Deep learning in seed image classification using domain randomization and convolutional neural networks

by

Niketa Penumajji

B.E., Savitribai Phule Pune University, India, 2016

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2021

Approved by:

Major Professor
Dr. Mitchell Neilsen

Copyright

Abstract

The objective of this thesis is to study the application of deep learning in seed image classification using convolutional neural networks. The images are captured using an unmanned aerial vehicle as might be used in a plant breeding program. Plant breeding programs extensively monitor the evolution of seed kernels for seed certification, wherein lies the need to appropriately label the seed kernels by type and quality. However, the breeding environments are large where the monitoring of seed kernels can be challenging due to the relatively small size of seed kernels. For this experiment, we use an unpiloted aerial vehicle to capture images at low altitude while being able to access even the remotest areas in the environments. In a small breeding program, they might use fixed cameras instead. A key bottleneck in the labelling of seeds using UAV imagery is drone altitude, i.e., the classification accuracy decreases as drone altitude increases due to lower image detail. Convolutional neural networks are a great tool for multi-class image classification where there is a training dataset that closely represents the different scenarios that the network might encounter during evaluation. However, with the seeds being in a breeder environment coupled with the varying image resolution and clarity, it is challenging to generate a training dataset that covers all possible evaluation scenario. This work addresses the challenge of training data creation using Domain Randomization wherein synthetic image datasets are generated from a meager sample of seed images captured by the bottom camera of an autonomously driven Parrot AR Drone 2.0. Also, the work includes a seed classification framework as a proof-of-concept using the convolutional neural networks of Microsoft's ResNet-100, Oxford's VGG-16, and VGG-19. To enhance the classification accuracy of the framework, an ensemble model is developed

resulting in an overall accuracy of 94.6%. The task of classification is performed on five distinct types of seeds, canola, rough rice, sorghum, soy, and wheat.

The second use case is the drones are used for aerial surveying where the UAV imagery is used to estimate the area of a plot. The work demonstrates how the area can be calculated using the pixels of an image and the camera specifications of the drone.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my sincere gratitude to Dr. Mitchell Neilsen for his expertise, guidance and patience throughout the process of writing my thesis. This work would not have reached its completion without his assistance and generous insights.

I would also like to extend my thanks to my committee members, Dr. Lior Shamir and Dr. Torben Amtoft for their suggestions and support.

I would like to thank my friend and fellow graduate student Venkat Margapuri for his treasured support which was influential in shaping my work.

Finally, my appreciation goes out to Friday James, Kai Zhao, Divyavani Lakkireddy, Sujith Gunturu and our collaborators from USDA for their support.

Dedication

I would like to dedicate this work to my parents Swamy and Sharda for believing in me and allowing me to explore opportunities. My sister, Aashikaa and my brother-in-law, Dheeraj for their constant encouragement and support.

Chapter 1 - Introduction

In the recent past, the term Artificial Intelligence has gained a lot of popularity. Albeit the term was coined in 1956, the virtual assistants used in smartphones, autonomous vehicles, intelligent chatbots, etc., show how AI (Artificial Intelligence) has evolved over the years. In simple terms, AI can be defined as the “concise effort to automate intellectual tasks performed by humans.”

1.1 The relation between Artificial Intelligence, Machine Learning, and Deep Learning

Consider an example of an online chess game that is better used to understand what AI is. In the game where one player is human and another computer, some form of intelligence is observed behind the moves played by the computer. In this case, AI refers to the output of a computer. It is exhibiting artificial intelligence. Here, the term AI does not refer to any details about how the problem is solved.

The solution, in this case, turned out to be not just mimicking human behavior but mimicking how humans learn. This is exactly the idea behind machine learning. Train an algorithm with lots of data and letting it figure out things is the idea behind ML (Machine Learning). As more and more algorithms developed tackling problems became easy. But a few problems that humans found easy were still hard for machines such as speech or text recognition etc. since machine learning is all about mimicking the human learning behavior, scientists have come up with the idea of mimicking the human brain which led to the development of neural networks.

Therefore, deep learning is all about using neural networks with more neurons, layers, and interconnectivity. The image below shows the relation between AI, ML, and deep learning.

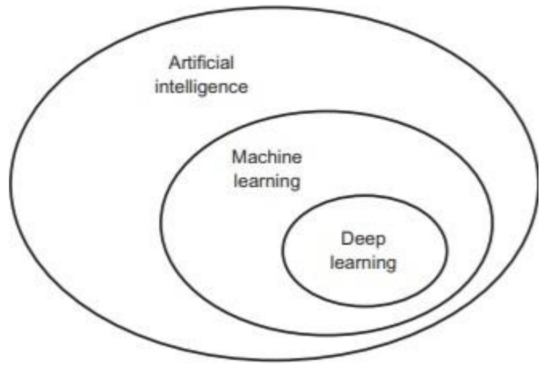

Figure 1: Relation between Artificial Intelligence, Machine learning and Deep learning

1.2 Deep learning with Autonomous vehicles

Research in the field of unpiloted aerial vehicles is evolving fast and gaining popularity. Attaining autonomous behavior for UAVs (unpiloted aerial vehicles) is active research. Current technology allows the UAVs to perform complex maneuvers and navigate autonomously be it indoors or in an open environment. In the recent past UAVs have become available for numerous applications such as remote sensing, security and surveillance, search, and rescue, delivery of goods, precision agriculture, etc. The work demonstrates how drone imagery is used in two principal areas: I) estimating the area of a plot. II) classification of seeds

1.2.1 Classification of seeds

Assessing various complex traits of seed such as its growth, development, tolerance, resistance, ecology, yield, and many other parameters of measurement which is often known as seed Phenotyping where one of its focus areas happens to be seed certification. Seed certification, performed in plant breeding environments, is an internationally recognized system to maintain and make available to the public, high quality seed and propagating materials of superior adapted crop varieties grown and distributed to ensure varietal identity and purity. Seed certification is based on the premise that proper identification of crop varieties is essential to everyone who handles seed. In seed certification seeds must be properly labeled which is often challenging as the breeding environments are large. Recent developments in the areas of deep learning have led to the evolution of powerful convolutional neural networks (CNN) that aid in the classification (or labeling) of a certain entity. CNNs perform best when they are trained on datasets that closely resemble the test criteria. However, the creation of a training dataset for seed labeling is challenging since seeds evolve leading to a change in their morphology. As a result, a universal dataset that resembles the evolution of seeds in a breeding environment is not available. More importantly, such datasets are required to be catered to a specific breeding environment. The creation of such datasets requires a profusion of seed sample availability which is not always possible. The need for a large seed sample is addressed by the application of a technique named Domain Randomization (DR) i.e., the idea of training models on simulated images that transfer to real images. The use of DR means that a large synthetic dataset that resembles real-world seeds in assorted sizes and orientations may be generated using a meager sample of real seeds. The capture of the sample of seeds for DR may be automated using autonomous unmanned aerial vehicles (UAV) aka drones. Drones can be flown autonomously at low altitudes and are a better choice to capture images in controlled plant breeding environments.

1.2.2 Aerial survey for plot area estimation

The use of UAVs for surveying purposes is becoming increasingly popular for several reasons. Professionals rely on drones to carry out a variety of surveying tasks. Drones are equipped with cameras that come with a maximum resolution of 5.2k ensuring the better image and video quality. A greater pixel density maximizes the overall accuracy. Since all the images are captured through the air, drones reduce the risk of potential danger when trying to capture data in a compromised or sensitive environment. Also, UAVs make it easier than ever before to access hard to reach locations. Large areas of land can be covered in a short period thus eliminating the limited capabilities of imagery acquisition. All the above reasons make UAVs the best choice for surveying over manual surveys. However, estimating the area to obtain the exact results can be challenging as various factors such as the height at which the drone is being flown and the overall resolution of the image needs to be considered. Increasing the drone height might result in an inferior image quality making the overall process difficult.

