upon pre-processing techniques being employed on the dataset.

Figure 6. Images showing examples of seed samples which were collected during this project and the issues presented. Such as close proximity of seeds, noisy background and blur respectively.

However, the introduction of this seed sample image dataset also introduced a number of significant issues to be resolved during the project regarding the quality and key characteristics of the image collection. For example, in some of the images it can be observed that some of the seeds are touching or are closely positioned, thus this posed the issue of introducing a heightened level of complexity to the segmentation operation and the requirement of distinctively identifying key features which belong to each individual seed. Additionally, some of the seed samples contained a background which featured a high level of noise, the texture of which challenged the software solution of the project to accurately discriminate background noisy pixels from the seed objects contained in the image, regardless of what type of background may be featured in the image.

Another prominent issue which was identified with some of the seed samples was an element of undesirable blurring distributed around the edges of some of the seed objects within the seed sample images. As such, the blurring effect which can be observed around the edges of some of the seed samples was likely caused by the focus effect on the camera device which was utilised to capture the seed sample images, thus causing a centre focus to where the light has diverged and converged (see figure 6). As a result, the blurring imposed further difficulty toward the distinction between which pixels more likely represent the seed object as opposed to the background, further inducing the challenge of applying a highly accurate image segmentation algorithm.

4.2 DATA PRE-PROCESSING

Prior to the input data being supplied to the CNN model at the core of this project, various algorithms must first be applied to ensure effective compatibility with the input dataset to ensure a successful delivery of the software solution. Thus, the following section will elaborate on actions undertaken to ensure effective data supply for the CNN architecture that will be developed in a later phase of the software solution.

4.2.1 GROUND TRUTH MASK GENERATION

A fundamental objective of this project is to ensure that the developed algorithm has been sufficiently tested and evaluated, such that the output segmented mask of the seed objects is justifiably representative of each seed object and is arguably useful data which could be incorporated within the field of botany. Thus, a ground truth mask is manually created through a direct observation of each seed sample image, whereby the ground truth mask is used as empirical evidence which is used as a depiction of the expected perfect segmentation output. To surmise, each ground truth mask can subsequently be compared with the output mask generated by the algorithm developed in this project to provide an estimate of the overall segmentation accuracy, which as previously discussed in chapter three will be estimated based on the mean intersect over union accuracy metric.

Figure 7. Example of a perceived ground truth mask for a random seed sample image.

Pixels of the seed were manually labelled by carefully drawing the region of interest (ROI) around each seed object, where upon completion the ground truth mask is saved as a new image file and was reloaded in later stages of the project. The ground truth mask depicts a binary mask where the binary values of 1, otherwise observed as the white pixels in the mask, indicate seed pixels and the binary values of 0, otherwise observed as the black pixels in the mask, indicate irrelevant background pixels. This process is repeated until the entirety of the input dataset have a ground truth mask equivalent generated.

4.2.2 SPLITTING THE INPUT DATASET

While a sufficient dataset comprising of samples of various species of seeds was collected, training the CNN model on 100% of this dataset would in turn present difficulties with the model assessment in regard to the overall accuracy of the model. Prior to the seed sample image data being input into the superpixel CNN seed image segmentation algorithm, this data was randomly split into two datasets deriving from the original dataset. To elaborate, 80% of the original seed samples were randomly selected and stored within a training dataset, and the remaining 20% were stored within a test dataset.

The training dataset of seed sample images were fundamentally used as input for training the CNN phase of the system architecture, where the neural network will learn and maintain the model weights based on the features that can be extracted from this dataset until the optimal weights have been calculated. Furthermore, 20% of the training dataset was also subjugated into a validation dataset, whereby this subset of seed sample images was used to provide an unbiased evaluation toward the statistical fitness of the CNN as the model weights and the hyperparameters were maintained and tuned. As such, the validation dataset aided the detection of the possibility of overfitting or underfitting the CNN with the training dataset.

In contrast, the test dataset was a small dataset which was utilised for testing the final classification and segmentation output of the model which was trained on the training dataset, further confirming that similar results could be obtained on a smaller subset of seed sample images to which the CNN model had no previous knowledge about. As such, the accuracy results achieved from the test dataset was the prominent focus when assessing the overall accuracy performance of the developed system architecture.

4.2.3 SUPERPIXEL DATASET GENERATION ALGORITHM

A vital issue which was required to be solved during the preliminary stages of the project was the consideration for the input dataset that would be utilised by the convolutional neural network. As such, the core of this project revolves around concept of developing a CNN architecture which inherits and learns from a dataset comprising of small clusters of pixels which represent various regions of the seed sample images. As a result, an image segmentation technique referred to as the simple linear iterative clustering (SLIC) developed by Achanta et al (2012) was utilised to segment the seed sample images into clusters of superpixels.

The SLIC segmentation algorithm generated superpixels predominantly based on the colour intensity similarity and proximity in the spatial domain. Imperatively, the SLIC algorithm initially converts the original RGB seed sample image into a CIELAB colour space, a five dimensional '$labxy$' space which is used for the clustering operations being performed by the algorithm. The 'l,a,b' dimensions denote the pixel colour intensities and the 'x,y' dimensions denote the pixel positions within the spatial domain.

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2}$$
$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \qquad \text{(Eq. 4.1)}$$

To this end, the measure of similarity and proximity between all of the pixels depicted within the CIELAB colour space of the input is estimated through a distance metric referred to as the Euclidean distance metric (see Eq. 4.1). Effectively, this metric is used to estimate the total distance, or the total difference, between each of the pixels in an attempt to measure the similarity and proximity of all pixels contained within each input seed sample image. For example, a lower Euclidean distance value can be argued to be indicative of a similar or closely positioned pixel in comparison to a pixel which resulted in a higher Euclidean distance value.

$$d_s = d_{lab} + \frac{m}{S} d_{xy} \qquad \text{(Eq. 4.2)}$$

However, Achanta et al (2012) noted that if the pixel distances within the spatial domain exceed the colour distance limit, then this could undesirably result in superpixels which did not conform to the region boundaries and only proximity within the spatial domain of the image. Thus, the spatial proximity distances d_{xy} are normalised by a compactness value 'm' divided by grid interval 'S' to ensure a balance between the colour similarity distances and the spatial proximity distances to produce an overall distance measure d_s (see Eq. 4.2).

$$G(x, y) = ||\boldsymbol{I}(x+1, y) - \boldsymbol{I}(x-1, y)||^2 + ||\boldsymbol{I}(x, y+1) - \boldsymbol{I}(x, y-1)||^2 \qquad \text{(Eq. 4.3)}$$

The SLIC algorithm initialises by sampling a pre-defined number of regularly spaced cluster centres, in turn transferring them to seed locations determined to have the lowest gradient position (see Eq. 4.3) in a 3 x 3 neighbourhood of pixels (Achanta et al, 2012), where in turn each pixel featured within the input image is assigned to the closest cluster centre using the previously explained distance measure d_s. Upon all of the pixels being associated with the nearest cluster centre, a new centre is then estimated based on the average $labxy$ pixels for each cluster, at which point the algorithm continues iterate this same process and attempts to assign each pixel in the image to the nearest cluster centre while incorporating the newly formed cluster centres until convergence is achieved.

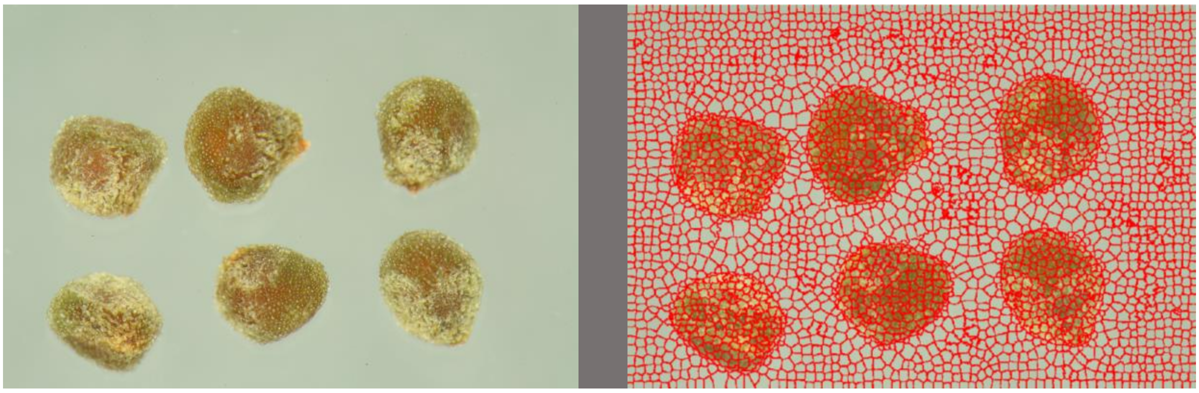

Figure 8. Example of SLIC being applied to a random seed sample image. Each region contained within the red boundaries are considered as superpixels.

The outcome of this algorithm should present a new grid of superpixels which have been generated onto a grid relative to the original input image as shown in figure 8, where each superpixel should describe a small region of similar pixels segmented on the original input image. As such, each featured superpixel will be utilised as part of the training dataset of the CNN architecture in a later phase of the solution design, where the CNN will attempt to learn any patterns that may exist throughout the entirety of the superpixel dataset that will be compiled based on all of the seed sample images.

4.2.4 GROUND TRUTH COMPARISON

A vital issue which remains is that the CNN currently possesses no logical methodology for learning the difference in patterns between different classifications of the superpixels, thus a subsequent objective of this algorithm is to be able to intelligently classify and separate each superpixel based on whether it describes seed pixels or background pixels. This is largely accomplished by comparing each generated superpixel with the ground truth mask of each input seed sample image.

The dataset generation algorithm accomplishes this task by iterating through the entirety of the superpixel dataset, using the global coordinates of each superpixel to extract and compare with the ground-truth mask that was manually established for each of the original input images. This operates such that if the calculated maximum intensity value of the ground-truth mask extracted based on the global coordinates of the superpixel is greater than zero, then it is assumed to be likely that this superpixel contains pixels which describe a seed in the original image and can subsequently categorised under the seed object classification. This algorithm should continue to iterate until the full grid of superpixels for the original input image have been considered. The outcome of this algorithm should output each superpixel that was generated as a new image file as a new dataset, classifying each superpixel into different folders depending on which classification each superpixel was determined to represent.

4.2.5 DATA AUGMENTATION

As previously discussed, the core of the software solution which is being developed in this project has been conceptualised toward combining the benefits of both superpixel image segmentation and a CNN model into a single solution. As described by Ahmad et al (2017), the learning effectiveness of deep CNNs are highly dependent on the availability of a sufficiently large training dataset, this in turn presents a prominent deep-learning issue of ensuring that the neural network model achieves a good statistical fit with the training data, avoiding further complications such as overfitting. Thus, a strategy undertaken to reduce the issue of overfitting was through a data augmentation technique, whereby the training dataset would be accommodated toward generalising the model with the intent of achieving a good statistical fit.

Statistically overfitting the CNN model refers to the model being undesirably synonymous with the derived dataset, as elaborated by Burnham and Anderson (2002), the essence of overfitting can be described as unknowingly extracting residual variation as though that variation was representative of the underlying model structure. Therefore, this can lead to complications in later phases of the software solution, such as failure in fitting the developed model with additional data and the inability to accurately classify superpixels derived from new seed sample images caused by a high model variance. In contrast, the desirable CNN model to be developed within this project is an accurate model which achieves a statistically good fit, whereby the model is comprised of a sufficient amount of bias and variance to ensure the model capability for seed superpixel classification.

One approach toward reducing overfitting the developed CNN model is by expanding the current training dataset, a task of which can be achieved through data augmentation prior to the training dataset being input into the neural network. Data augmentation is utilised to effectively apply various transformation and deformations to the labelled dataset, the output of which provides new image samples which in turn can be used as additional training data for the CNN model involved in this project. Specifically, the original training dataset was augmented by applying different image transformation techniques such as scale, rotation, translation, zoom and horizontal flipping to create five unique variations of each seed sample superpixel image contained in the original training dataset as demonstrated in figure 9.

Figure 9. Example of the augmented dataset, where the leftmost superpixel is the original superpixel and the remaining superpixels are the augmented five variants.

Advantageously, the subtle discrepancies created for each of the five variants of the original training dataset in turn increases the total size of the dataset by approximately five times of the original size, thus improving the model generalisation and reduced the issue of statistically overfitting with the training dataset.

4.2.6 DATA NORMALISATION

Prior to the input data being loaded and analysed in later phases of the system architecture, it can be observed that a prominent issue which arises is the high characteristic variability that exists within the superpixel image dataset. The input dataset that is being considered in this project is fundamentally an extremely large dataset of superpixels depicted from seed sample images, this in turn introduces the issue of a difference in data image size caused by the contrasting sizes of the generated superpixels. For example, a superpixel describing a segment of a seed could be of an image size of 20 x 12 x 3, whereas a different superpixel could be found to be of an image size of 30 x 24 x 3. Difference in image sizes can cause fundamental issues toward model training compatibility with the provided dataset, thus normalisation techniques were applied on each batch of seed sample superpixel images that were input into the CNN model.

Furthermore, the process of normalisation follows and downscaling or upscaling each input image where required until the new size of the input image converges with a pre-set size value, a value of which is applied consistently amongst all of the input seed sample superpixel images. For example, the aforementioned superpixel image of size 20 x 12 x 3 would be normalised to an image size of 20 x 20 x 3, similarly a different superpixel of image size 30 x 24 x 3 would be normalised to an image size of 20 x 20 x 3. The outcome accomplished through the normalisation technique is the delivery of an input dataset which is of a consistent size of 20 x 20 x 3, a dataset of which is subsequently compatible for implementation into the developed CNN architecture and can be input into the next phase of the software solution.

