Canadian Medicinal Plant Classification using Convolutional Neural Network and Transfer Learning

Shaon Bhatta Shuvo
School of Computer Science
University of Windsor
Windsor, Canada
shuvos@uwindsor.ca

Akram Vasighizaker
School of Computer Science
University of Windsor
Windsor, Canada
vasighi@uwindsor.ca

Akshay Shah
School of Computer Science
University of Windsor
Windsor, Canada
shah1bz@uwindsor.ca

Maryam MahdavyRad
School of Computer Science
University of Windsor
Windsor, Canada
mahdavy@uwindsor.ca

Malita Dodti
School of Computer Science
University of Windsor
Windsor, Canada
dodti@uwindsor.ca

***Abstract*—Nowadays, computerized plant species classification systems are used to help the people in the detection of the various species. However, the automated analysis of plant species is challenging as compared to human interpretation. This research as been provided in this field for the better classification of plant species. Even now, these methodologies lack an exact classification of the plant species. The challenge is due to the inappropriate classification algorithm. In Particular, when we consider the medicinal plant species recognition, the accuracy will be the main criteria. In this research, the suggested system implements the deep learning technique to obtain high accuracy in the classification process using computer prediction methods.The Convolutional Neural Network (CNN) is employed beside transfer learning for deep learning of medicinal plant images. This research work has been carried out on the flower images dataset of four Canadian medical plants; namely, Clubmoss, Dandelion, Lobelia, and Bloodroot, which is fed as the training dataset for the CNN and machine learning-based proposed system. Finally, an accuracy of 96% has been achieved in classification of the medicinal plant species.**

***Index Terms*—Image Classification, Medicinal Plant classification, Convolutional Neural Network, Transfer Learning**

I. INTRODUCTION

A medicinal plant is delineated as a plant that is collected from the wild or planted plant for its medicinal value. Plants have been utilized in curing human diseases for thousands of centuries and are the source of a significant percentage of medicines. Canadian medicinal plants have a prolonged history of consumption with hundreds of classes being employed by First Nations Canadians in their old-style medicine [1]. In recent years, some computational approaches have been introduced, particularly in image processing domain, for plant classification. In this regard, Neural Networks represent novel techniques for image processing, with large potentials. The most commonly used Neutral Networks for image processing is Convolutional Neural Network (CNN).

The rest of the paper are distributed as follows: In section 2, we gave a brief description of the motivation of our work. In section 3, we discussed some of the remarkable researches related to our work. Data collection procedure has been discussed in section 4. In section 5, we described the background study including the CNN, transfer learning, and detailed implementation of our model including the data prepossessing. Results are discussed discussed in sections 6. Finally, conclusion and future works are discussed in section 7.

II. MOTIVATION

Medicinal plants have been utilized in curing human diseases for thousands of centuries and are the source of a significant percentage of medicines. Canadian medicinal plants have a prolonged history of consumption with hundreds of classes being employed by First Nations Canadians in their old-style medicine. Medicinal plant species classification is critical for medicine production and conservation. Local peoples are not enough knowledgeable of their urban medicinal plants and their usages. Therefore, classifying the Canadian medicinal plant image using Convolutional Neural Network by high accuracy image classification model could be useful to identify different types of species.

III. LITERATURE REVIEW

There is no particular research has been done to detect medicinal plants of Canada using image classification technique with the best of our knowledge. However, there are different proposed techniques to classify objects or flowers, and some of them employed deep learning approaches. In most researches, classification challenges are deal with using CNN-based algorithms. The Convolutional Neural Networks (CNN) shows considerable accomplishment in different research and real-world projects. One of the earliest applications of CNN was handwritten recognition [2]. By developing of CNN technique, innovative models, which include new layers, are presented. In ImageNet challenges, CNN has been utilized mostly with different datasets mixtures. Some researchers compared human detection abilities and a trained network on image recognition. The output of these comparisons depicts that humans can detect an image with accuracy about 73.1% while the trained network shows up 64% accuracy. Also, after employing Convolutional Neural Networks to a similar dataset, it revealed the 74.9% accuracy, so it depicts better results than human recognition [3].