1.3 Related Work

The work by Yoda, et. al. [9] applies the technique of domain randomization to train Mask R-CNN, an instance segmentation neural network on a synthetic image dataset of 20 different cultivars belonging to barley, four cultivars of wheat, and one cultivar each of rice, oat and lettuce. The average precision (AP) values computed based on mask regions at the varying intersection over union (IoU) thresholds achieved comparable AP50 values of 96% and 95% for the synthetic and real-world datasets respectively. Ward, Moghadam, and Hudson [10] conducted similar experiments to perform automated segmentation of individual leaves of a plant for high-throughput phenotyping. The proposed technique leveraged synthetic images of plants generated from real plant images. The Mask R-CNN architecture was trained with a combination of real and syntheticimages of Arabidopsis plants. The proposed technique achieved a 90% leaf segmentation score on the A1 test set and outperformed the state-of-the-art approaches for the CVPPP Leaf Segmentation Challenge (LSC). Venkat Margapuri [6] conducted similar research for the classification of seedsusing domain randomization on self-supervised learning algorithms such as Momentum Contrast

(MoCo) and Build Your Own Latent (BYOL) where ResNet-50 is used as the backbone in each of the networks. Gulzar, Hamid, Soomro, Alwan and Journaux [11] developed a seed classification system for 14 different types of seeds employing the concept of transfer learning on a pre-trained model of VGG-16. The results showed a classification accuracy of 99% on a test image dataset of 234 images. Buters, Belton, and Cross in [12] conducted experiments to perform seed and seedling detection and classification using unmanned aerial vehicles. The technique of object-based image analysis (OBIA) with eRecognition software was used as part of the experiments. The results showed the feasibility of low-cost commercially available UAVs in monitoring ecological recovery. Coulibaly, Kamsu-Foguem, Kamissako, and Traore [28] proposed a deep learning technique with transfer learning in millet crop images using VGG-16. The goal of the study was to identify mildew disease in pearl millet. Fine-tuning was performed on a pre-trained model of VGG-16 by adding two fully connected layers and training the model on the image dataset. An accuracy of 95%, the precision of 90.50%, recall of 94.5%, and f1score of 91.75%. In the study conducted by Mukti and Biswas [14], transfer learning-based plant disease detection was performed using ResNet-50. The dataset included 38 different classes of plant leaf images wherin 70295 training images and 17572 validation images were used. Fine-tuning was performed to enhance the performance of the ResNet-50 model that eventually yielded a training accuracy of 99.80%. 33 images belonging to the 38 classes of leaves were used for testing. The proposed model yielded an accuracy of 100% on the test dataset. Muneer and Fati [15] proposed a deep learning technique to classify and recognize the desired herb from thousands of herbs using shape and texture features. The proposed system employed two classifiers, Support Vector Machine (SVM) and Deep Learning Neural Network (DLNN). The models were tested on a dataset containing 1000 herbs. The results showed that the SVM model achieved recognition accuracy of 74.63% whereas the DLNN achieved 93% accuracy. Furthermore, the processing time was four seconds for SVM and five seconds for DLNN. A deep-learning classification system for identifying weeds using high-resolution UAV imagery was proposed by Bah, Dericquebourg, Hafiane, and Canals in [16]. The proposed system was applied to images of vegetables captured about 20m above the soil using a UAV. The results showed that weed detection was effective in different crop fields with overall precisions of 93%, 81%, and 69% obtained for beet, spinach, and bean respectively. Bristeau, Vissière, Callou, and Petit [17] discuss the navigation and control technology embedded in the Parrot AR Drone. The article sheds light on the low-cost inertial sensors, computer vision techniques, attitude and velocity estimation, and control architecture used in the UAV. The work described in [18] analyzes the morphometric characteristics of seed to predict the behavior of the seed in terms of development, tolerance, and yield in various environmental conditions. The work uses domain randomization to train on networks Mask R-CNN and YOLO (You Only Look Once), for seed phenotyping using Tensorflow. Synthetic datasets are used as part of the work to train the networks.

Zhang et. al. [31] combined RL with model predictive control to train a deep neural network on obstacle avoidance when deployed on a UAV. Zho, et. al. [2] proposed an RL based model that navigates in an indoor scene to search for a target object. Their Siamese model takes an input image of target object and a scene observation. Chaplot, et. al. [3] introduced Active Neural Localization which predicts a likelihood map for the agent location on the map and uses such information to predict its policy. Ross, et. al. [4] were able to produce on-board UAV model to avoid obstacles in the forest using the DAgger algorithm [5] which is one of the most used Imitation Learning algorithms. The trained model depends on different kinds of visual features from the input images. Kelchtermans, et. al. [29] used LSTM neural network and imitation learning to perform the navigation based on a sequence of input images. Some work also used direct Supervised Learning for control. Giusti, et. al. [7] developed a DL based model to control a drone in order to fly over forest trails. The actions produced by the network were discrete (go right - go straight - go left). Smolyanskiy, et. al. [8] used a similar model for trail following and added entropyreward to stabilize the drone navigation.

Chapter 2 - Aerial Survey

A framework in Python-OpenCV is used to extract the frames from the video captured using the bottom camera of Parrot AR Drone 2.0. The experiments are initially conducted in an indoor setting to estimate the area of a small plot which is a square in this case. To verify the accuracy of the results similar experiment is conducted by capturing a video of an open football field. In the indoor setting, a square has been set up on the ground with each of its sides measuring 1ft. The drone is made to fly above the square at a height of 0.6m. Flying the drone at a height lesser than 1m allows capturing a clear image of the square. As the height of the drone increases, smaller images are captured, and this reduces the image quality. Image quality plays an important role here as the area is estimated based on the pixels.

2.1 Drone control software

A Parrot AR Drone 2.0 is used as part of the experiment. It is a quadcopter manufactured by the French company Parrot SA. The drone uses a 3630 OMAP CPU, a processor that is based upon a 32-bit ARM cortex and runs with 1 GHz. The drone uses 4 brushless motors running at 28.5 revolutions/minute which are controlled by an 8 MIPS AVR CPU on each motor controller. The drone has a front and a bottom camera to capture images and videos. The front camera provides an HD image resolution (720p-30fps) whereas the bottom camera provides QVGA (320 x 240) at 60fps. The drone may be controlled via a mobile application supported on Android and iOS provided by Parrot SA. Besides the mobile app, the quadcopter has support for autonomous navigation through libraries supported in Robot Operating System (ROS) and nodeJS. It turns out the Parrot AR comes with an excellent node.js client library called node-ar-drone that is perfect for building onto. JavaScript turns out to be a great language for controlling drones because it is so inherently event driven. These libraries are leveraged to operate the drone autonomously both for area estimation and in the breeding environment for seed classification. While the node-ar-drone library provides a plethora of functions to operate the quadcopter, the ones used for the current work are take-off, land, spin, and video capture and stop. The behavior of the functions is self-intuitive from the names of the functions. By default, the Parrot AR Drone 2.0 serves a wireless network that clients connect to. After installing the node library, it is possible to fly the drone using a node.js REPL (Read-Evaluate-Print-Loop) and steer it. To send a feed from the drone's camera the best way is to open a connection and send a continuous stream of PNGs to the web server of the website that is currently used. The web server continuously pulls PNGs from the drone's camera using the AR drone library. The video captured by the drone is output in .H264 format when the video capture function of the node-ar-drone library is used. Videos in raw .H264 format don't carry rate information. However, different frames of the video are required to be extracted for estimating the area and for the creation of synthetic images. As a work-around, the FFmpeg library is used to convert the video in .H264 format to .mp4 format rendering it conducive for frame extraction using the Python-OpenCV framework.

2.2 Area estimation

To estimate the area first an indoor setup is considered where a square is laid with each of its sides measuring 1 foot. The drone is made to fly autonomously above it at a height of 0.6m. A continuous stream of images is captured using the bottom camera of the drone from which the frames are extracted using the Python-OpenCV framework. To estimate the area, the pixel distance needs to be calculated of any side of the square. This can be achieved with the help of any measuring tool; GIMP is used as part of the work. The indoor setup is as follows:

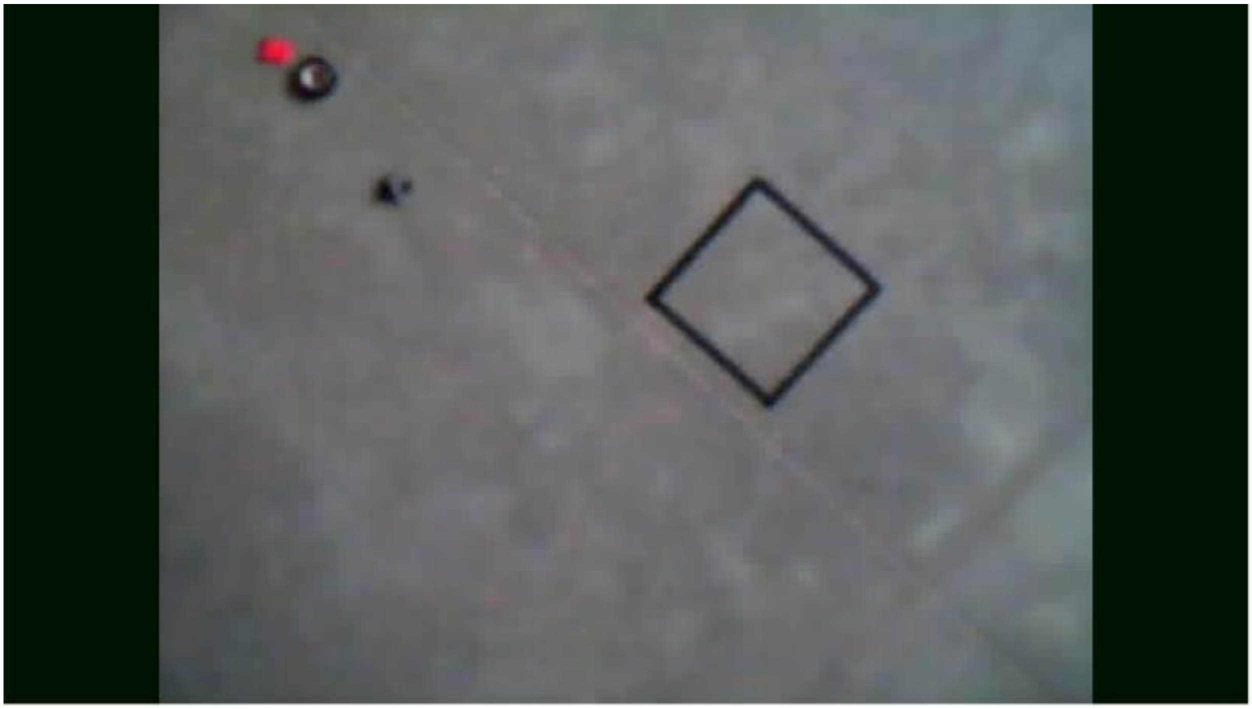

Figure 2: Indoor setup of a square

Once the pixel distance is obtained the next step is to determine the ground sample distance (GSD) of each pixel. The ground sample distance is the distance between the centers of two adjacent pixels of the image taken. Since the images are captured via the drone a few specifications of the drone such as the camera sensor width and the focal length are required to calculate the GSD. The Parrot AR Drone 2.0 uses a CMOS sensor whose sensor width is 6.17mm with a focal length of 4.55mm. The below figure provides a visual representation of the camera sensor and field of view. Using the width of the camera sensor, focal length, and drone altitude the ground sample distance can be calculated.

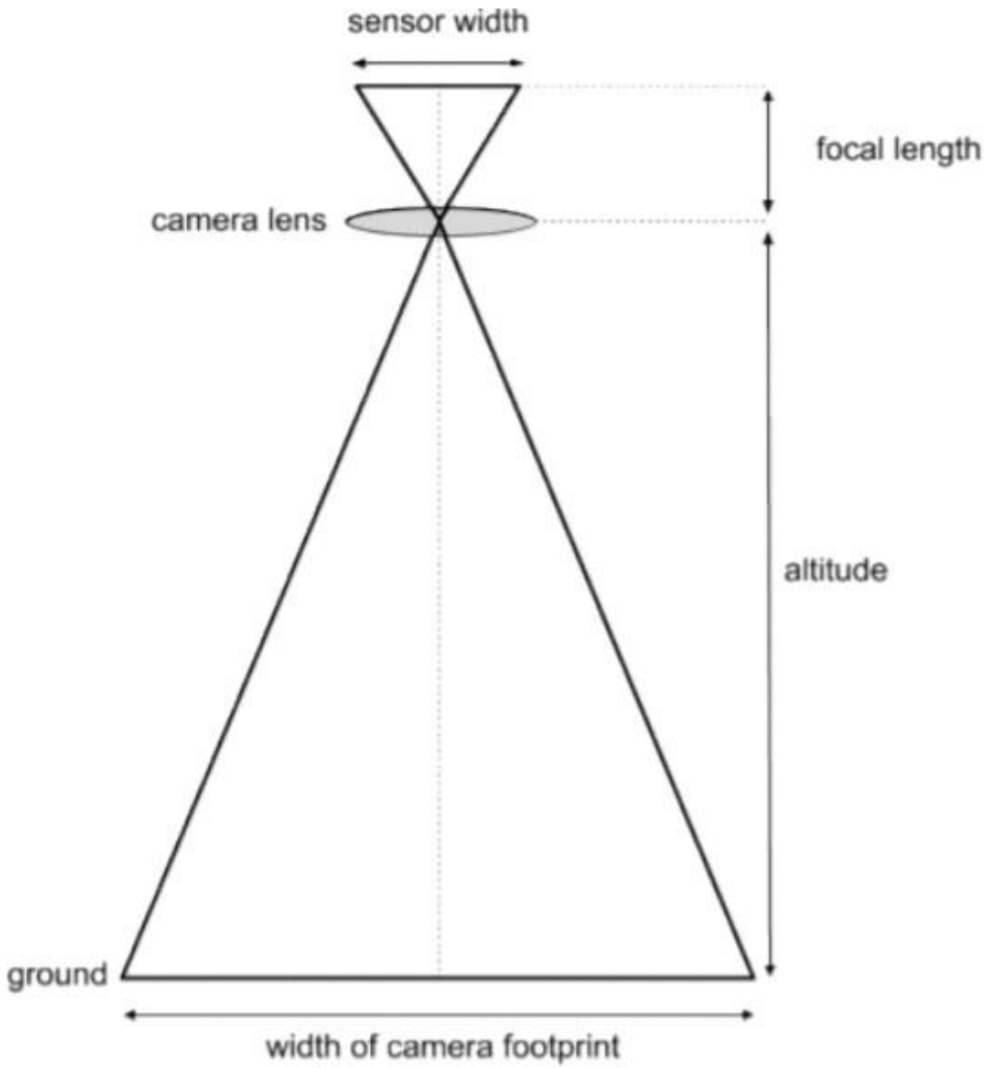

Figure 3: Visual representation of camera sensor and field of view [12]