However, while image distortion could cause an amount of feature loss, Wang et al (2017) argued that superpixel segmentation supplements the advantage of ensuring that the superpixels are generated at a more relative segmentation size distribution in comparison to rivalling methodologies. Therefore, the appropriation of a normalised dataset based on a superpixel generation dataset is likely to require minimal image distortion in comparison to rivalling methodologies, which in turn ensures that effective features can be extracted and utilised within the next phase of the software solution. To surmise, the input images contained within all three datasets were resized to a normalised image size of 20 x 20 x 3 through bilinear interpolation. This new image size was considered as a sufficient new image size through careful observation of the generated superpixel datasets, this was determined as a normalised size which would cause minimal distortion to the majority of the superpixel images as this was a new image size which wasn't drastically dissimilar to the original superpixel dataset.

4.3 CONVOLUTIONAL NEURAL NETWORK

As previously discussed, the desired outcome from the successful completion of this project was to deliver a software solution which was robustly capable of providing an image segmentation methodology using a deep-learning framework, which in turn could be advantageously utilised by various Botanists for the identification of plant species in future systems (see Chapter 1). As such, the core of the developed system architecture follows the advantages of deep-learning framework referred to as a convolutional neural network (CNN). Furthermore, a vital objective which the CNN approach adhered to was to establish an approach toward identifying and learning any classification patterns or correlations that may exist among a large dataset of seed sample images, the knowledge of which could be utilised for determining the classifications of objects in future seed sample images. For example, the desired outcome for the CNN model developed in this project was to learn and recognise characteristics from a subset of seed sample superpixel images and correctly classify them into a total of two different classifications, namely either a seed or background superpixel classification.

A CNN model can be described as being a supervised neural network deep learning model, which facilitates object recognition by being rigorously trained on a large training dataset containing sets of labelled seed sample images. As such, each labelled image can be considered as being incorporated within the initial input layer during the initialisation stage of the CNN, where small batches of the images will be selected and input into a series of different hidden layers featured as part of the model.

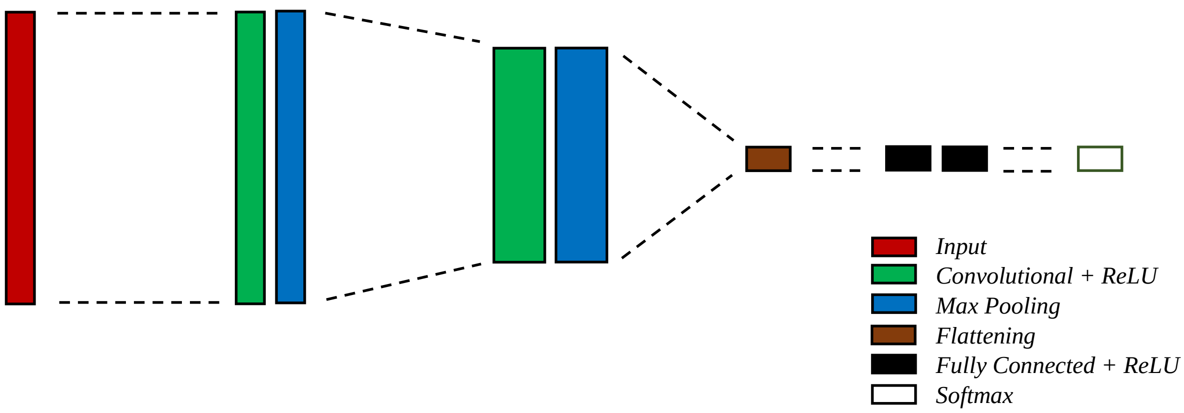

Figure 10. Superpixel CNN segmentation architecture implemented during this project.

Fundamentally, the CNN involved in this project will take batches of superpixel images which have been extracted from various regions of the original seed sample image as input, where each batch of images will be continuously propagated through a total of nine different layers contained within the neural network during the training phase which can be observed in figure 10. As such, backward propagation is used, where the model initialises with a random set of parameters which are slowly adapted each time the correct output is achieved during the training phase of the CNN.

The following subsections will entail a discussion regarding what processes are employed during each stage of the developed CNN architecture and how it influenced the generated superpixels derived from the original seed sample images.

4.3.1 CONVOLUTION LAYER

To this end, the preliminary stages of the hidden layers feature a series of two different convolution layers. These convolution layers largely operate by distributing a set of filters, where each filter convolves around a small subset of pixels extracted from the input image though the application of linear and matrix multiplication calculations. The filter in question, otherwise referred to as a neuron or kernel, comprise of an array of numeric values which describe the values of the weights, the values of which are multiplied against the pixel intensity values contained within the current subset of pixels extracted from the original input image through element-wise multiplication.

Subsequently, the expected output from this operation should be a scalar value which represents the sum of the element-wise multiplications applied throughout the full depth of the input image, which in the case of the images being handled during this project should be a depth of dimensions, representative of the three colour channels featured within the original RGB seed sample images. As such, the convolution layer continues to apply this functionality by sliding the filter to the next subset of pixels extracted from the original input image under a pre-set stride value, an operation of which is continuously applied until all possible subsets derived from the original input image have been considered. Upon completion, the outcome of the convolution layer presents an activation map, otherwise referred to as a feature map, containing an array of values derived as a result of the repeated convolution of the filters with the original input image.

As previously discussed, the primary objective which is trying to be accomplished through the application of the convolution layer is for the establishment of feature identification for any important features which may exist within the input image, for example a stronger output signal would be signalled within the activation map if a particular subset of pixels corresponds with the filter. To this end, the activation map output from the convolution layer can be considered as being indicative toward the detection of any high-level features that may exist in local-subset segments of the input image. Additionally, the detection of features within a similar area of the completed activation map can also be indicative of a more complex feature existing.

As previously mentioned, the CNN architecture implemented within this project incorporates a total of two consecutive convolution layers. The first convolutional layer features a total of eight filters of size 5 x 5 x 3, whereas the second convolutional layer feature a total of sixteen filters of size 3 x 3 x 3.

4.3.2 ACTIVATION FUNCTION

Furthermore, an activation function, otherwise referred to as a nonlinear layer, is then subsequently incorporated following the completed application of the convolution layer. The activation function largely features an activation function which determines the appropriate final value for each filter or neuron that were considered, introducing nonlinearity to the previous linear operations that were performed as part of the convolution layer.

$$f(x) = \max(0, x) \qquad \text{(Eq. 4.4)}$$

To this end, a rectified linear unit (ReLU) activation function was developed and implemented, this effectively operates by iterating through the entirety of the existing activation map and determines whether a particular filter output or neuron should be activated: namely ensuring that the activation value results to a value between 0 and 1. This task is performed by iterating through all recorded neurons contained within the newly defined activation map and applies the activation function (see Eq. 4.4), where the ReLU activation function sets all of the negative activations to 0.

4.3.3 POOLING LAYER

However, while convolution and activation layers were successfully implemented into the design of the CNN, a prominent issue that continues to challenge the efficiency and robustness of the developed CNN is the control of overfitting the model to the training dataset which the model will be inheriting from. As such, a pooling layer is immediately applied following the application of the previous convolution and activation layers. The pooling layer operates by applying a new small filter with a pre-set stride value of two across the entirety of the input image, effectively applying down-sampling the input image by spatially resizing the image across all of the possible colour channels.

In regard to the pooling layer design that was developed for the neural network involved for this project, a max-pooling function was implemented. As such, small pooling filters of size 2 x 2 x 3 would translate between different pixels of the input seed sample images on a stride value of 2, where the maximum or highest pixel intensity within the filter window will be selected as the new pixel intensity value when the image is down-sampled in preparation for the following layer in the CNN. This operation would continue to be applied until the max-pooling filters have been applied throughout the entirety of the original input image, the output of which should be a resized or down-sampled version of the original input image.

$$W_2 = \frac{(W_1 - F)}{S + 1}$$

$$H_2 = \frac{(H_1 - F)}{S + 1}$$

$$D_2 = D_1 \qquad \text{(Eq. 4.5)}$$

The new width size of the layer after applying the pooling layer can be estimated by subtracting the filter size of two away from the total width of the previous layer, which is then divided by the total stride plus one. Similarly, the new height can be estimated by subtracting the filter size of two away from the total height of the previous layer, which is subsequently divided by the total stride plus one. However, the number of channels would remain unchanged, as the pooling layer ultimately aims to shrink the spatial domain of the superpixel image (see Eq. 4.5). For example, the first pooling layer would be applied after the convolutional layer of size 20 x 20 x 3, therefore the pooling layer would effectively shrink this layer size down to a size of 10 x 10 x 3, thus a 50% reduction in image size.

4.3.4 FULLY-CONNECTED LAYER

While the previously applied convolution layers offer the capability toward learning feature identification and extraction by relying on the local spatial coherence that exists within a small receptive field of the input image, a prominent issue which continues to present itself is the methodology to which the CNN should classify each input image based on the previously discussed feature extraction phase of the model. Fundamentally, this is accomplished within the CNN through the employment of a fully-connected layer during the concluding layers of the neural network architecture.

Specifically, the fully connected layer accepts a batch of input images, whereby each batch of input images are multiplied against the trainable weights with the addition of an amount of bias. For example, the size of the image derived from the previous convolution layers could amount to roughly 20 x 20 x 3, thus this then transforms into 1 x (20 x 20 x 3) otherwise represented as 1 x 1200 through the flattening layer.

In contrast to the convolution layers, the fully connected layers operate by utilising all possible connections that may be present within the image as opposed to operating under a local limitation and operating on only a small segment of the input image. As such, a flattening layer is employed by combining all of the previously calculated local feature maps and reduces the dimensionality output of the previous layer to a single dimension, removing any previous spatial reasoning that may have been employed in previous layers. To surmise, the output presented from this layer is a single dimension array of values which contain all possible combination of neuron values that were previously obtained during the convolution layers of the CNN, the output of which can then be considered as input for the following fully-connected layers of the neural network.

Two fully-connected layers are utilised to learn features based on the full combination of feature maps that were created as a result of the previous layers. To this end, the fully-connected layer accomplishes high level reasoning within the CNN by considering all possible neurons from each activation map defined in the previous layers, and fully connects each neuron to all of the other known neurons that exist within the neural network. Advantageously, the fully-connected layer utilises this knowledge to tune the weight parameters and determine an appropriate class probability distribution value based on the activation maps derived from the previous layers.

4.3.5 SOFTMAX OUTPUT LAYER

However, the final step involved with the developed CNN is the fundamental classification prediction phase. To this end, the output vector of the final fully connected layer is input into a softmax classification layer within the final output layer, whereby the softmax algorithm ultimately determines the probabilities of a superpixel belonging to either a seed or a background classification between a probabilistic range of real values of 0 to 1. As a result, a new vector of probabilistic values was created as the same size as the output vector supplied by the fully-connected layer, where the sum of the new probabilistic values for each class should sum to 1.

4.3.6 MODEL TRAINING

The process of training the CNN model with the input training dataset of seed sample images encompasses continuously learning through batches of samples which are propagated through the previously discussed neural network layer structure through a series of iterations during each epoch. However, the core of the system architecture revolves around the CNN's capability to learn from each batch of samples, using this knowledge to maintain a learning rate for each trainable weight and adapt them for the problem to be solved by the CNN model to a high accuracy.

As previously mentioned, the CNN propagates the model weights through a technique referred to as backward propagation, where the neural network effectively starts with randomly initialised parameters and continuously adapts the model weights to ensure that the correct classification output is achieved while training the model. For example, if a seed superpixel is mistakenly classified as a background superpixel with a probability of 0.7, the CNN model slowly adapts the parameters such that the probability of this particular superpixel being classified as a seed is increased during the next iteration of training.

The adaptive moment estimation optimisation algorithm, otherwise referred to as the Adam optimisation algorithm, was implemented within the CNN architecture. To this end, this optimisation technique was applied to minimise the loss function between a set of weights and in turn ensuring that the quantification of high quality weights can be modelled, the process of which was consistently applied during each iteration of input batches until a sufficient number of epochs had passed and the model was in turn deemed as sufficiently trained.

$$Epoch = \frac{total\ number\ of\ images}{batch\ \ size} \qquad \text{(Eq. 4.6)}$$

Thus, this training process continues to iterate until all batches of images have been processed during the first epoch of the training phase of the model as shown in equation 4.6, at which point the training phase proceeds to the second epoch where same training process is applied once more. For example, the training dataset is comprised of a total of approximately 303,457 unique superpixel images and is input into the CNN through a batch size of 64 images, therefore one epoch would equate to approximately 4,804 iterations to fully process the training dataset. Furthermore, additional training epochs for the CNN may have resulted in overfitting the model to the training data by training the model for an unnecessary number of epochs, thus through careful observation of the validation accuracy and error the model continued to train for a total of 15 epochs.

4.4 IMAGE RECONSTRUCTION

While the model may have been trained sufficiently with the training dataset, a prominent issue which remains to be resolved is how the model can be applied to a new dataset of images for the purpose of applying image segmentation and attempt to extract the seed foreground objects any new seed sample images. As previously discussed, the CNN phase of the developed software solution demonstrated the capability for evaluating the superpixel classification for any new seed sample images that may be passed into it, thus by extension demonstrating the capability to identify whether a superpixel could contain pixels that describe a seed within a new seed sample image.

Therefore, the final phase of the software solution applies the previously discussed superpixel image segmentation algorithm on a new seed sample image which has not been included in the superpixel dataset that was input into the CNN architecture, and by extension a new seed sample image to which the CNN has no prior knowledge. Consequently, any patterns that exist throughout the entirety of the superpixel dataset which were learned by the CNN architecture can be utilised to solve the issue of deciding which of the superpixels generated from the new seed sample image should be incorporated within the final output image.

For the superpixel convolutional neural network image segmentation algorithm to be applied on a new seed sample image, new superpixels would be calculated and generated based on the new seed sample image, while utilising the previously discussed superpixel segmentation algorithm. The outcome of this presents a new dataset comprising of superpixels which can be considered as candidate superpixels, where each of the newly generated superpixels are normalised to the same state as the superpixels which the CNN model was trained upon.

Thus, the developed CNN model would be reloaded and subsequently utilised when iterating through each new superpixel contained in the new seed sample image, where the neural network can subsequently evaluate each superpixel and determine which classification the superpixel is the most likely to represent, namely seed or the background, based on the knowledge to which the model has learned and trained with. As previously discussed, only superpixels which justifiably represent seed object pixels in the new seed sample images should be considered and featured when reconstructing the new output image, which in turn segments the foreground seed objects away from the background. As a result, if the trained CNN believes a superpixel most likely represents a seed superpixel, then this superpixel copied into the new output image matrix referencing the global spatial pixel coordinates of the original image to ensure that the superpixel is copied into its original spatial pixel coordinates. In contrast, if the trained CNN believes a superpixel most likely represents a background superpixel, then this result is immediately discarded and refused for incorporation into the final output image matrix.

In conclusion, the final output from this algorithm should be a distinct and accurate mask which justifiably represents a full segmentation of any identified seed objects that may be contained within the input image, whereas the remaining pixels are set to an intensity value of 0 and ignored.

4.5 EVALUATION

The following section will be exploring the final phases of the project, encompassing the methodology toward assessing the effectiveness and efficiency of the developed software solution and provide an analytical overview of the overall performance of the superpixel convolutional neural network seed segmentation algorithm relative to accomplishing the previously elaborated aim of this project (see chapter 1).

As previously elaborated in chapter 3, the fundamental component which was assessed during this project was the superpixel convolutional neural network classification accuracy performance and its capability toward learning features of seeds based on the generated superpixels and use this knowledge to classify future superpixels that may be generated based on new seed sample images. As such, the trained superpixel convolutional neural network algorithm attempted to perform a series of superpixel classifications based on the test dataset containing different seed sample images, a dataset which the trained superpixel convolutional neural network has no knowledge about.

Table 2. Confusion matrix output when applying the developed CNN model on the test dataset.

	Predicted: Seed	Predicted: Background
Actual: Seed	30,821	2,636
Actual: Background	489	41,110

To surmise, the trained superpixel CNN performance was assessed against a total of 75,056 unique superpixel images contained within the test dataset. Overall, the confusion matrix demonstrated promising results regarding the ratio of the TP and TN values, 30,821 and 41,110 respectively, which in turn inferred a positive performance toward evaluating the classification value of a particular superpixel in alignment to the actual value.

On the other hand, it can also be observed that a critical result obtained was the FN value of 2,636, which in turn portrayed that a notably high proportion of the background superpixels were incorrectly classified as a seed superpixel. As such, this proportion of misclassifications could in turn impose an impact during the image reconstruction phase of the system architecture, as these misclassifications could cause undesirable superpixels depicting segments of the background to be incorporated in the new reconstructed version of the input seed sample image.

Furthermore, these results were then input into a subsequent classification accuracy formula, whereby an accuracy metric was calculated by evaluating the correct classifications against all classifications that were made by the CNN architecture.

$$Accuracy = \frac{30821 + 41110}{30821 + 41110 + 489 + 2636} * 100 \qquad \text{(Eq. 4.7)}$$

$$Precision = \frac{30821}{30821 + 489} \qquad \text{(Eq. 4.8)}$$

$$Recall = \frac{30821}{30821 + 2636} \qquad \text{(Eq. 4.9)}$$

Overall, the superpixel convolutional neural network seed image segmentation algorithm achieved an impressive classification accuracy value of approximately 95.84% accuracy when evaluating classification values for the test dataset involved within this project. Additionally, the algorithm obtained a precision score of 98.43% and a recall score of 92.12%.

However, to further assess the software solution developed during this project, further analysis was employed to evaluate the quality of the final segmentation applied when the algorithm attempted to effectively extract any seed objects that may exist within each seed sample image. To this end, a high-quality segmentation is expected relative to the high classification accuracy that was previously achieved with the CNN architecture of the software solution, such that each seed object is cleanly extracted while distinctly excluding any other superpixels that may exist within the image.

Figure 11. Approximate example of how the bounding boxes for the predicted output mask and the ground truth mask respectively would be compared when calculating the IoU.

Additionally, as previously explained in chapter 3, the segmentation accuracy of the developed algorithm was further analysed through an estimation of the mean intersect over union (IoU) accuracy metric, which compared the output predicted seed mask with the ground truth, thereby evaluating the overlap ratio between the two masks to estimate the accuracy of the final segmentation. This is done by comparing two bounding box outputs which extract the mask output from both the predicted mask and the ground truth mask, where the overlay ratio estimated through IoU will signify how accurate the output segmentation was to the expected output.

This segmentation accuracy metric yielded an underwhelming mean IoU accuracy score of 73.18% with the test dataset, however it should be noted that this mean IoU accuracy score continues to incorporate a small number of background superpixels around the borders of some of the seed samples which will greatly reduce the mean IoU accuracy value. For example, figure 11 shows how the unnecessary background superpixels near the top border of the predicted mask will affect the IoU accuracy calculation due to a large distortion with the predicted mask bounding box.

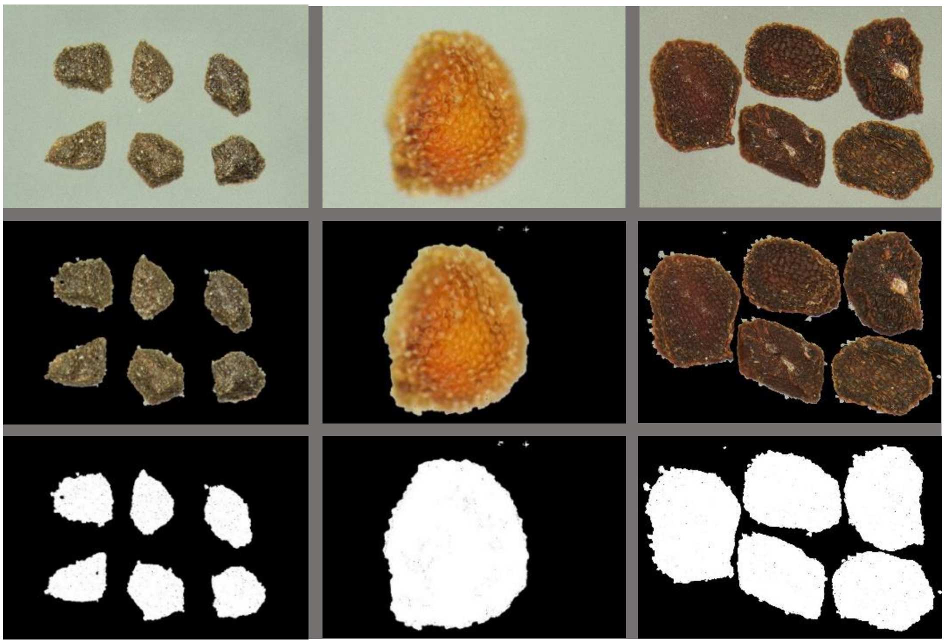

Figure 12. A comparison between three different outputs when the superpixel CNN seed image segmentation algorithm is applied on three different seed image samples.

Overall, it can be observed that while in most cases the developed seed is clearly extracted from the image and distinct from any previously surrounding background superpixels, in some instances remnants of undesired background superpixels can be observed to be scattered amongst the final output mask as shown in figure 12 and appendix A. Thus, it can be argued that the developed software solution is far from perfect and is clearly still susceptible to the inclusion of undesirable superpixels representative of the background pixels, for example the middle seed sample example in the figure above features a cluster of undesirable superpixels near the upper image border.

Furthermore, it could be argued that the inclusion of these unnecessary superpixels in the final output image could be the direct consequence of the previous observations made from the confusion matrix approximation on the test dataset, where a FN value of 2,636 was collected which depicted the ratio of superpixels which were incorrectly classified as a seed superpixel. Thus, it could be argued that this caused the CNN model to clearly interpret some of the background superpixels incorrectly and classify them as seed superpixels, consequently causing the algorithm to incorporate these false superpixels in the final output image.

Despite this, it can also be argued that this project yielded a positive outcome, demonstrating an accurate image segmentation algorithm which is clearly representative of the seed objects contained in each seed sample image. Furthermore, as previously elaborated in chapter 4.1, one of the acknowledged challenges presented with the utilisation of the seed sample datasets that were collected was the fundamental issue of how the algorithm would interpret seeds that are either touching or are very closely positioned. To this end, the algorithm can be observed to advantageously grasp the benefits of the superpixel segmentation sensitivity, whereby the seed superpixels can be observed to cleanly segment around the small space between the two seeds in the seed sample example.

CHAPTER 5: PROJECT CONCLUSION

The following chapter will encompass a final analytical discussion in regard to the overall findings of the project, providing further details regarding the correlation of the final algorithmic performance in relation to the initially established aim and objectives of this project.

As previously elaborated, the fundamental aim of this project entailed the desire to develop a seed segmentation and feature extraction algorithm which utilised deep-learning strategies for the purposes of performing accurate feature extraction capable of identifying core characteristics of seeds originating from various plant species (see chapter 1), and in turn using this knowledge to discriminate the seeds away from the background and other objects which may exist in the image.

As such, one of the fundamental objectives involved to ensure the accomplishment of this project aim was the development of a deep convolutional neural network architecture which had the capacity to extract key features from a given set of input images which could be used to learn key characteristics of various seed species, which in turn would provide a means for accurate classification of any seed objects within a particular seed sample image and provide a means for extracting the seed objects. As such, superpixel segmentation clearly provided effective segmentation results, the superpixel regions of which were then input into the CNN phase of the system architecture. To this end, the developed CNN model showed clear capacity for recognising key features of various seed species across all datasets that were handled during this project, demonstrating a classification accuracy of approximately 95.84% when operating on the test dataset, thus showing that the trained CNN model was capable of correctly recognising 95.84% superpixels out of all of the superpixels that were processed within a dataset of images which the trained CNN model has no knowledge about.

Overall, these results were indicative of a positive and robust CNN performance which would be capable of classifying superpixels that could be generated based on any new seed sample images that could be provided and analysed by various botanists. However, while features were successfully classified and extracted from the seed sample images, the aim of this project also advocated the successful application of accurate image segmentation capable of accurately and robustly extracting any seed objects that exist throughout the seed sample images. The superpixel CNN image segmentation algorithm demonstrated a satisfactory outcome toward providing botanists with an effective mask which could be used to possibly calculate any subsequent photometric analytical operations, as the seed objects are clearly extracted with minimal data loss as a direct consequence of the sufficient performance that was achieved with the CNN phase of the system. Though further improvements are evident, as the final image output from the algorithm continues to include background superpixels, which in turn could potentially negatively impact any subsequent analytical operations which could be applied to the segmented image.

Furthermore, while the mean IoU segmentation accuracy metric of 73.18% was indicative of a less impressive performance compared to the classification accuracy of the CNN model, further post-processing enhancement techniques could be utilised to help remove excess background pixels that were incorporated in the final segmentation. Specifically, an issue observed with the final segmentation outputs is that a small number of background superpixels were mistakenly identified as seed superpixels and incorporated around the border of the output mask. This heavily influences the final mean IoU accuracy metric, as the bounding box encompassing the predicted mask would technically be much larger than the ground truth mask and in turn demonstrated a lower IoU performance. Further refinement to the CNN model and the application of post-processing techniques such as morphological operations could help to minimise this issue and maximise the overall segmentation accuracy.

To surmise, the trained superpixel CNN image segmentation algorithm developed at the outcome of this project can be concluded as providing an effective means toward providing accurate seed extraction capabilities through the utilisation of deep-learning strategies, thus this trained algorithm could effectively be applied to any given dataset of new seed sample images supplemented by various botanists and perform accurate image segmentation.

However, further research could be employed into research individual seed species classification and segmentation through the combination of superpixel segmentation and deep convolutional neural networks. This could help various botanists with an automatic assessment of seed species as soon as new seed sample data is input into the superpixel CNN algorithm, however possible challenges that could arise with this alternative approach is the requirement of more data. While the data collected during this project was sufficient for accomplishing the predefined aim and objectives of this project, more examples of each individual seed species would likely be required as a pre-requisite to ensure that a high classification accuracy can still be achieved despite the increase in the number of possible classes.

CHAPTER 6: REFLECTIVE ANALYSIS

The following chapter will encompass an analytical discussion regarding the efficacy of the methodological processes that were employed throughout the course of this project, exploring the impact of significant decisions against their desired impact.

6.1 PROJECT EXECUTION

To surmise, the execution of this project can be argued to be successful when the completion of the majority of the initial objectives derived from the project aim are considered, thus the superpixel CNN image segmentation algorithm is capable of generating an accurate mask depiction of the seed objects that may exist in a particular seed sample image. However, it can be argued that the difficulty of the project was significantly underestimated, particularly as the lack of prior experience and knowledge within the field of machine learning imposed an arduous task during development which in turn required a higher than anticipated amount of project resources. For example, the task of preparing the input dataset for sufficient compatibility with the developed CNN model architecture imposed unexpected generalisation issues, which lead to an overall weaker performance upon applying the reconstruction phase of the CNN on a sample from the test dataset of seed samples and demonstrated a poorer segmentation output as a result. Fortunately, this particular issue was amended through the application of the previously elaborated data augmentation technique, which greatly expanded the original training dataset and in turn improved the generalisation of the model.

On the other hand, in consideration of the fact that no prior knowledge in the field of machine learning or deep learning was acquired before undertaking this project, it can be argued that a wealth of knowledge has been learned during the short experiences involved within this project. Implementing a segmentation technique with a CNN architecture has allowed observations into the many benefits which can be gained through deep-learning strategies, for example being able to obtain a high segmentation accuracy score on unknown seed sample images using the knowledge regarding image features learned from a different selection of seed sample images.

6.2 PROJECT MANAGEMENT METHODOLOGY

As previously discussed, the project management methodology followed throughout the course of this project was an approach which was largely inspired by the fundamental principles defined within an agile paradigm, whereby the project was managed under a series of small iterative software development lifecycles until a satisfying outcome is achieved with the superpixel convolutional neural network seed image segmentation algorithm (see chapter 3). While it can be argued that this project management paradigm enabled the development of a theoretically higher quality segmentation algorithm due to the copious amount of testing that was employed on core components of the project, it can also be observed as an acknowledged flaw when concerned with the strategies employed. For example, project resources were consistently appointed toward the development of the CNN architecture to ensure a highly accurate software solution was developed at the outcome of the project. To summarise, it can be argued that this project may have benefitted from a different project management methodology as a strict deadline was enforced, for example a project management which prioritised development speed as opposed to component quality assurance such as a waterfall paradigm inspired methodology.

6.3 UTILISATION OF A GPU DEVICE WITH THE CNN

Despite the success of this project, it can be observed that the elapsed time to initially train the superpixel CNN model was significantly high, though it should be noted that a significantly low amount of elapsed of time is required for evaluating superpixels and utilising the trained model for image segmentation. However, if a surplus of project resources were available, further investigation would be employed to develop the superpixel CNN seed image segmentation algorithm with compatibility with a GPU device. As such, the potential benefits that could be reaped from this are twofold; a reduction in the elapsed time for fully training the CNN model and an increase in the potential neural network capacity.

At the conclusion of this project, the software solution was developed under a serial system architecture through the utilisation of just a central processing unit (CPU) device, however as suggested previously by Liu et al (2012) the training time complexity performance of the developed CNN architecture could theoretically be exponentially improved by integrating operations with an additional GPU device. As such, the CNN model developed during this project could beneficially delegate operations between the different devices and enable a parallel system architecture, whereby hidden layer operations could be completed in synchronisation between the CPU and GPU devices.

Furthermore, previous research conducted by Krizhevsky et al (2012) indicated that the utilisation of a single GPU device limits the total size of the neural network, whereas operating through an integration of multiple GPU devices could greatly expand the neural network capacity that can be utilised. Thus, the CNN architecture developed during this project could benefit from this strategy by enabling parallel hidden layer capabilities by splitting the hidden layer operations between multiple GPU devices, increasing the neural network capacity depending on the number of GPU devices which are incorporated. This in turn would advantageously ensure that a larger dataset of seed sample images could be trained upon by the neural network, the impact of which would theoretically simulate larger learning environment for the CNN model to which more knowledge regarding the seed features can be abstracted.

6.4 DATASET COLLECTION

Similarly, if more time was available, additional project resources would be delegated toward the acquisition of a larger initial seed sample image dataset, which would contain more samples of both existing seed species and new seed species which can be input in the CNN model. Previous research conducted by Gheisari et al (2017) suggested deep-learning strategies are effective when attempting analyse and learn useful knowledge from large amounts data as well as data collected from a variety of different sources, which is also supported by the research conducted by Krizhevsky et al (2012) which demonstrated a performance increase as a result of their neural network being trained with a larger dataset of images through the utilisation of two Nvidia GeForce GTX 580 GPU devices. To surmise, while it can be argued that a sufficient dataset was acquired for training the CNN model involved during this project after applying a superpixel dataset generation algorithm, further increasing the initial seed sample dataset with more examples would further simulate a big data learning environment to which the CNN model could adhere to, thereby also theoretically improving the classification accuracy performance of the developed CNN model.

REFERENCE LIST

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Sard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X. (2016) TensorFlow: A system for large-scale machine learning. In: *12th USENIX Symposium on Operating Systems Design and Implementation*, Savannah, Georgia, United States of America, 2-4 November. California, United States of America: USENIX. Available from: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf [accessed: 07-04-2018].

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S. (2012) SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11) 2274-2281. Available from: http://www.ieeexplore.ieee.org/document/6205760/ [accessed: 17-03-2018].

Agatonovic-Kustrin, S., Beresford, R. (2000) Basic Concepts of Artificial Neural Network (ANN) Modelling and its Application in Pharmaceutical Research. *Journal of Pharmaceutical and Biomedical Analysis*, 22(5) 717-727. Available from: http://www.sciencedirect.com/science/article/pii/S0731708599002721 [accessed: 16-10-2017].

Ahmad, J., Muhammad K., Baik, S.W. (2017) Data augmentation-assisted deep learning of hand-drawn partially colored sketches for visual search. *Public Library of Science (PLOS) One*, 12(8) 1-19. Available from: http://www.journals.plos.org/plosone/article?id=10.1371/journal.pone.0183838 [accessed: 21-03-2018].

Barré, P., Stöver, B.C., Müller, K.F., Steinhage, V. (2017) Leafnet: A Computer Vision System for Automatic Plant Species Identification. *Elsevier Ecological Informatics*, 40, 50-56. Available from: http://www.sciencedirect.com/science/article/pii/S1574954116302515 [accessed: 10-10-2017].

Burnham, K.P., Anderson, D.R. (2002) *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*, 2nd edition. New York: Springer-Verlag. [accessed: 26-03-2018].

Chitra, S.H.H., Suguna, S. (2016) Comparative Study on Image Analysis Techniques for Plant Seed Images. In: *2nd IEEE International Conference on Engineering and Technology (ICETECH)*, Coimbatore, India, 17-18 March. United States of America: IEEE, 1-7. Available from: http://www.ieeexplore.ieee.org.proxy.library.lincoln.ac.uk/document/7569367/ [accessed: 18-10-2017].

Cockburn, A. and Highsmith, J. (2007) *Agile Software Development: The Cooperative Gam*e, 2nd edition. United States of America: Pearson Education, Inc.

Frossard, D. (2016) *VGG in TensorFlow. Model and pre-trained parameters for VGG16 in TensorFlow*. Available from: http://www.cs.toronto.edu/~frossard/post/vgg16/ [accessed: 16-04-2018].

École Polytechnique Fédérale de Lausanne (2017) *SLIC Superpixels*. Available from: https://www.ivrl.epfl.ch/research/superpixels [accessed: 11-04-2018].

Egmont-Petersen, M., de Ridder, D., Handels, H. (2002) Image Processing with Neural Networks – A Review. *Pattern Recognition*, 35(10) 2279-2301. Available from: http://www.sciencedirect.com/science/article/pii/S0031320301001789 [accessed: 14-10-2017].

Ertam, F., Aydin, G. (2017) Data classification with deep learning using TensorFlow. In: *2017 International Conference on Computer Science and Engineering (UBMK)*, Antalya, Turkey, 5-8 October 2017. United States of America: IEEE, 755-759. Available from: https://www.ieeexplore-ieee-org.proxy.library.lincoln.ac.uk/document/8093521/ [accessed: 07-04-2018].

Fonnegra, R.D., Blair, B., Díaz, G.M. (2017) Performance comparison of deep learning frameworks in image classification problems using convolutional and recurrent networks. In: *2017 IEEE Colombian Conference on Communications and Computing (COLCOM)*, Cartagena, Colombia, 16-18 August. New York: United States of America: IEEE, 1-6. Available from: https://www.ieeexplore-ieee-org.proxy.library.lincoln.ac.uk/document/8088219/ [accessed: 08-04-2018].

Ghazi, M.M., Yanikoglu, B., Aptoula, E. (2017) Plant identification using deep neural networks via optimization of transfer learning parameters. *Elsevier Neurocomputing*, 235 228-235. Available from: https://www.sciencedirect.com/science/article/pii/S0925231217300498 [accessed: 29-03-2018].

Gheisari, M., Wang, G., Zakirul, M., Bhuiyan, A. (2017) A Survey on Deep Learning in Big Data. In: *IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, Guangzhou, China, 21-24 July. New York: United States of America: IEEE, 173-180. Available from: http://www.ieeexplore.ieee.org/abstract/document/8005992/ [accessed: 17-03-2018].

Giannini, F., Laveglia, V., Rossi, A., Zanca, D., Zugarini, A. (2017) Neural networks for beginners. A fast implementation in Matlab, Torch, TensorFlow [pre-print]. Available from: https://www.arxiv.org/abs/1703.05298 [accessed: 05-04-2018].

Han, Z., Zhao, Y. (2009) A Method of Detecting Peanut Cultivars and Quality Based on the Appearance Characteristic Recognition. *Information and Computing Science*, 2, 21-24. Available from: http://www.ieeexplore.ieee.org.proxy.library.lincoln.ac.uk/document/5168997/ [accessed: 12-10-2017].

Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012) ImageNet Classification with Deep Convolutional Neural Networks. In: *26th Annual Conference on Neural Information Processing Systems*, Nevada, United States of America, 3-6 December. United States of America: NIPS, 1-9. Available from: https://www.papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf [accessed: 12-10-2017].

Kuhn, M., Johnson, K. (2013) *Applied Predictive Modeling*, 1st edition. United States of America: Springer Publishing.

Lakshman, M., Sinha, L., Biswas, M., Charles, M., Arora, N.K. (2000) Quantitative Vs qualitative research methods. *The Indian Journal of Pediatrics*, 67(5) 369-377. Available from: https://www.link-springer-com.proxy.library.lincoln.ac.uk/article/10.1007%2FBF02820690 [accessed: 10-04-2018].

Liu, Y., Xiao, Y., Wang, L., Pan, J., Yan, Y. (2012) Parallel Implementation of Neural Networks Training on Graphic Processing Unit. In: *5th IEEE Conference on BioMedical Engineering and Informatics*, Chongqing, China, 16-18 October. United States of America: IEEE, 1-4. Available from: http://www.ieeexplore.ieee.org.proxy.library.lincoln.ac.uk/document/6513078/media [accessed: 14-10-2017].

Meso, P., Jain, R. (2006) Agile Software Development: Adaptive Systems Principles and Best Practices. *Contemporary Practices in Systems Development*, 32(3) 177-191. Available from: http://www.tandfonline.com/doi/abs/10.1201/1078.10580530/46108.23.3.20060601/93704.3 [accessed 11-04-2018].

Mori, S. (2017) Deep Architecture Neural Network-Based Real-Time Image Processing for Image-Guided Radiotherapy. *Physica Medica*, 40, 79-84. Available from: http://www.sciencedirect.com.proxy.library.lincoln.ac.uk/science/article/pii/S1120179717302302?_rdoc=1&_fmt=high&_origin=gateway&_docanchor=&md5=b8429449ccfc9c30159a5f9aeaa92ffb [accessed: 14-10-2017].

Nattkemper, T.W., Wersing, H., Schubertm, W., Ritter, H. (2002) A Neural Network Architecture for Automatic Segmentation of Fluorescence Micrographs. *Neurocomputing*, 48(4) 357-367. Available from: http://www.sciencedirect.com/science/article/pii/S0925231201006427 [accessed: 12-10-2017].

Ren, X., Malik, J. (2003) Learning a Classification Model for Segmentation. In: *9th IEEE International Conference on Computer Vision*, Nice, France, 13-16 October. New York, United States of America: IEEE, 1-9. Available from: http://www.ieeexplore.ieee.org/abstract/document/1238308/ [accessed: 18-03-2018].

Shapiro, L.G., Stockman, G.C. (2001) *Computer Vision*, 1st edition. United Kingdom: Pearson.

Sutton, R.S. and Barto, A.G. (1998) *Introduction to reinforcement learning*, 2nd edition. United States of America: MIT Press Cambridge.

Varma, V.S., Durga, K.K., Keshavulu, K. (2013) Seed Image Analysis: Its Applications in Seed Science Research. *International Research Journal of Agricultural Sciences*, 1(2) 30-36. Available from: http://www.advancedresearchjournals.org/journal/irjas/archive/june-2013-vol-1(2)/seed-image-analysis-its-applications-in-seed-science-research [accessed: 10-10-2017].

Wang, Z., Guo, X., Wu, X., Wang, Z. (2017) Superpixel segmentation based on multiple seed growth. In: *2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, Xiamen, China, 6-9 November. New York, United States of America: IEEE, 529-534. Available from: http://www.ieeexplore.ieee.org/document/8266535/ [accessed: 17-03-2018].

Wilson, J.M. (2003) Gantt charts: a centenary appreciation. *European Journal of Operational Research*, 149(2) 430-437. Available from http://www.sciencedirect.com/science/article/pii/S0377221702007695 [accessed 28-03-2018].

Yaram, S. (2016) Machine learning algorithms for document clustering and fraud detection. In: *IEEE 2016 International Conference on Data Science and Engineering (ICDSE)*, Cochin, India, 23-25 August. New York, United States of America: IEEE, 1-6. Available from: https://www.ieeexplore.ieee.org/abstract/document/7823950/ [accessed: 10-04-2018].

Zhong, G., Ling, X., Wang, L (2018) From shallow feature learning to deep learning: benefits from the width and depth of deep architectures. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 0(0) 1-14. Available from: https://onlinelibrary-wiley-com.proxy.library.lincoln.ac.uk/doi/full/10.1002/widm.1255 [accessed: 23-04-2018].

APPENDIX

APPENDIX A: EXAMPLE SEGMENTATION OUTPUTS

Appendix A shows a subset of examples of the final segmentation outputs produced by the superpixel CNN seed segmentation algorithm at a higher resolution. These output examples are derived from seed sample images contained within the test dataset.