Then, in a study, a deep CNN-based framework of a hierarchical structure was presented that applies a transfer learning method to modify a deep CNN model. His conclusions indicate that this approach can efficiently improve classification accuracy [4]. On the other research, a CNN method for the flower classification problem was presented. Their conclusions demonstrate the accuracy of 84.02% [5]. Additionally, a CNN framework is presented for plant classification. This framework was presented for the classification of several varieties of plants from the image database collected from intelligent agriculture stations. In this research, a CNN-based architecture is employed for feature extraction of various plant images, which is utilized based on the TARBIL database and achieved 97.47% accuracy on 16 various plant types. This research shows that the CNN-based classification result in more accuracy compares to the SVM-based classification [6]. In another study, a leaf classification framework has been proposed by applying the dual-path deep CNN. This method contains two main functions, firstly the shape and texture attributes are analyzed; secondly, the found attributes are optimized for the classification. This method shows high accuracy in classification of about 99.28% on the Flavia dataset [7].

Ghazi et al. [4] applied transfer learning over the Life CLEF plant dataset with the help of pre-trained models like AlexNet, GoogleNet, and VGGNet. For all these deep convolutional neural networks, fine-tuning is performed, and various parameters are analyzed after data augmentation. Parameters like batch size and the number of iterations are analyzed. And on a recent research four variety of transfer learning models is compared n four datasets. And the output shows the effectiveness role of transfer learning on increasing the performance of prior plant classification models [8].Also, in recent research for classifying natural images by applying the transfer learning approach, they achieved 99.7% overall accuracy [9]. Then Sun et al. [10] proposed a 26-layer ResNet (Residual Network) model for plant identification. BJFU100 dataset is used, and it consists of 10000 images of 100 ornamental plant species found in Beijing Forestry University campus. For experimental analysis, BJFU100 and Flavia datasets are utilized. In deep residual networks, 18, 26, 34, and 50 layers are considered. Amongst the four sets of layers considered, ResNet26 out-performed the other three models. For experimental training, the learning rate is set to 0.001. Flavia dataset accuracy (99.65%) result is compared with other approaches like Radial Basis Probabilistic Neural Network (RBPNN), Deep Belief Network with dropout (DBN), Support Vector Machine (SVM), andResNet26. ResNet26 architecture produced an accuracy of 91.78% recognition rate for the BJFU100 dataset. Barre et al. [11] developed a LeafNet, a CNN-based plant identification system.The leaflet consisted of five sets of 2 convolutional layers and one max-pooling layer followed by one convolution, one max-pooling layer, and three fully connected layers. The leaflet is tested over Leafsnap, Foliage, and Flavia datasets.

In recent flower classification researches, various neural network classification models are compared; however, their primary focus is on LeNet and AlexNet. Their best result is shown on the AlexNet model, which is implemented with Sigmoid for assigning initial weights [12]. Furthermore, on another very recent research for flower classification, a hybrid method is utilized together with Convolutional Neural Network models and feature selection methods. In this suggested model Convolutional Neural Network is employed for feature extraction. And then, for selecting between achieved features, feature selection methods are utilized. Their classification model achievement completed by the Support Vector Machine (SVM) technique was 98.91% [13]. Likewise, Saini and Khamparia present a plant leaf classification using a deep Convolutional Neural Network method based on a five thousand leaf images of two plant. Their result shows 99.96% accuracy on the training dataset and 99.90% on the test dataset [14]. In other recent research, they utilized VGG19, three layers CNN and five layers CNN network for classification species of succulent plant. This method reaches a high accuracy of 99.77%. Their dataset includes 3632 images, which are ten species of succulent plants and non-succulent plants [15]. Some researchers also used hybrid models to detect local foods [16] and birds [17]. Similarly, in another research, a hybrid method is proposed for plant classification. Their recommended method consists of two parts; applying CNN for feature extraction and then on the second phase train the random forest model. In this work, PlantCLEF 2019 dataset was used for the experimental part. Their tested model produces generally higher accuracy than prior strategies [18]. Moreover, in other similar research, a regional convolution neural network (RCNN) utilized for the detection of plants. They employed a fast RCNN model, which consists of a Convolutional Neural Network for extracting features and support vector machine (SVM) for classification. The plants studied in this research are the medicinal plants that can be displayed in various locations like the Himalayas or can be produced in the local garden [19]. One of another recent main application of transfer learning in classification plants is in smarts farms [20] [21] [22] and plant disease recognition systems [20] [23] [24].

Based on the significant number of the literature survey, it is clear that the reported work on plant classification over Canadian plant species is sparse. Also, numerous research works are carried out using features such as shape, texture, color, morphological, or physiological features. Reported works on plant species classification using deep learning architecture are limited. Hence in this project, an investigation is performed using Convolutional Neural Network in order to achieve a higher plant classification rate.

IV. Data

Numerous medicinal species of woodlands as slow-growing perennials are found near Canada's Waterfall area. In this study, we utilize dataset contains 1805 images, includes approximately 400 images for each species. Four group of species are selected in this study namely Clubmoss, Dandelion, Lobelia, and Bloodroot. We use texture, shape, color,physiological or morphological as the features set of the data. The most important advantageous of clubmoss are its usage to treat kidney and urinary disorders. Moreover, Dandelion is a native plant of Canada and commonly used as a weed, growing at one of the many Canadian camping grounds. Health values dandelion provides is to treat joint complaints, liver disorders, skin conditions, and anemia. Also, Lobelia is local to Prince Edward Island, Nova Scotia, New Brunswick, and parts of Ontario and Quebec. It also widely known as Indian tobacco. It is beneficial mostly for relieving respiratory ailments such as bronchitis and asthma. smoking the leaves or brewing them into a tea are two main usage of it. Furthermore, great Lakes, as well as Nova Scotia and parts of New Brunswick are residence of Bloodroot. It alleviate skin issues and respiratory ailments. Moreover, Bloodroot is toxic and one should use it carefully under the prescription of a qualified physician.

In this study we follow some steps for data collection. i) Articulate the problem; Knowing what one wants to predict helps in deciding the data valuable to collect. Data Exploration in the categories of Classification, Clustering, Regression, and Ranking helps with the decision. ii)Establish Data Collection Mechanism; Process of collecting the Data which can be Automated or Manual based on the requirement. iii)Format Data; File format of the images stored need to be same for maintaining the consistency. iv) Reduce Size; Data need to be collected based on the target needs to be achieved which is critical for our Dataset. v) Complete Data Cleansing; Data with missing, erroneous or fewer representative values is removed to make prediction more accurate.

V. Background Study and Methodology

A. Background Study

1) Convolutional Neural Network (CNN): A Convolutional Neural Network, CNN, is a deep learning architecture [25]. Image classification is one of the problems that a CNN could do and is a trained network that can classify images into one of a thousand pre-determined categories. One can employ a CNN to do image processing including image detection, segmentation, and classification. The main advantage of CNN compared to a typical NN is that it automatically detects the significant features without any supervision. A CNN is consists of various layers that transform an input into the output. The complexity of the learned features increases in every hidden layer. For example, detection simple features are learned in the first hidden layer, like edges, and the detection of more complex shapes in the last one. A CNN model is composed of two main components: the feature extraction part and the classification part (Fig 1).

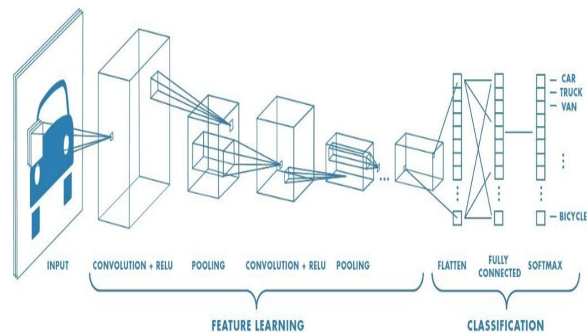

Fig. 1. The Overview of a CNN and its main components [26]

In order to understand a CNN architecture, we introduce several concepts. Compared to a typical neural network in which each input layer's neuron is linked to the hidden layer's neurons. In a CNN we have Local Receptive Fields, a small number of input layer's neurons which are connected to the hidden layer's neurons. The local receptive field use convolution to translate an image into a feature map. Convolution can perform the mathematical convolution operation by moving a filter across the image. At every region, an element-wise matrix multiplication and summation of the result are done. This sum create the feature map, the yellow area in Fig 2.

Non-linearity features make powerful any type of neural network. A neural network can achieve this using an activation function by passing the weighted sum of its inputs to the next layer. CNN use the same function and applies the transformation to the output of each neuron by passing the result of the convolution operation through an activation function. ReLU or Rectified Linear Unit is a popular activation function that maps the output of a neuron to the highest value in the next layer (Fig 3).

A pooling step can use for the dimensionality reduction of the features map by compressing the output of small number of neurons into a one output (Fig 4). We call this a kind of down-sampling the feature map which keeps the important features on CNN automatically. Meanwhile, this can leads to reduction of the number of parameters to learn the model.

A fully connected layer can make it flat the output of the last pooling layer to a 1 dimension vector of values. As it can be seen in Fig 5, between 4 nodes and 5 nodes a fully connected layer is just a dot product of the 1x4 input vector, yellow nodes, with the 4x5 weight matrix W1. The result of this matrix multiplication is a 1x5 vector, shown as the red nodes. We then multiply this 1x5 vector with a 5x5 matrix

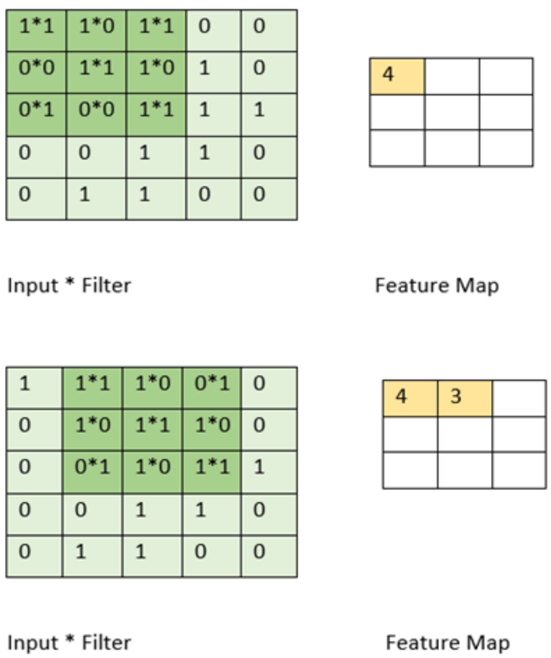

Fig. 2. Creation of feature map by local receptive field translation across an image

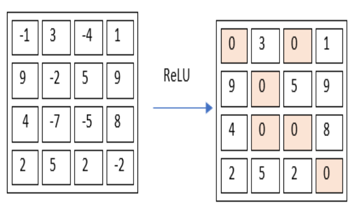

Fig. 3. ReLU activation function maps the output of a neuron to the highest value

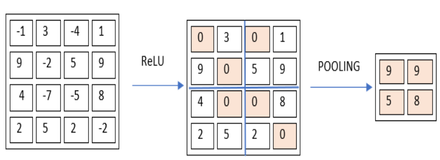

Fig. 4. Pooling layer reduces the dimensionality of the features map

W2, representing in a 1x5 vector, the green nodes. Finally using 5x1 matrix W3 we get the output.

Dropout is the most popular regularization technique, a technique which is used in order to prevent overfitting for deep neural networks. The idea is that during training time, a neuron and all the inputs and outputs to this neuron is temporarily dropped or disabled with the dropout-rate p (Fig 6).

In the case of having a few samples to train, overfitting is unavoidable. But fortunately, data augmentation is a solution to enhance generalization performance on small datasets. This

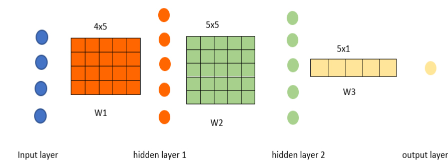

Fig. 5. A fully connected layer is used to flatten the output of the last pooling layer

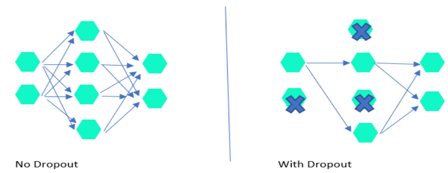

Fig. 6. Dropout regularization technique

is also a regularization technique that is advantageous in some applications in which achieving new training data is difficult. Therefore, we can use data augmentation as a way to enriches or augments by creating new training examples via a random transformation of the current dataset. Common transformations are shifting, rotation, exposure, resizing, contrast change, adjustment, and etc. (Fig 7). It is worth mentioning that, data augmentation should only apply to the training data, but not the validation or test set.

Fig. 7. Data augmentation

2) Transfer Learning: There are some challenges before building the model; small dataset, low-resolution images, lots of noise, and the variations inside the same class images. Deep CNN with transfer learning is the solution to overcome these challenges [27]. The idea is to use the information learned from one specific problem to apply to a comparable one. For example, a CNN model trained to differentiate between animals could be used to train a new model that identify trucks and cars (Fig 8).

The process that takes a pre-trained network, re-training it on a new data, and modifying it is called Transfer Learning. Based on the definition: Model A is successfully trained to solve source task A using a large dataset A. However, the dataset B for a target task B is too small, preventing Model B from training efficiently. Thus, we use part of model A to predict results for task B (Fig 9).

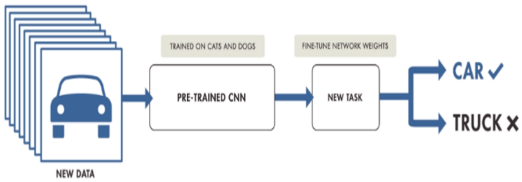

Fig. 8. The overview of Transfer Learning [26]

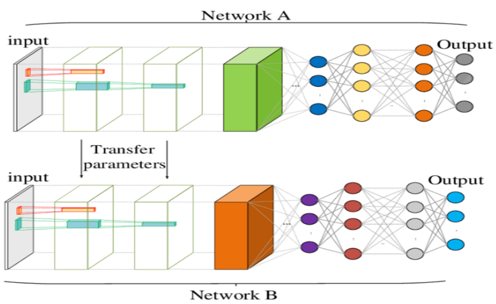

Fig. 9. The process of a pre-trained network in Transfer Learning [28]

There are several different strategies for Transfer learning. The simplest strategy is to solve a on a different Target task but in the same Domain by directly using a full or part of an already pre-trained model from a source task. Also, feature extractor is another usage of a pre-trained model instead of using the whole model. This approach allows us to directly apply new dataset to solve an entirely different problem. Since the new task employ the pre-trained complex model as features, we are allowed to train a simpler and faster linear model to modify the output based on the new dataset. This strategy is best when the target task dataset is very small. Finally, we can go one step further, by not only training the output classifier, but also fine-tune weights in some layers of the pre-trained model. This strategy is best when the target task dataset is significantly big.

3) Inception V3: When we have a relatively small dataset, a super-effective technique is to use Transfer Learning where we use a pre-trained model. This model has been trained on an extremely huge dataset, and we would be able to transfer weights which were learned through hundreds of hours of training on multiple high powered GPUs. Big corporations tend to release such models to the public, aiming to enhance the development of this field. Some pre-trained models used directly include Inception V3, BERT as well as YOLO, GloVe, UnsupervisedMT and etc. Many such models are open-sourced such as VGG-19 and Inception-v3. They were trained on millions of images with extremely high computing power which can be very expensive to achieve from scratch.

InceptionV3 is a model that is made up of results of several research [29]. It includes convolutions, pooling layer, dropout techniques, and fully connected layers. A dataset of 1,000 categories from the ImageNet [12] dataset is employed to train this model. The ImageNet consists of ten million URLs of labeled images.

B. Data Preprocessing

1) Data Distribution: In this section, we explain data distribution, data augmentation, and data rescaling and resizing. First of all we divide dataset into three subsets including train, test, and validation set. The initial dataset was included 1805 images. We separate 80% of it for training, 10% for test, and 10% for validation set. The size of each classes is shown in Table I. Also, the distribution of each subset is displayed in Fig 10. As we can see, the proportion of each class is equal in the three subset.

TABLE I
THE SIZE OF SAMPLES IN TRAIN, TEST, AND VALIDATION SET

	The number of samples			
	Bloodroot	Clubmoss	Dandelion	Lobelia
Training set	360	524	250	371
Test set	36	52	25	37
Validation set	36	52	25	37

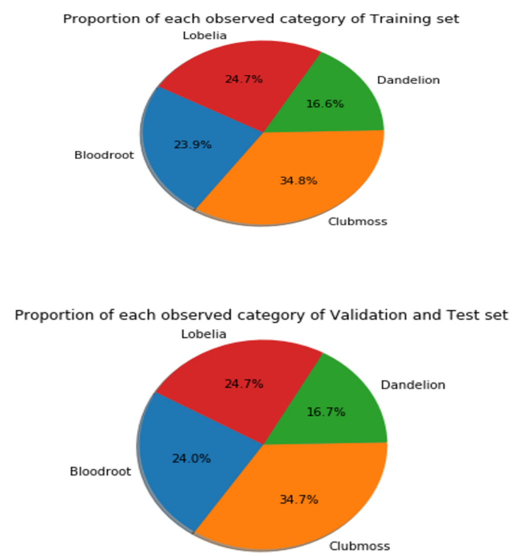

Fig. 10. The proportion of samples in train, test, and validation set

2) Data Augmentation: Since we have a small number of data, we use data augmentation technique to increase the size of data by generating different shapes of images from the given data. We use ImageDataGenerator class form keras.preporssing library for training dataset to obtain augmentation by creating train data object. Rotation, shifting, shearing, zooming, and flipping are different data augmentation we apply to our dataset. Rotation do randomly rotate pictures in range 0 to 180. Shifting do randomly translate images horizontally or vertically in a range of a fraction of total height or width. Flipping do at a random manner flip half of the images horizontally. In summary, augmentation transforms images randomly into different shapes, so that there is impossible to have an image duplicat. This helps the model generalize better and eventually reduces the chance of overfitting.

3) Data Rescaling and Resizing: All the images are consist of RGB values with pixel range from 0 to 255. However, we need to rescale these values between 0 and 1, because values in range 0 to 255 would be too high for our model to process. Therefore, we use the same ImageDataGenerator class to update train data and create two new object validation data and test data from validation and test set respectively.

Also, since we collect data from different sources, images have different height and width. Therefore, we need to resize images to have the same size in terms of width and height in training, validation, and testing set. In this case we keep height and width pixel size 150 to generate the final training set. Moreover, we apply the same approach to generate validation and test set using validation data and test data object respectively to resize each images into 150X150 pixels. A random batch of training set of size 25 is shown in Fig 11.

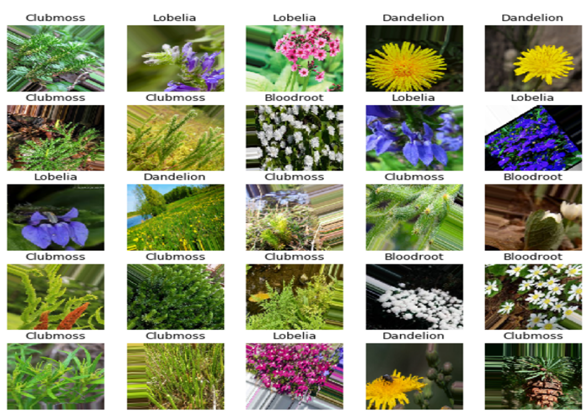

Fig. 11. A random batch of training set

C. Model Implementation

In this section, we explain model construction, compilation, and validation. To construct the model, we use InceptionV3 model as a pre-trained model/base model. The model is available in keras.application library. The model construction includes the following steps: i) First, we initialize InceptionV3 model as our base model. The parameters trained on ImageNet dataset. ii) Then, we add a Global Average Pooling (GAP) layer to the model to reduce the spatial dimensions of a three dimensional tensors, which eventually leads to minimize overfitting. It reduces h * w * d dimensions to 1 *1 * d dimensions by taking the average of h and w values (Fig 12).

iii) Then, we add a fully connected layer, FC, of 1024 hidden nodes. iv) Finally, the FC layer is connected to our output layer to predict output. The FC layer mainly takes the output from GAP as input and provides output to our final output layer to predict the best label for each image. We use 'Softmax' as the activation function, because the number of output classes are more than two. We use 65 layers in the model and make 249 layers non-trainable. The summary of the constructed model is provided in II.

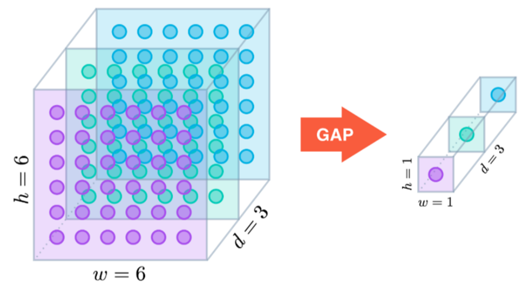

Fig. 12. Reduce of the spatial dimensions of a three dimensional tensors using Global Average Pooling layer (GAP) [28]

TABLE II
THE SUMMARY OF THE FINAL CONSTRUCTED MODEL

The total number of layers	**Non-trainable Layers**	**Trainable Layers**
314	249	65

In order to compile the method, we use RMSprop with learning rate 0.0001 as the optimizer and categorical cross-entropy as loss function. After rescaling the images, and using Image Augmentation, we flow them in batches of 15 using Early Stopping regularization function on the train and test data. Validation data do not augmented, just test and train set.

In order to obtain the best result, we also investigate other optimizers, such as RMSProp, Adam, and SGD and compare the results. We calculate precision, recall, and accuracy as the performance evaluation metrics.

VI. RESULTS AND DISCUSSION

The result is one of the most important parts of any project. In this project, we achieved 95.3% test accuracy on the Incept V3 model. It requires 4 epochs to achieve this accuracy. Table III displays the detailed result of all epochs. The first 4 column includes training loss, training precision, training recall, and training accuracy while the rest four-column includes validation loss, validation precision, validation recall, and validation accuracy. Finally, we achieved 95.33% test accuracy with 0.91 precision and 0.85 recall.

TABLE III
TRAINING AND VALIDATION RESULT

Epoch	**Train Loss**	**Train Pre-ci-sion**	**Train Re-call**	**Train Ac-cu-racy**	**Val Loss**	**Val Pre-ci-sion**	**Val Re-call**	**Val Ac-cu-racy**
1	0.3098	0.8848	0.7687	0.8983	0.2555	0.8931	0.7961	0.9200
2	0.3321	0.8952	0.8044	0.8937	0.1318	0.8970	0.8157	0.9067
3	0.2737	0.9000	0.8253	0.9116	0.7863	0.9025	0.8342	0.9133
4	0.2282	0.9056	0.8418	0.9336	0.2996	0.9090	0.8496	0.9200

Moreover, we achieved 95.33% accuracy with RMSProp. In the case of SGD and Adam, it requires 6 epochs and accuracy is also quite good but lower than the RMSProp. Table IV shows the detailed result from different optimizers to compare them.

TABLE IV
TEST RESULT WITH DIFFERENT OPTIMIZER

Optimizer	Total Epoch	Test Loss	Test Precision	Test Recall	Test Accuracy
RMSProp	4	0.02	0.91	0.85	95.33
SGD	6	0.07	0.95	0.94	92.67
Adam	6	0.04	0.95	0.94	93.33

In order to compare the results, we also implement our algorithm on the dataset without applying image augmentation. In this case, we saw a sharp reduction in terms of accuracy with RMSProp and SGD. Without Image augmentation, we achieved only 75.33% test accuracy on RMSProp. But, in the case of Adam, we found 90.00% test accuracy which is nearly the same as the image augmentation case. Table V displays the detailed result from different optimizers without image augmentation.

TABLE V
TEST RESULT WITHOUT IMAGE AUGMENTATION

Optimizer	Total Epoch	Test Loss	Test Precision	Test Recall	Test Accuracy
RMSProp	8	4.05	0.91	0.73	75.33
SGD	5	0.15	0.87	0.82	78.67
Adam	5	0.28	0.94	0.88	90.00

As a result, after performing an operation on different test cases, we found that for the small dataset, image augmentation can increase the accuracy. In our study, the performance of RMSprop optimizer is higher than SGD and Adam. With the help of default Inception V3, we achieve 95.33% accuracy. Fig 13 shows the accuracy, loss, precision, and recall for training and validation of the model. The results for testing the model are provided in Table VI.

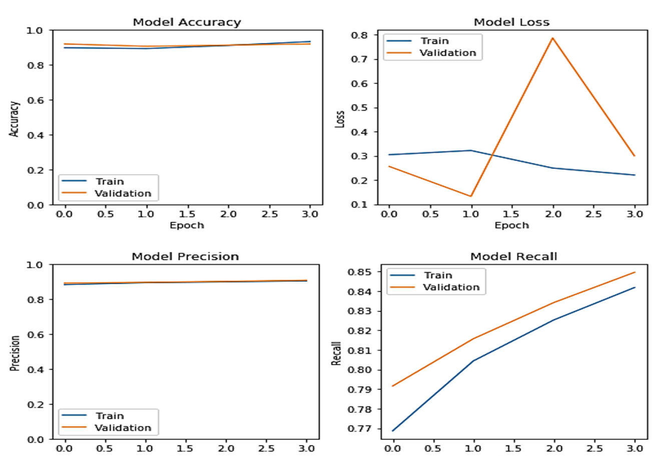

Fig. 13. The comparison between train and validation in terms of accuracy, loss, precision, and recall

TABLE VI
THE RESULTS FOR TESTING THE MODEL IN TERMS OF ACCURACY, LOSS, PRECISION, AND RECALL

Test Loss	0.02
Test Precision	0.91
Test Recall	0.85
Test Accuracy	95.33%

VII. CONCLUSION

To conclude, we aim to create an application that can classify Canadian medicinal plants. In this study, we use InceptionV3 module with transfer learning and image augmentation. In a dense layer, we use ReLU as an activation function because it is one of the most popular activation function. We also used early stopping regularization with patience parameter 3. By conducting our experiment with different optimizers, we find that our application RMSProp provides better accuracy. At the end, we achieve 95.33% test accuracy using RMSProp.

In the future, we plan to increase the accuracy by expanding the size of dataset. Currently, our application can only classify the four plants. In the Future, we would like that it can classify more plants. We also will test our application on the other deep models like Inception V4 and Inception ResNet.

REFERENCES

[1] Y. Uprety, H. Asselin, A. Dhakal, and N. Julien, "Traditional use of medicinal plants in the boreal forest of canada: review and perspectives," *Journal of ethnobiology and ethnomedicine*, vol. 8, no. 1, p. 7, 2012.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[3] Y. Yang and T. M. Hospedales, "Deep neural networks for sketch recognition," *arXiv preprint arXiv:1501.07873*, vol. 1, no. 2, p. 3, 2015.

[4] G.-h. Song, X.-g. Jin, G.-l. Chen, and Y. Nie, "Two-level hierarchical feature learning for image classification," *Frontiers of Information Technology & Electronic Engineering*, vol. 17, no. 9, pp. 897–906, 2016.

[5] Y. Liu, F. Tang, D. Zhou, Y. Meng, and W. Dong, "Flower classification via convolutional neural network," in *2016 IEEE International Conference on Functional-Structural Plant Growth Modeling, Simulation, Visualization and Applications (FSPMA)*, pp. 110–116, IEEE, 2016.

[6] H. Yalcin and S. Razavi, "Plant classification using convolutional neural networks," in *2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, pp. 1–5, IEEE, 2016.

[7] M. P. Shah, S. Singha, and S. P. Awate, "Leaf classification using marginalized shape context and shape+ texture dual-path deep convolutional neural network," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 860–864, IEEE, 2017.

[8] A. Kaya, A. S. Keceli, C. Catal, H. Y. Yalic, H. Temucin, and B. Tekinerdogan, "Analysis of transfer learning for deep neural network based plant classification models," *Computers and electronics in agriculture*, vol. 158, pp. 20–29, 2019.

[9] D. Ezzat, A. E. Hassanien, M. H. N. Taha, S. Bhattacharyya, and S. Vaclav, "Transfer learning with a fine-tuned cnn model for classifying augmented natural images," in *International Conference on Innovative Computing and Communications*, pp. 843–856, Springer, 2020.

[10] Y. Sun, Y. Liu, G. Wang, and H. Zhang, "Deep learning for plant identification in natural environment," *Computational intelligence and neuroscience*, vol. 2017, 2017.

[11] P. Barré, B. C. Stöver, K. F. Müller, and V. Steinhage, "Leafnet: A computer vision system for automatic plant species identification," *Ecological Informatics*, vol. 40, pp. 50–56, 2017.

[12] K. Mitrović and D. Milošević, "Flower classification with convolutional neural networks," in *2019 23rd International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 845–850, IEEE, 2019.

[13] M. TOĞAÇAR, B. ERGEN, and Z. CÖMERT, "Classification of flower species by using features extracted from the intersection of feature selection methods in convolutional neural network models," *Measurement*, p. 107703, 2020.

[14] G. Saini, A. Khamparia, and A. K. Luhach, "Classification of plants using convolutional neural network," in *First International Conference on Sustainable Technologies for Computational Intelligence*, pp. 551–561, Springer, 2020.

[15] A. K. Das, M. A. Iqbal, B. Paul, A. Rakshit, and M. Zahid, "Classification of succulent plant using convolutional neural network,"

[16] T. Nishat, I. Md Romyull, and S. Shaon Bhatta, "A convolution neural network based classification approach for recognizing traditional foods of bangladesh from food images," in *Proceedings of the International Conference on Intelligent Systems Design and Application(ISDA)*, pp. 844–852, Springer, 2018.

[17] I. Md Romyull, T. Nishat, and S. Shaon Bhatta, "Mobilenet model for classifying local birds of bangladesh from image content using convolutional neural network," in *Proceedings of the 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–4, IEEE, 2019.

[18] S. Banerjee and R. Pamula, "Random forest boosted cnn: An empirical technique for plant classification," in *Proceedings of the Global AI Congress 2019*, pp. 251–261, Springer, 2020.

[19] R. N. Lochan, A. S. Tomar, and R. Srinivasan, "Plant detection and classification using fast region-based convolution neural networks," in *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, pp. 623–634, Springer, 2020.

[20] B. Espejo-Garcia, N. Mylonas, L. Athanasakos, S. Fountas, and I. Vasilakoglou, "Towards weeds identification assistance through transfer learning," *Computers and Electronics in Agriculture*, vol. 171, p. 105306, 2020.

[21] P. Bosilj, E. Aptoula, T. Duckett, and G. Cielniak, "Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture," *Journal of Field Robotics*, vol. 37, no. 1, pp. 7–19, 2020.

[22] R. G. de Luna, E. P. Dadios, A. A. Bandala, and R. R. P. Vicerr, "Tomato growth stage monitoring for smart farm using deep transfer learning with machine learning-based maturity grading," *AGRIVITA, Journal of Agricultural Science*, vol. 42, no. 1, 2020.

[23] J. Arunnehru, B. Vidhyasagar, and H. A. Basha, "Plant leaf diseases recognition using convolutional neural network and transfer learning," in *International Conference on Communication, Computing and Electronics Systems*, pp. 221–229, Springer, 2020.

[24] K. Rangarajan Aravind and P. Raja, "Automated disease classification in (selected) agricultural crops using transfer learning," *Automatika*, vol. 61, no. 2, pp. 260–272, 2020.

[25] