Image Classification with a Deep Learning Model for classifying Maltese endemic and non-endemic flowering plants.

Justin Borg, Darren Scerri
Institute of Information & Communication Technology
Malta College or Arts, Science & Technology
Corradino Hill
Paola PLA 9032
{justin.borg.e21729, daren.scerri}@mcast.edu.mt

***Abstract*—Nowadays flower classification in Malta and around the world has increasingly become crucial for conservation efforts, medicinal research, botanical work and many other fields which rely on flower classification. This paper presents an approach of using Computer Vision to conduct flower image classification using a deep learning model specifically made to classify eight different types of flowering plants, including the Cheirolophus Crassifolius plant also known as the Maltese Centaury, Malta's national plant. The process consists of using a Convolutional Neural Network (CNN) for image classification purposes, for the model's architecture we followed the Keras Sequential model, alongside other popular image classification techniques, all of which is running on the TensorFlow platform. The proposed model managed to achieve its highest score result of 80% Validation accuracy and a class confidence overall result of 78.79%.**

***Index Terms*—Computer Vision, Flower Image Classification, Deep Learning Model, Convolutional Neural Network (CNN), Keras Sequential, TensorFlow.**

I. Introduction

As one of the more important species in the world, flowering plants are a source of life and food for birds, insects, animals and also humans. There exist around 369000 different species of flowering plants all across the world [1], and around 1100 species in Malta. Therefore flower classification is a highly tedious concept given its complexity. Computer vision technologies are nowadays being coupled with deep learning models, to carry out image classification in order to recognize and classify all sorts of things, including flowering plants. In this study we train and test a deep learning model to classify non-endemic flowering plants, and also the Maltese endemic flowering plant Cheirolophus Crassifolius also known as the Maltese Centaury flower.

The purpose of this study is to investigate the use of computer vision, image classification and a deep learning model in aid to automate the process of classifying flowering plants. The main dependant variable in this study is the accuracy and prediction confidence of the deep learning model in classifying the various flowering plants, with the independent variables being the kind of deep learning model used, the image data (dataset) and the data augmentation preprocessing techniques.

This paper will focus on using this technology in attempt to classify a variety of flowering plants. Hence the hypothesis of this study is that image classification with deep learning models can be used to classify Maltese endemic and non-endemic flowering plants. Given the above hypothesis two research questions were created for this study. *"How is the accuracy of image classification affected in deep learning models when using different types of data augmentation pre-processing layers?"* and subsequently *"How effective is image classification in classifying Maltese endemic and non-endemic flowering plants?"*.

II. Literature Review

A. Overview of Computer Vision and Image Recognition

Computer Vision is a powerful tool comprised of a combination of image processing and pattern recognition, it can be deduced that the output of the process of Computer Vision, is image understanding [2]. The extensive usage of Computer Vision and Image Processing has lead to a large attraction from scholars utilising this technology in a large scale of different applied areas [2]. The most evident example for the use of Image Classification is Facebook. This world-renowned application is able to recognize a human's face with only a few labelled images, with approximately 98% accuracy [3].

B. Overview of Image Classification within Deep Learning models

The inspiration for Deep Learning Algorithms comes from the basic cognitive system of the human body, a system which is capable of learning complex data representations [4]. Keras, an open-source, high-level deep learning API has become an industry standard for building intelligent deep learning architectures. TensorFlow is a back-end framework or engine on which neural network architectures perform on [4]. Building a neural network for the purpose of Image Classification is typically made up of a number of steps, the process starts with gathering a large dataset of images, next an architecture is chosen, a Convolutional Neural Network (CNN) is typically used for Image Classification as according to [5], the CNN architecture has strong plausible evidence, furthermore it was also stated that in a study where traffic sign detection was being conducted, the *"CNN-based system has even surpassed human capability in benchmarking tests"*. The model is then initialised, optimized and trained with the dataset. Lastly the model is tested with test data and its performance is reviewed [1].

C. Flower Classification

The process of classifying and arranging flowers according to their unique characteristics is known as flower classification. In order to identify a plant's taxonomic groupings, a variety of characteristics including petal arrangement, color, shape, size, and overall structure are examined. With the shortcoming of expert taxonomy (the science of classifying organisms), automated recognition and classification has provided a great alternative to human dependant taxonomy [1]. Furthermore, with such a wide variety of flower plant species that exist and are always being discovered, there is always limitations that come with human classification.

D. Exploring the use of Image Classification to classify flowering plants

The application of Image Recognition in Flower Classification proves to be important due to its high range of applicability in various domains such as, Precision Agriculture, Botanical Research, Plant Monitoring, Gardening, Floriculture, Ayurvadic Treatment [1] [6].

Authors in [1] proposed a deep learning based method in conjunction with a transfer learning model for Flower Classification. A study was carried out where they tested on a public Kaggle flower dataset using a CNN model architecture. As shown in Figure. 1, the proposed model had an input image size of 224x224 with four convolutional layers, max pooling was used in all layers and finally the model was given a flattening and a dense layer [1].

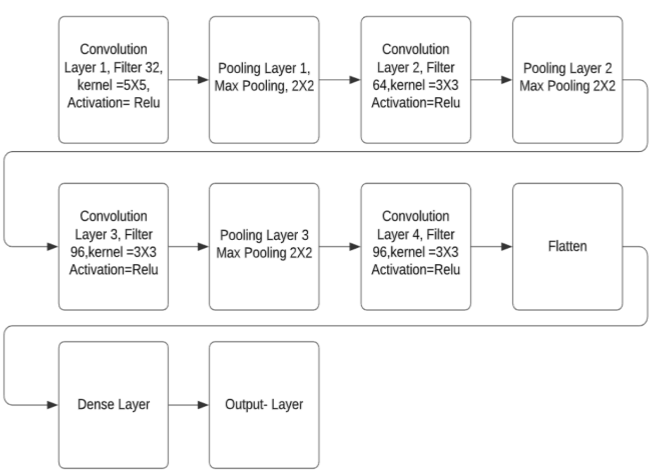

Fig. 1. Proposed CNN Framework

The results were then compared with three other models as shown in Figure. 2. This study concluded that their proposed model achieved very positive Validation and Testing results and hence it was observed that Artificial Intelligence (AI) and Machine Learning (ML) technologies have *"potential to benefit agriculture idustry greatly with automation, it can be used in marketing agricultural product"*.

TABLE- 1 RESULT SUMMARY

Sr. no.	Model	Validation Accuracy	Testing Accuracy
1	Proposed CNN Framework	95%	91%
2	Transfer learning VGG16	95.39	89.35%
3	Transfer learning Resnet50	97.07%	92.12%
4	Transfer learning MobileNetV2	91.88%	71.75%

Fig. 2. Results

In another study conducted by [7], the authors accentuated that since Neural Network (NN) *"was designed to solve complex problems such as pattern recognition and classification, therefore NN plays a significant role in classification process."*. Hence, it is able to classify complex datasets such as flower datasets. This study aimed to address the use of NN and image processing for the understanding of flower image features, in particular the Malaysian Blooming Flower. This model focused heavily on the Data Preparation phase, the flower images that made up the dataset was captured based on [8] methodology, this method focuses on capturing the images with the flower being centered and focused, with a defocused background. These images were then Filtered for any errors caused by noise, and segmented in order to acquire the colour in the regions of interest. Image Thresholding and Region Filling, followed as seen in Figure. 3.

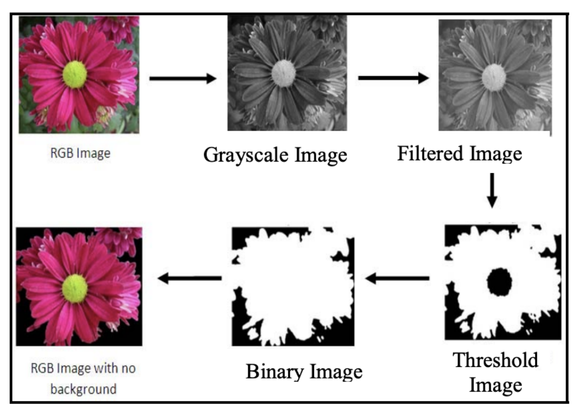

Fig. 3. Data Preparation

This study obtained positive results and concluded that NN are a factual potential in building a Malaysian Blooming Flower model and that their model can be used in the future to develop a Malaysian Blooming Flower recognition system [7].

In the paper by [9], the authors conducted a study on Flower Image Recognition by using a system that uses edge and color characteristics of a flower image in order to classify them. Edge characteristics are applied through Hu's seven moment algorithm. Basic Red, Green, Blue (RGB), Hue and Saturation characteristics are obtained from histograms and the K-nearest neighbour algorithm is used for the Flower classification itself [9]. The output of this system returns as the top three most similar flower images. This system managed to achieve an accuracy greater than 80% [9].

III. Research Methodology

In this initial research we explore the use of image classification techniques and a deep learning algorithm to classify different species of flowering plants, including the Cheirolophus Crassifolius a Maltese Endemic plant and several other non-endemic flowering plants. The Pipeline in Figure. 4 outlines how this objective is to be attained.

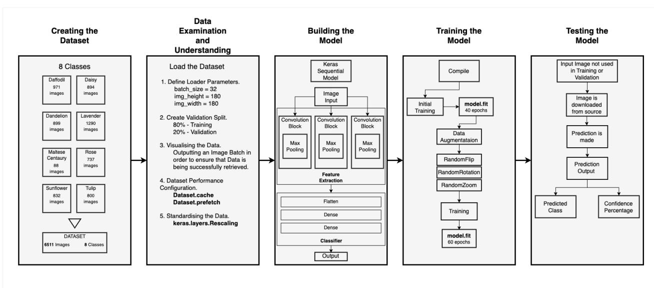

Fig. 4. Research Pipeline

A. Creating the Dataset

In the first phase of this prototype a dataset was created by combining several publicly available flowering plants datasets. The finished dataset was made up of eight classes meaning eight different flowering plant species and with a total of 6511 images. In order to ensure fair and equal results it was important to have similar image amounts for each class. As shows in Figure. 5 the classes have a similar image data count with the exception of the Maltese Centaury flower (Cheirolophus Crassifolius). This can be considered as a research limitation as there are currently few publicly available images of this Maltese endemic flower.

```
- Daffodil (971 images)
- Daisy (894 images)
- Dandelion (899 images)
- Lavander (1290 images)
- Maltese Centaury (88 images)
- Rose (737 images)
- Sunflower (832 images)
- Tulip (800 images)
```

Fig. 5. Dataset

B. Data Preperation

The next phase of the study was to retrieve data from the dataset, understand it and configure it so that it can be manipulated in the next phase. The dataset is firstly downloaded and made locally available. Some small tests were then run to ensure that the dataset is being recognized and successfully retrieved by the model. Parameters are then set to the dataset to configure the loader, the batch size is set to 32 and the image width and height is defined to 180x180 pixels. A validation split was then set to the database, an 80% training and 20% validation method approach was taken. This means that the model used 80% of a classes images for its training and the other 20% for validation purposes. As can be observed in Figure. 6, the model at this stage was able to retrieve and split the dataset. Next the model outputs a batch with 9 images in order to ensure image visualisation as shown in Figure. 7. The final step in data preparation was to utilise buffered prefetching, this ensures that yielding of data is done from disk without having I/O blocking. This was executed by using the TensorFlow *Dataset.cache* method, this method is able to retain the images stored in memory after they have been loaded off the disk during the first epoch, this method is useful as it eliminates bottleneck possibilities. TensorFlow's *Dataset.prefetch* method was also used for overlapping data preprocessing and model execution while training.

```
Found 6511 files belonging to 8 classes.
Using 5209 files for training.
Found 6511 files belonging to 8 classes.
Using 1302 files for validation.
['daffodil', 'daisy', 'dandelion', 'lavender', 'malteseCentaury', 'rose', 'sunflower', 'tulip']
```

Fig. 6. Validation Split

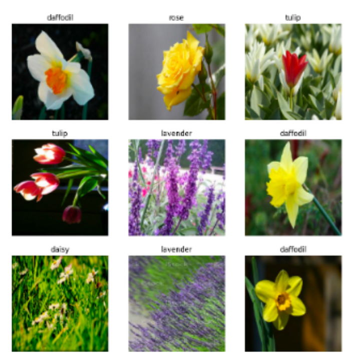

Fig. 7. Data Visualisation

C. Building the Model

In this study the approach of building a Convolutional Neural Network (CNN) was taken in order to conduct Flower Image Classification. In particular we used the Keras Sequential model to build our CNN, which is running on top of the TensorFlow platform. This model consists of three convolutional blocks, each containing a max pooling layer. As depicted by the Research Pipeline, the process is first to carry out Feature Extraction through the three convolutional blocks and then Classification through a Flatten and two Dense Classifier layers. The fully-connected 128 unit Dense layer is activated by the Rectified Linear Unit (ReLu) activation function. The ReLu function has become a widely used activation function for many types of neural networks, it tends to facilitate the training process of a model and often also achieves better performance results [10].

D. Training the Model

The model was then compiled by using the Keras Adam algorithm optimizer and the SparseCategoricalCrossentropy loss function, this function determines the crossentropy loss between the labels and the predictions. The process of training the model started by assigning 40 epochs (data iterations) for training with the *Keras.fit* method, this method simply trains the model for a fixed given amount of epochs. This process took around 45 minutes to finish and after this process was done, the model outputted Training and Validation Accuracy and Loss graphs.

Overfitting presented itself as being an issue in this phase of the model. Overfitting occurs when there isn't a large enough number of training samples hence the model tends to learn form unwanted noise and details from training examples, which results in loss of accuracy. To combat this, we added Data augmentation to the model. This approach generates additional training data from the current present examples by enhancing them with randomised transformations that produce convincing images. The Data Augmentation approach was divided into two iterations. In the first iteration the *layers.RandomFlip* preprocessing layer was implemented whereas in the second iteration, we additionally implemented another two preprocessing layers, the *layers.RandomRotation* and the *layers.RandomZoom*. A new neural network was created, and a final additional Dropout layer technique was implemented to it. This technique randomly sets the activation to a value of 0 which furthermore assists against overfitting. The model was then compiled and trained with a higher value of 60 epochs. This process took approximately 1 hour and 15 minutes to be completed.

E. Testing the Model

The trained model was then tested with new Flower images which were not included in either the training or the validation images. A testing image URL path was provided to the model with a few specifications such as image height and width. The model initiates testing by first downloading the provided image through URL. The model then outputs its prediction by using TensorFlow's *model.predict*. The model next outputs the percentage of confidence at which this prediction was made, as shown in Figure. 8.

```
Downloading data from http://i.pinimg.com/originals/b9/e3/28/b9e32847549f86d7516e24f0784c6ac2.jpg
29661/29661 [==============================] - 0s 0us/step
1/1 [==============================] - 0s 29ms/step
This image most likely belongs to dandelion with a 91.37 percent confidence.
A dandelion is a flowering plant with yellow petals and fluffy white seed heads that disperse in the wind.
```

Fig. 8. Prediction

F. Technologies Used

The hardware technologies that were used for conducting this experiment were a laptop with a CPU specification of 2.2GHz 6-Core Intel Core i7, Intel UHD Graphics 630 and 16GB of DDR4 Memory. In this study the laptop's hardware specification did not directly affect the accuracy of the results as the number of data iterations (epochs) wasn't very high and the same algorithm was used all throughout the study. Therefore although better hardware would have made the training process faster, the accuracy would not have been affected. This study also required a good internet connection since the dataset and testing images required to be downloaded from the internet, before being locally accessed.

This study used a number of publicly available software technologies and libraries to build and train the model and afterwards conduct image classification on flowering plant images. Python 3.11.1 was used as a programming language and Visual Studio Code was used as the code editor. The TensorFlow machine-learning platform was used to develop the neural network. TensorFlow also proves to be a very good choice when working with python language, as they integrate together nicely [11]. The Keras high-level API, in specific the Keras Sequential model was also used, running on top of the TensorFlow platform it serves as our main model structure. The numpy and PIL libraries were also used for building the model.

G. Research Methods

This study was designed to produce quantitative performance metrics from the prototype itself. The prototype was setup so that whilst it is training, testing and classifying, it produces graphs and diagrams with the respective results that it is achieving. These graphs and diagrams will be used to analyse the results that were achieved by the prototype in attempt to conduct Flower Image Classification on Maltese endemic and non-endemic flowering plants. The relevant metrics were Training And Validation Accuracy graphs, Training and Validation Loss graphs, for both iterations. Confidence percentage tables were also extracted from the predicted results, based on the average result for 3 images, within each class.

IV. Findings & Discussion of Results

A. Training and Validation Results

For this study we focused particularly on the results pertaining to the Validation Accuracy as it is the most affected variable when testing data augmentation within image classification.

As shown in Table I, Iteration 1 obtained a 65% Validation accuracy in Phase 1 of the study which contained 40 epochs, this was a significantly low result and therefore we decided to implement the aforementioned data augmentation preprocessing layer *layers.RandomFlip* technique prior to Phase 2 training, which contained 60 epochs. Iteration 1 Phase 2 managed to obtain a significantly better result with 75% Validation accuracy. Furthermore, a second iteration (Iteration

TABLE I
VALIDATION ACCURACY RESULTS

	Iteration 1	Iteration 2
Phase 1 Validation Accuracy	65%	65%
Phase 2 Validation Accuracy	75%	80%

2) was conducted to further research whether adding additional augmentation prepossessing layers would further increase the validation accuracy of the results. Phase 1 of Iteration 2 wasn't any different from the one in Iteration 1, it was the initial training with 40 epochs and was done to simply mirror the first phase of Iteration 1. As expected this resulted into an almost identical Validation accuracy graph as the one in Iteration 1, with an overall of 65%. In Phase 2 of Iteration 2 the *layers.RandomZoom* and *layers.RandomRotation* preprocessing layers were implemented and after 60 epochs of training the model showed a further significant Validation accuracy of 80%.

The Validation Accuracy results were extracted from the graphs shown in Figure. 9. These graphs represent the accuracy or loss plotted against the number of epochs, respectively. In both Iterations in phase 1, it can be seen that the model kept a linear amount of accuracy from start to finish, hence we see a sort of flat Validation accuracy line. This changes drastically in Iteration 2 Phase 2 (bottom right graph) as we see that the Validation accuracy starts low at the first few epochs but increases up to 0.8 (80%) towards the end of the 60 epochs. Furthermore the representation of the training and validation accuracy lines being closer together indicate that there was less overfitting present, in that specific phase.

With these results we deduced and answered the question, that the Validation Accuracy is in fact affected and in this case improved when data augmentation preprocessing layers are added to a deep learning model conducting image classification.

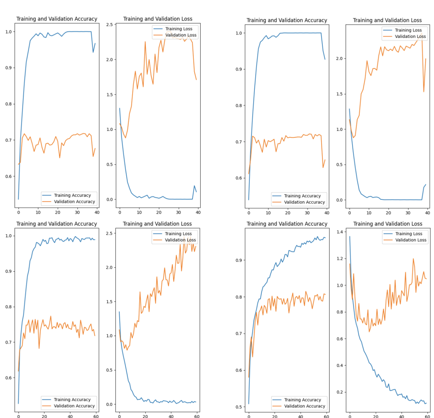

Fig. 9. Accuracy Results

B. Prediction Confidence results

Confidence percentages were extracted from each class, this result came from the prediction that the model made when it was given three new images for each of the eight classes, prediction percentage results can be seen in Figure. 10. In Iteration 1 the model obtained irregular results as classes such as Daisy, Lavender and Rose predictions were very high whereas other classes such as Tulip and Sunflower obtained very poor predictions. Overall we see that the majority of the classes gain a very significant confidence increase in Iteration 2 when compared to Iteration 1, this change occurred because of the addition of the preprocessing layers. In Iteration 2 the results became more regular as there were no drastic differences in the averages for the predictions.

Iteration 1 - Data Augmentation (RandomFlip)

	Daffodil	Daisy	Dandelion	Lavender	Rose	Sunflower	Tulip	Maltese Centaury
Image1	Daffodil 99.47	Daisy 100.00	Lavender 97.61	Lavender 100.00	Rose 100.00	Daffodil 99.74	Dandelion 63.48	Maltese Centaury 99.97
Image2	Daisy 98.82	Daisy 99.68	Dandelion 99.53	Lavender 100.00	Rose 68.96	Sunflower 100.00	Rose 99.51	Lavender 99.87
Image3	Daffodil 100.00	Daisy 100.00	Dandelion 99.99	Lavender 100.00	Rose 100.00	Dandelion 99.85	Daisy 74.20	Maltese Centaury 99.41
Average (%)	66.49%	99.89%	66.51%	100.00%	89.65%	33.33%	0.00%	66.46%
Overall %	65.29%							

Iteration 2 - Data Augmentation (RandomFlip, RanadomZoom and RandomRotation)

	Daffodil	Daisy	Dandelion	Lavender	Rose	Sunflower	Tulip	Maltese Centaury
Image1	Daffodil 100.00	Daisy 100.00	Dandelion 90.68	Lavender 100.00	Rose 100.00	Tulip 57.28	Tulip 57.28	Maltese Centaury 99.69
Image2	Daisy 99.12	Daisy 99.82	Dandelion 91.37	Lavender 83.88	Rose 98.38	Sunflower 99.99	Rose 50.87	Rose 75.58
Image3	Daffodil 77.21	Daisy 100.00	Dandelion 88.77	Lavender 98.33	Rose 100.00	Sunflower 73.56	Tulip 99.88	Maltese Centaury 99.90
Average (%)	59.07%	99.94%	90.27%	94.07%	99.46%	57.85%	63.10%	66.53%
Overall %	78.79%							

Fig. 10. Prediction Results

Another significant finding was that the Maltese Centuary flower managed to obtain a prediction average of 66.50% while only having had 88 images in its dataset class, which is significantly lower than a class such as the Tulip which had 800 images in its dataset class but only obtained a prediction average of 31.55%. This can hinder the thought if in Flower image recognition specifically with a Sequential model, the size of the dataset is less significant than the quality of the given images. We deduced that the Maltese Centaury flower managed to obtain such result with a very small dataset class because it is a very unique looking flower, therefore the model was able to recognize and classify it more effectively. This theory can also be supported when we further analyse the table results in Figure 10. The model made wrong predictions of flowers which share characteristics such as the Daffodil the Daisy and the Sunflower, hence why we conclude that the flower classification is mostly effective on flowers with unique characteristics such as the Maltese Centaury flower.

The overall average percentage which represents confidence was calculated for each class, this was done by adding up all the three percentages and diving by the amount of correct classification prediction, i.e. in Iteration 1 the Daffodil class obtained only two correct classifications, as the second image was predicted to be a Daisy which was incorrect therefore was considered as a 0% prediction value. Overall percentages were further calculated by adding up all the average percentages of the classes and dividing by the number of total classes. Iteration 1 obtained a overall of 65.29% whereas Iteration 2 managed to obtain 78.79%, meaning there was a delta of 13.50% between the two iterations. This further proves that the addition of data augmentation preprocessing layers also obtained significantly better predictions and confidence percentages.

With these results we can say that image classification in classifying Maltese endemic and non-endemic flowering plants is feasible and effective with a 78% confidence rate. This initial prototype model could furthermore be optimized in order to obtain a higher confidence percentage but this prototype serves as a good proof of concept for classifying Maltese endemic and non-endemic flowering plants.

C. Comparison of Results

The authors in [1] who conducted a similar study, trained and tested a model on the Kaggle flower dataset. Their dataset contained 4323 images with a validation split of 3890 images for training and 433 for validation. This dataset is significantly smaller than the one used in our study which contained 6511 images with a validation split of 5209 images for training and 1302 for validation. This study used an image size of 224x224 pixels compared to our 180x180 pixels. Similarly this study made use of a CNN framework, although this study's model consisted of four convolutional layers instead of three. This study also made use of the ReLu activation function that was used in our study as well. Flattening and Dense layers were also similarly used. The architecture used by this study's model can be seen in Figure. 11.

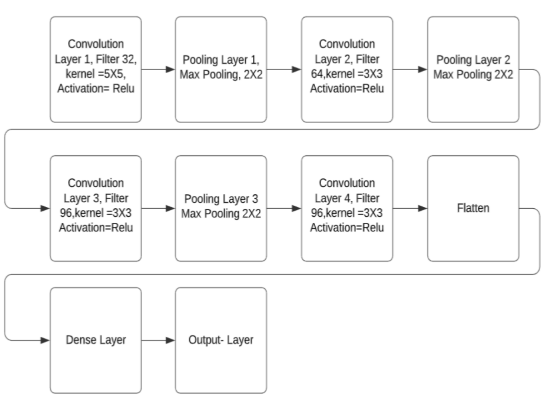

Fig. 11. Four-layer Framework

With this we deduce that both models used very similar architecture frameworks and libraries. The model in this study managed to obtain high Validation accuracy results, the highest being 95%. It can be concluded that the study conducted by [1] managed to obtain higher results than ours because of a number of different factors, mainly being the use of pre-trained weights (ImageNet) and furthermore the use of a four convolutional layer architecture, as opposed to a three convolutional layer architecture used in our study, while also not condemning the possibility of the quality of the dataset used being superior to ours.

V. Conclusion

This study sought to develop and test a deep learning based model to classify flowering plants. Image classification along with deep learning models can be used to classify Maltese endemic and non-endemic flowering plants, the results indicate that this hypothesis holds true. Validation accuracy and Prediction confidence results demonstrate the model's ability to be able to classify various flowering plants. Furthermore, the results in this study indicate how the effectiveness of the model is enhanced when preprocessing layers are added to the data augmentation phases in the model. These results address both research questions as specified before.

While this performance is promising as a proof-of-concept, it may not yet possess the sophisticated requirements for it to be used in more complex applications within the field of flower classification.

A. Limitations and Further Implementation

This study also encountered some limitations mainly relating to the dataset, as the publicly available datasets on flowering plants was restricted, especially the one pertaining to the Maltese Centaury flower, hence why there was an imbalance in the amount of images this class had, compared to the other classes. Documentation and literature on this topic and the use of deep learning models for the purpose of flower classification was also not very robust.

Further research and implementation to the prototype can certainly make this prototype more accurate and achieve better results. For this to be achieved, a larger dataset which uses more precise photography techniques to obtain more centered and focused flower images, would be needed. Further implementation towards data augmentation could also eliminate completely the overfitting issue present in the current version of the prototype.

References

[1] C. Narvekar and M. Rao, "Flower classification using cnn and transfer learning in cnn- agriculture perspective," in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, 2020, pp. 660–664.

[2] V. Wiley and T. Lucas, "Computer vision and image processing: A paper review," *International Journal of Artificial Intelligence Research*, vol. 2, p. 22, 02 2018.

[3] J. Kothari, "A case study of image classification based on deep learning using tensorflow," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 6, pp. 3888–3892, 04 2018.

[4] K. Ramasubramanian, A. Singh, K. Ramasubramanian, and A. Singh, "Deep learning using keras and tensorflow," *Machine Learning Using R: With Time Series and Industry-Based Use Cases in R*, pp. 667–688, 2019.
[5] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen, "Medical image classification with convolutional neural network," in *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, 2014, pp. 844–848.
[6] "Literature review on flower classification," *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)*, vol. 04, no. 02, 2015.
[7] F. Siraj, M. A. Salahuddin, and S. A. M. Yusof, "Digital image classification for malaysian blooming flower," in *2010 Second International Conference on Computational Intelligence, Modelling and Simulation*, 2010, pp. 33–38.
[8] T. Saitoh, K. Aoki, and T. Kaneko, "Automatic recognition of blooming flowers," vol. 1, 09 2004, pp. 27 – 30 Vol.1.
[9] T. Tiay, P. Benyaphaichit, and P. Riyamongkol, "Flower recognition system based on image processing," *Proceedings of the 2014 3rd ICT International Senior Project Conference, ICT-ISPC 2014*, pp. 99–102, 10 2014.
[10] J. Brownlee, "A gentle introduction to the rectified linear unit (relu)," Aug 2020. [Online]. Available: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/
[11] S. Yegulalp, "What is tensorflow? the machine learning library explained," Jun 2022. [Online]. Available: https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html