GSD is calculated as follows:

GSD = (sensor_width * altitude * 100) / (focal_length * image_width)

The sensor width and the focal length are obtained from the drone used as part of the experiment. As mentioned earlier in section 2 the drown is flown at an altitude of 0.6m. Here a 640x320 image is considered hence the image width is 320 pixels. Therefore, the final values are:

sensor_width = 6.17 mm

focal_length = 4.55 mm

altitude = 0.6 m

image_width = 320pixels

plugging the values in the GSD equation gives:

GSD = (6.17 * 0.6 * 100) / (4.55 * 320) which equates to

GSD =0.2542 cm/pixel

The value of the ground sample distance of the photo considered as part of the experiment is 0.2542 cm for every pixel. Once the GSD is obtained the next step is to determine the width of the camera footprint as shown in the fig (in meters). Camera footprint can be calculated as follows:

footprint_width = GSD * pixel_width_of_image

= 0.2542 cm/pixel * 320pixels

footprint_width = 81.344 cm

since the footprint width is needed in meters dividing the number by 100 converts the obtained answer to meters. Therefore, the footprint width is **0.81344 m.** This gives us the footprint width of the entire photo; however, the distance between two ends of any side of the square is needed. To calculate the pixel distance of the side of a square the obtained footprint width in meters is multiplied with the pixel distance of the length of the side.

Length of each side = (0.81344 * 35) / 100 = 0.2847 m

where 35 is the pixel distance of the side of a square that is obtained using the GIMP tool.

To view this distance in imperial units the conversion of meters to feet is done:

1 m = 3.281 ft

applying it to the metric distance gives:

0.2847 m * (3.281 ft / 1m)

= 0.99 ft

The meter units cancel out thus leaving 0.99 ft ~ 1 ft. Therefore, the length of each side in the square as calculated is 1 ft. Similar experiments are conducted by capturing images in an outdoor setting. The image below shows a football field where the distance between any 2 lines is 10 yards. The same is proven using the above logic.

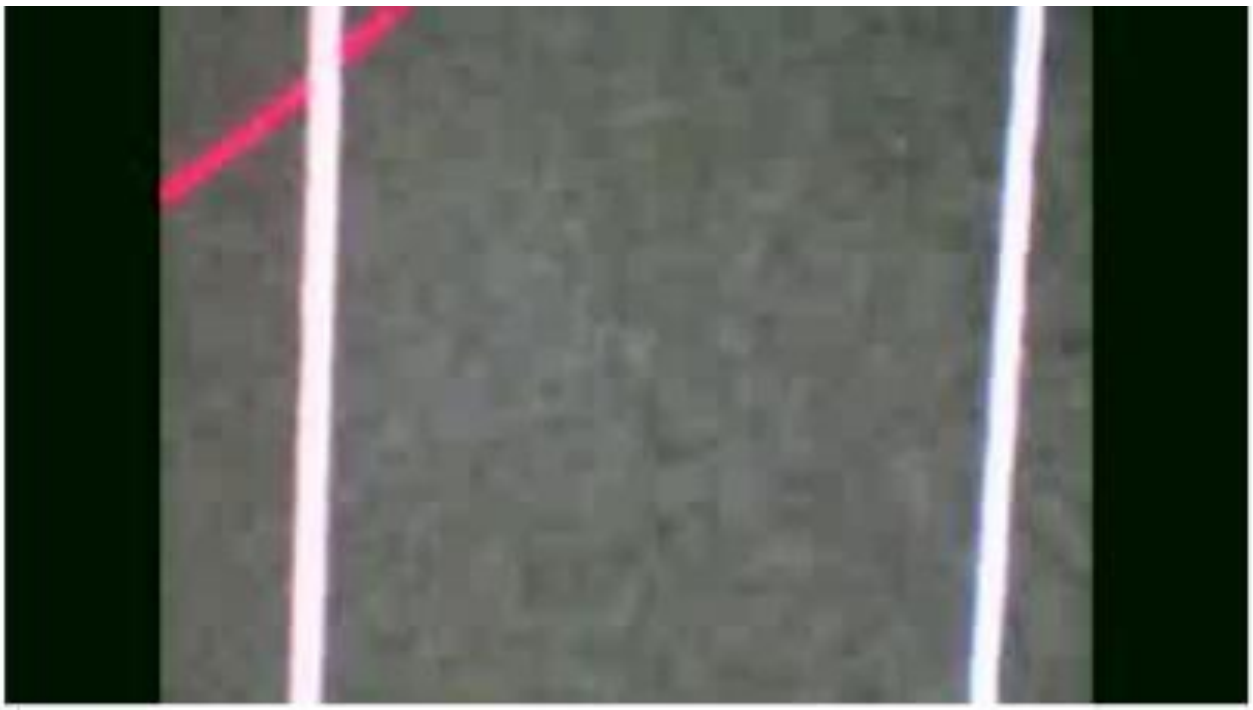

Figure 4: Image of a football field

Chapter 3 - Seed Classification Framework

A framework in Python-OpenCV is used to generate synthetic image datasets. Domain randomization is a systematic approach to generate data that aims to enhance the generalization of the machine learning algorithms to new environments. The approach tries to find a representation that generalizes across different environments, known as domains. The main purpose of DR (domain randomization) is to provide enough simulated variability at training time so that the model can generalize to real world data. A Domain Randomization framework using these datasets is developed as part of the work which constitutes of steps:

1. Initially frames are extracted from the videos captured by the drones and the frames on which the seeds are visible are identified.
2. The frames are loaded into GIMP to extract the seeds on the image. The foreground on each image represents the seeds, this can be extracted by separating the alpha channel from the image.
3. An empty canvas is chosen onto which the extracted seeds are laid. The canvas acts as a background for the image. The ideal canvas is the test site where the seeds are laid for phenotyping.
4. The extracted foreground of the seeds is laid on the canvas in different sizes and orientations to resemble the seeds in the phenotyping environment.

Image augmentation is applied to the generated synthetic seeds at random. Each of the seeds is put through a sequence of brightness adjustment, horizontal flip, vertical flip, rotation, and scaling of various degrees before being laid on the canvas.

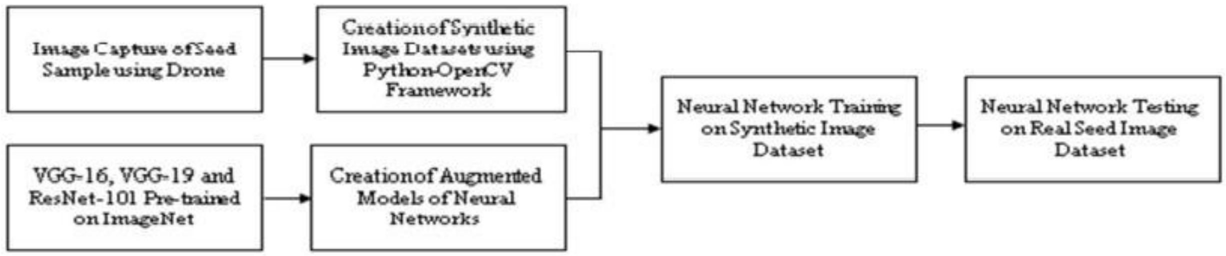

Figure 5: Workflow of the seed classification system

3.1 Synthetic Dataset

The seeds of canola, rough rice, sorghum, soy, and wheat are used for the experiment. 30 seeds of each category are randomly selected from a large pool of available seeds of each seed type and are placed in the prototypical seed phenotyping environment. The Parrot AR Drone 2.0 is flown autonomously over the phenotyping environment and videos are captured using the bottom camera of the drone. As part of the experiment, the drone is made to fly at three different heights of 0.3m, 0.5m, and 0.7m. Five images of the white light-emitting lightbox are used as the canvas(background) for the synthetic images. Although the lightboxes look the same while emitting light, it is not the case and minor discrepancies in the intensity are noticed. Thus, using the images of different lightboxes ensure that these discrepancies are captured during the creation of the synthetic dataset. The synthetic images generated are of the size 224x224x3 in order to match the input size of the neural networks used to train the dataset. For each of the seeds, 1000 synthetic images are generated each image consisting of approximately 20 -50 seeds. As the seeds are captured from three different heights a diverse dataset is generated i.e., the 1000 images of a single seed comprise 333 images captured at a height of 0.3m, 333 at 0.5m, and the rest at 0.7m. overall 5000 images are generated amongst all the seeds. One of the difficulties in classifying the seeds is the appearance of the seeds at different heights. It might not be an issue if the classification of seeds is done from the images captured at a single height. However, it is not always feasible while using drones as they are mobile and are influenced by external factors such as wind and obstacles. It is observed that a given seed appears smaller as the height increases and vice-versa. Owing to this, the training dataset comprising of images from a fixed height may not be a representative sample of the real-world test set that the neural network might encounter. Hence, the drone is programmed to capture images from varying heights of 0.3m, 0.5m, and 0.7m to ensure that a real-world representative sample of seeds is obtained.

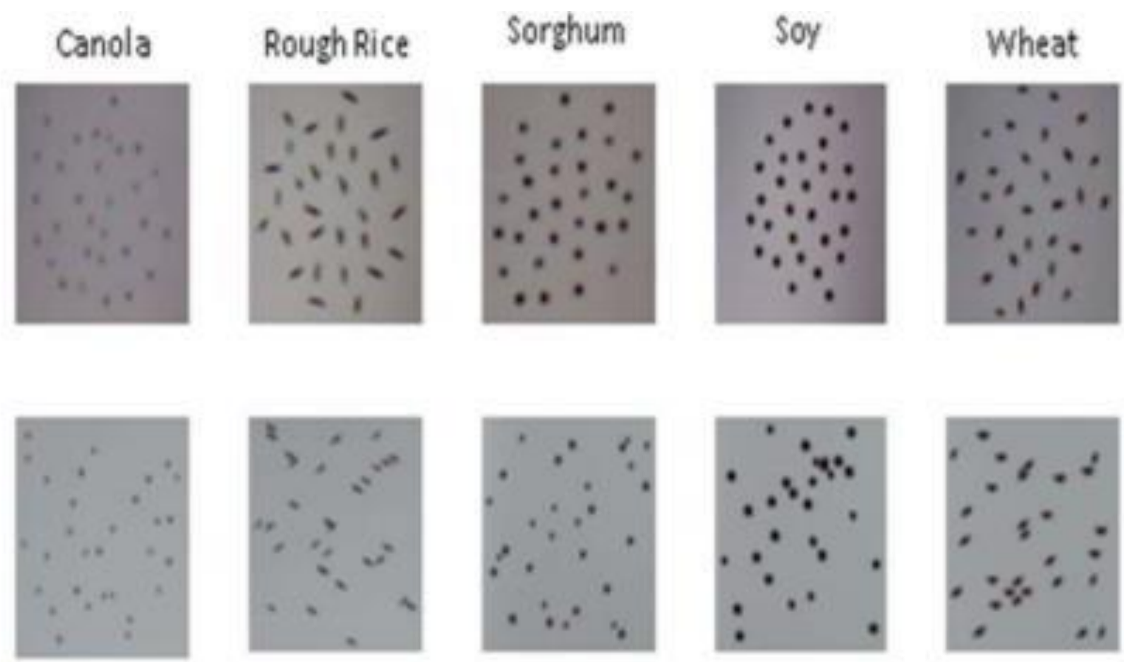

Figure 6: Top row shows real seeds captured by the UAV; Bottom row shows synthetic images generated by the DR framework using the real seeds.

Figure 7: Image Capture of Seeds in Phenotyping Environment using Parrot AR Drone 2.0

3.2 Neural Networks

In this work, three convolutional neural networks namely Oxford's VGG-16, VGG-19, and Microsoft's ResNet-101 are used. The architecture of each of the following networks is described as follows.

3.2.1 VGG – 16

VGG16 is a CNN proposed by K. Simonyan and A. Zisserman from the University of Oxford. The model achieves an accuracy of 92.7% in ImageNet which is a dataset consisting of over 14 million images belonging to 1000 different classes. In this architecture, the first two layers are convolutional layers with 3x3 filters. A total of 64 filters are used in the first two layers hence it ends up having a volume of 224x224x64 as the height and width of the convolution are the same. Next, a pooling layer is added which reduces the height and width of a volume thus bringing it down from 224x224x64 to 112x112x64. A few more convolutional layers are added with 128 filters are thus resulting in a new dimension 112x112x128. Another pooling layer is added so the dimension now is 56x56x128. The architecture now has 2 convolutional layers with 64 filters and 2 more layers with 256 filters. A few more layers with 512 filter and pooling layers are added resulting in the final architecture too have a total of 7x7x512 fully connected layers with 4096 units, resulting in a softmax output for one in 1000 classes. All the hidden layers are equipped with rectification (ReLU) non-linearity. A wonderful thing about VGG – 16 is that instead of having too many hyper parameters it focuses on just having one convolutional layer with 3x3 filters with a stride of 1 and with the same padding. In all the Max pooling layers 2x2 filters with a stride of 2 are used. Number 16 in VGG -16 refers to the fact that it has 16 layers with certain weights.

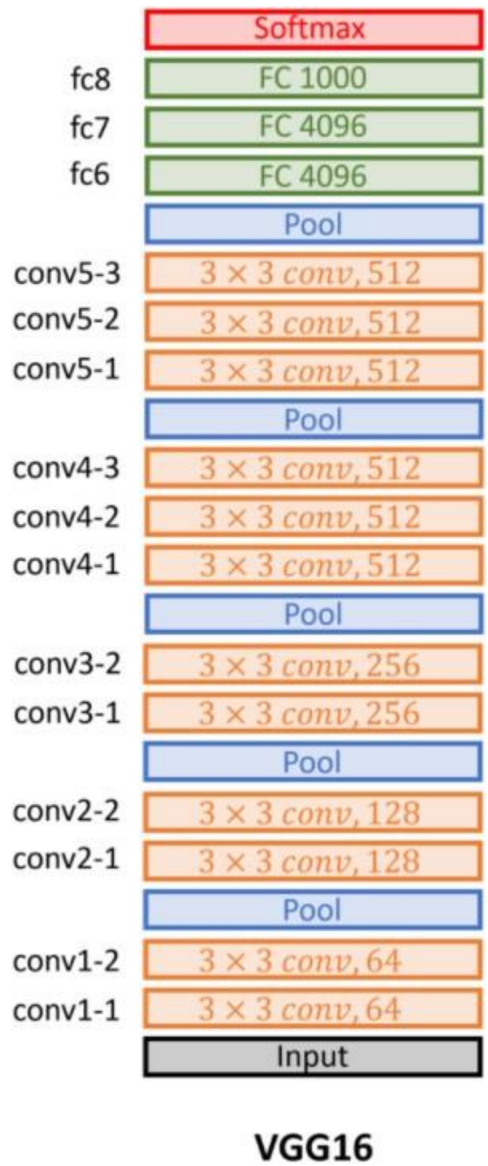

Figure 8: Architecture of VGG-16

3.2.2 VGG – 19

VGG – 19 is also a variant of the VGG model which consists of a total of 19 layers with 16 convolutional layers, 3 fully connected layers, 5 MaxPool layers, and 1 SoftMax layer. It takes a fixed size (224x224) RGB image input. The model does a preprocessing step which is to subtract the mean RGB value from each pixel, which is computed over the entire training dataset. Kernel of 3x3 size with a stride of 1pixel is used. The model uses spatial padding to preserve the spatial resolution of the image. A max-pooling layer of 2x2 and with a stride of 2 followed by a Rectified linear unit (ReLu) to improve the computational time and classification accuracy is used.

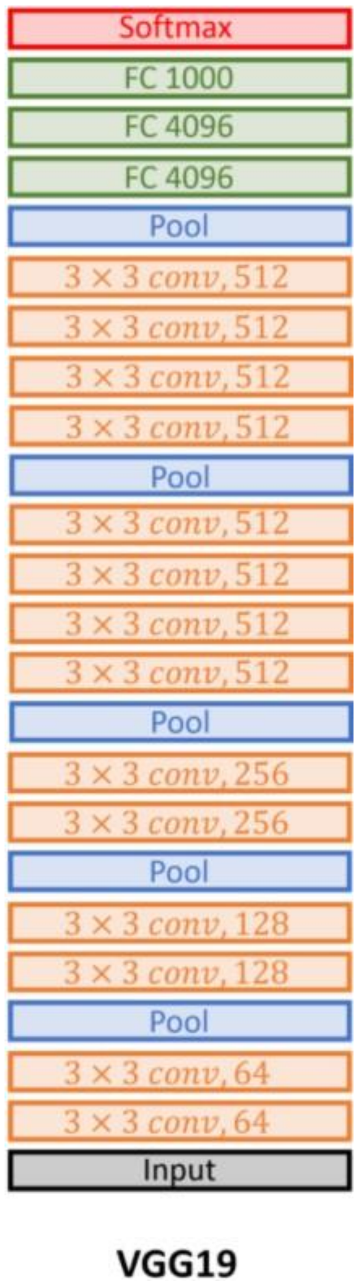

Figure 9: Layers in VGG-19 Network

3.2.3 Resnet-101

Residual Networks are one of the most groundbreaking works in the field of computer vision/deep learning over the last few years. ResNet makes it possible to train up to hundreds or even thousands of layers while still achieving compelling performance. Other than image classification many computer vision applications such as object detection and face recognition also seem to perform well with the model. Resnet comes with multiple variants such as ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-110, ResNet-152, ResNet-164, ResNet-1202 etc. All these models have the same concept, but they are implemented with a different number of layers.

The convolutional layers in the ResNet architecture mostly have 3x3 filters which follow two basic principles:

i) if the output feature map is same the layers must have the same number of filters.

ii) if the size of the feature map is halved, then the number of filters is doubled in order to preserve the time complexity of each layer.

Resnet models comparatively have fewer filters than VGG nets. The core idea of ResNet is introducing a so-called "identity shortcut connection" that skips one or more layers. The network uses an identity shortcut F(x{W}+x) whenever the input and the output are of the same dimension. In the case of different dimensions, the shortcut performs identity mapping with an extra 0 entries padded to increase dimensions which causes the introduction of an additional parameter. The shortcut F(x{W}+x) is used to match the dimension done by 1x1 convolutions. The figure below shows ResNet-34-layer residual architecture.

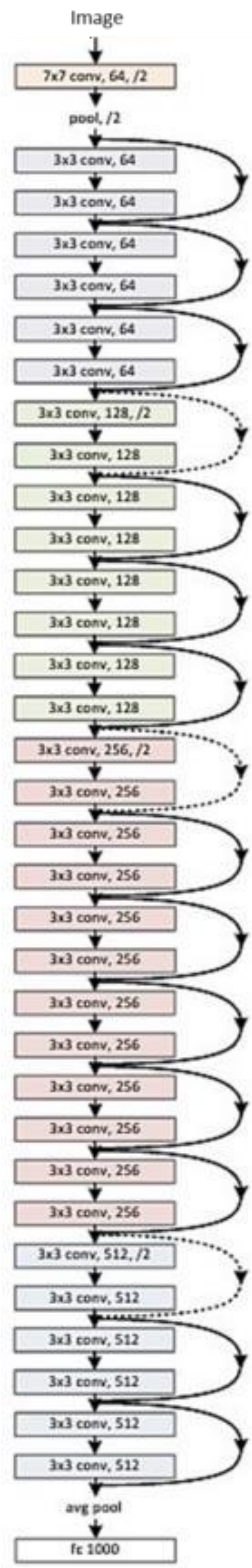

Figure 10: Resnet Architecture

3.3 Experiments & Results

The three neural network architectures are imported from the canned architectures provided by Keras as part of their Applications module. The imported models are pretrained on the ImageNet dataset that comprises of 1000 classes. The key idea behind the use of pre-trained models trained on ImageNet is that the notion of Transfer Learning may apply. Transfer learning is a technique of applying the knowledge gained from previous learning to solve a new problem or a related one. The ImageNet dataset does not even remotely contain anything like the dataset used as part of the work. In case where the source and the target don't share a common domain applying transfer learning may not be successful and often results in poor performance. Since the networks are huge, training them from scratch requires excessive amounts of data. To better train the network while preserving its knowledge from the ImageNet dataset, the architecture of the neural networks is augmented. As for augmented architectures, three Sequential models (one for each of VGG-16, VGG-19, and ResNet-101) are built from the imported models wherein all the layers except the fully connected layers and softmax layer are copied to the Sequential models. The VGG models consist of three fully connected layers whereas ResNet-101 has one. Three fully connected layers are added to the Sequential models followed by a softmax layer for classification. All the transferred layers from the pre-trained models are rendered frozen for training and only the three fully connected layers added to the model are trained. The dataset used for the experiment is the synthetic image dataset described in 3.1. 80% of the dataset is used for training and 20% of the dataset is used for validation. Experiments with different hyperparameters are conducted in the training phase. Upon training each model with a different set of hyperparameters, the models that yield the best accuracy and loss are saved. The hyperparameters used for each of the best models are as shown in Table 1.

Hyperparameter	VGG-16	VGG-19	ResNet-101
Nodes per trained layer	512	512	1024
Learning Rate	1 x e^{-3}	1 x e^{-3}	1 x e^{-3}
Learning Rate Decay	100 steps @0.96	100 steps @0.96	No Decay
Dropout	0.4	0.5	0.5
Batch Size	32	32	32
Optimizer	Adam	SGD	Adam

Table 1: Hyperparameter values for Training

The training accuracies and losses of the best models for the fine-tuned neural networks are as shown in Fig. 9

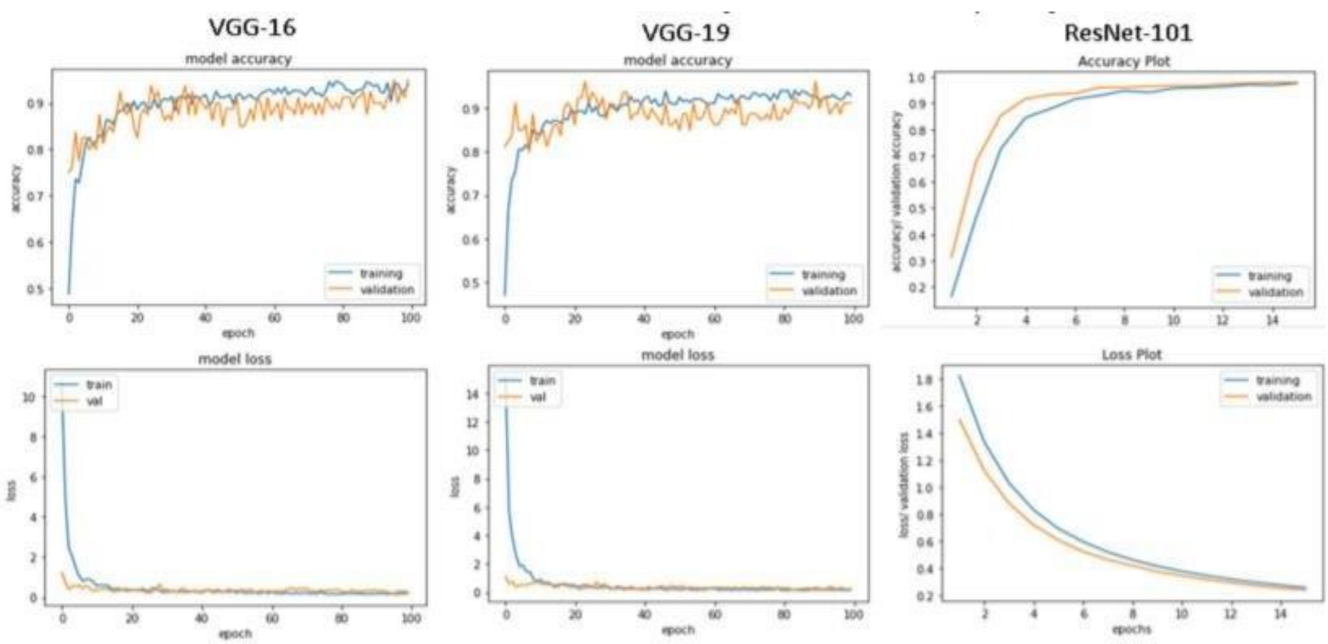

Figure 11: Training and Validation Accuracies and Losses for Fine Tuned VGG-16, VGG-19 and ResNet-101

Overfitting is commonly observed across all three of the models. The use of Dropout helped with reducing the overfitting of the models. However, it results in the validation accuracy being higher than training accuracy in parts. Overall, the validation accuracy for each of the fine-tuned models of VGG-16, VGG-19, and ResNet-101 at the end of the training phase is 96.43%, 90.03%, and 97.81% respectively.

3.4 Evaluation

The image dataset for evaluation is generated from the images captured by the Parrot AR Drone 2.0 while flying over the seed phenotyping environment at different heights. While the images used during the training phase are generated from videos captured from pre-defined heights of 0.3m, 0.5m, and 0.7m, the images for the test dataset are generated from videos captured from varying heights of less than one meter. The image dataset is generated by extracting frames from videos using OpenCV. The images for the test dataset consist of 75 images containing 20 – 50 seeds of a certain seed type captured from varying heights. Overall, a test dataset consisting of 450 images is used to evaluate the performance of each of the networks. The results of evaluation based on the metrics of Accuracy, Precision, and Recall are as shown in Table 2.

The metrics are briefly described below:

Accuracy: Accuracy is defined as the fraction of the samples in the dataset correctly classified by the classifier. It is given by $(true\ positives + true\ negatives)/\ (true\ positives + true\ negatives + false\ positives + false\ negatives)$.

Precision: Precision is given by $true\ positives/\ (true\ positives + false\ positives)$. Precision indicates the number of positives correctly classified by the classifier.

Recall: Recall is defined as the True Positive Rate of a classifier and given by $true\ positives/(true\ positives + false\ negatives)$. Recall indicates the number of positives detected by the classifier of all the positives in the dataset. The results show that ResNet-101 lends itself best to the task of classification achieving an overall accuracy of 92% while VGG-16 and VGG-19 lag at 89% and 86% respectively. However, the performance of all the models is sub-par in comparison to the accuracy of the validation set during training.

	VGG-16			**VGG-19**			**ResNet-101**		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Canola	0.91	0.76	0.79	0.86	0.67	0.67	0.91	0.78	0.78
Rough Rice	0.91	0.78	0.78	0.90	0.77	0.80	0.93	0.81	0.87
Sorghum	0.90	0.70	0.77	0.75	0.66	0.60	0.91	0.78	0.77
Soy	0.90	0.75	0.80	0.90	0.74	0.77	0.93	0.88	0.81
Wheat	0.84	0.78	0.76	0.89	0.73	0.75	0.94	0.85	0.87
Overall	0.89			0.86			0.92		

Table 2: Evaluation Results on the Fine-Tuned Models of VGG-16, VGG-19 and ResNet-101

3.5 Ensemble Model

Ensemble Learning refers to the generation and combination of multiple inducers to solve a machine learning task [32]. To better the classification performance, an ensemble whose predictions are based on the individual predictions made by each of the models is developed. The models are referred to as base estimators. The ensemble model works by summing the categorical softmax outputs of each of the models for a given sample. The prediction made by the model is the largest resultant categorical value. Ensemble models are used to overcome the challenges faced while building a single estimator such as high variance, low accuracy, and noise. In a single estimator, the model is very sensitive to the provided inputs when trying to learn the features resulting in high variance. Since the model uses a single estimator, a single algorithm is used to fit the entire training data which often fails to produce the required output. Also, the model heavily relies on one or a few features while making predictions. Thus, a single algorithm may not make an accurate prediction for a given dataset. Every machine learning algorithm has its limitations and producing a model with high accuracy may be challenging. Building a model by combining multiple models might boost the overall accuracy. An ensemble model can be built by aggregating the output from each model with two objectives: reducing the model error and maintaining its generalization. The workflow of the ensemble model is as shown in Fig. 10 and the results obtained are as shown in table 3. The results show an improvement in overall accuracy to 94.6% showing a 2.6% improvement in accuracy compared to the best individual model of ResNet-101. Besides, a significant improvement in the precision and recall for each of the individual classes is also observed which indicates better quality of predictions.

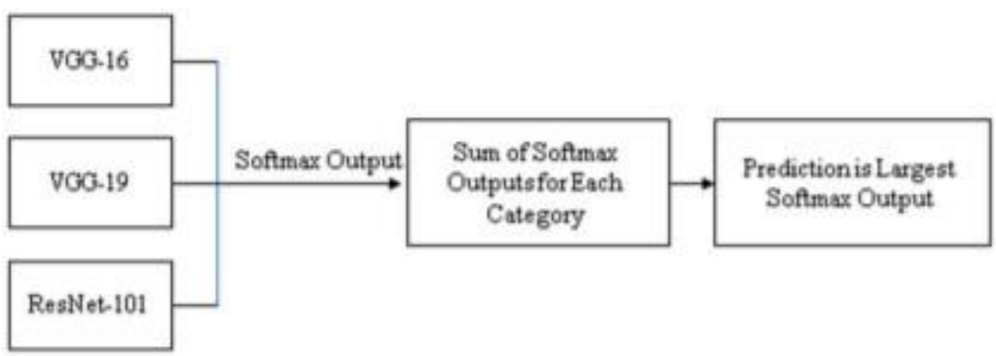

Figure 12: Workflow of Ensemble Model

ENSEMBLE MODEL			
	Accuracy	Precision	Recall
Canola	0.94	0.84	0.90
Rough Rice	0.95	0.88	0.89
Sorghum	0.94	0.91	0.82
Soy	0.95	0.83	0.96
Wheat	0.95	0.88	0.90
Overall	0.94		

Table 3: Evaluation Results on Ensemble Model

Chapter 4 - Conclusion and future work

A seed classification framework is demonstrated for seed phenotyping using deep learning techniques and synthetic image datasets. A prototypical seed phenotyping environment is used to demonstrate the feasibility of low-altitude UAV imagery and synthetic datasets to address the profusion of training data required to train neural network architectures. The use of transfer learning and fine-tuning is employed to efficiently conduct the task of classification. Besides, an ensemble model is built to improve upon the quality of predictions made by the neural networks. The next steps include leveraging a drone that is sturdy enough for stable outdoor flights and testing the framework in an outdoor environment where the acquired drone imagery is complex to work with due to varying backgrounds. Also, the impact of drone image resolution and quality on the generated synthetic image datasets is to be studied.

Currently, for the plot estimation, the AR Parrot Drone's node libraries namely node AR drone is being leveraged along with REPL to steer the drone around and capture images. Further steps would include building a UI (user interface) using React and Express JS that would have options such as right, left, front, back, start and stop buttons. The start and stop buttons functionality would be to take off and land the drone whereas the right, left, front, and back would be to move the drone around in different directions.

The python and JavaScript code used for the thesis can be found at:

https://github.com/Niketa912-P

Bibliography

[1] Margapuri, V., Penumajji, N., & Neilsen, M. Seed Classification using Synthetic Image Datasets Generated from Low-Altitude UAV Imagery.

[2] Y. Zhu et al., “Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning,” 2016.

[3] D. S. Chaplot, E. Parisotto, and R. Salakhutdinov, “Active Neural Localization,” pp. 1–15, 2018.

[4] S. Ross et al., “Learning monocular reactive UAV control in cluttered natural environments,” Proc. - IEEE Int. Conf. Robot. Autom., pp. 1765–1772, 2013.

[5] S. Ross, G. Gordon, and D. Bagnell, “A reduction of imitation learning and structured prediction to no-regret online learning,” in Proceedings of the fourteenth international conference on artificial intelligence and statistics, 2011, pp. 627–635.

[6] Margapuri, V., & Neilsen, M. (2021). Classification of Seeds using Domain Randomization on Self-Supervised Learning Frameworks. *arXiv preprint arXiv:2103.15578*.

[7] A. Giusti et al., “A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots,” IEEE Robot. Autom. Lett., vol. 1, no. 2, pp. 661–667, 2016.

[8] N. Smolyanskiy, A. Kamenev, J. Smith, and S. Birchfield, “Toward low-flying autonomous MAV trail navigation using deep neural networks for environmental awareness,” IEEE Int. Conf. Intell. Robot. Syst., vol. 2017–Septe, pp. 4241–4247, 2017.

[9] Toda, Y., Okura, F., Ito, J., Okada, S., Kinoshita, T., Tsuji, H., & Saisho, D. (2020). Training instance segmentation neural network with synthetic datasets for crop seed phenotyping. Communications biology, 3(1), 1-12.

[10] Ward, D., Moghadam, P., & Hudson, N. (2018). Deep leaf segmentation using synthetic data. arXiv preprint arXiv:1807.10931.

[11] Gulzar, Y., Hamid, Y., Soomro, A. B., Alwan, A. A., & Journaux, L. (2020). A Convolution Neural Network-Based Seed Classification System. Symmetry, 12(12), 2018.

[12] Buters, T., Belton, D., & Cross, A. (2019). Seed and seedling detection using unmanned aerial vehicles and automated image classification in the monitoring of ecological recovery. Drones, 3(3), 53.

[13] Margapuri, V., Courtney, C., & Neilsen, M. (2021). Image Processing for High-Throughput Phenotyping of Seeds. *EPiC Series in Computing*, *75*, 69-79.

[14] Mukti, I. Z., & Biswas, D. (2019, December). Transfer learning based plant diseases detection using ResNet50. In 2019 4th International Conference on Electrical Information and Communication Technology (EICT) (pp. 1-6). IEEE.

[15] Muneer, A., & Fati, S. M. (2020). Efficient and Automated Herbs Classification Approach Based on Shape and Texture Features using Deep Learning. IEEE Access, 8, 196747-196764.

[16] Bah, M. D., Dericquebourg, E., Hafiane, A., & Canals, R. (2018, July). Deep learning based classification system for identifying weeds using high-resolution UAV imagery. In Science and Information Conference (pp. 176-187). Springer, Cham.

[17] Bristeau, P. J., Callou, F., Vissière, D., & Petit, N. (2011). The navigation and control technology inside the ar. drone micro uav. IFAC Proceedings Volumes, 44(1), 1477-1484.

[18] Margapuri, V., & Neilsen, M. (2021). Seed Phenotyping on Neural Networks using Domain Randomization and Transfer Learning. In *2021 ASABE Annual International Virtual Meeting* (p. 1). American Society of Agricultural and Biological Engineers.

[19] https://www.oreilly.com/content/how-to-build-an-autonomous-voice-controlled-face-recognizingdrone-for-200/

[20] https://www.instructables.com/Autonomous-AR-Parrot-Drone-20-Flying/

[21] Margapuri, V., Lavezzi, G., Stewart, R., & Wagner, D. (2020). Bombus Species Image Classification. *arXiv preprint arXiv:2006.11374*.

[22] https://learn.droneblocks.io/courses/191196/lectures/2908697

[23] Atabansi, C. C., Chen, T., Cao, R., & Xu, X. (2021, April). Transfer Learning Technique with VGG-16 for Near-Infrared Facial Expression Recognition. In Journal of Physics: Conference Series (Vol. 1873, No. 1, p. 012033). IOP Publishing.

[24] Margapuri, V. (2020). Smart Motion Detection System using Raspberry Pi. *arXiv preprint arXiv:2006.06442*.

[25] Mao, J., Tran, D., Ma, W., Naumovski, N., Kellett, J., & Slattery, A. (2020, December). Application of Deep Learning in Automated Meal Recognition. In 2020 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 58-63). IEEE.

[26] Mateen, M., Wen, J., Song, S., & Huang, Z. (2019). Fundus image classification using VGG-19 architecture with PCA and SVD. Symmetry, 11(1), 1.

[27] Mnih, V., & Hinton, G. E. (2010, September). Learning to detect roads in high-resolution aerial images. In European Conference on Computer Vision (pp. 210-223). Springer, Berlin, Heidelberg

[28] Coulibaly, S., Kamsu-Foguem, B., Kamissoko, D., & Traore, D. (2019). Deep neural networks with transfer learning in millet crop images. Computers in Industry, 108, 115-120.

[29] K. Kelchtermans and T. Tuytelaars, “How hard is it to cross the room? -- Training (Recurrent) Neural Networks to steer a UAV,” 2017.

[30] Margapuri, V., & Neilsen, M. (2019). Prototype Implementation of Temperature Control System with CAN and FreeRTOS on STM32F407 Discovery Boards. *EPiC Series in Computing*, *63*, 130-139.

[31] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, "Learning Deep Control Policies for Autonomous Aerial Vehicles with MPC-Guided Policy Search," 2015.

[32] Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1249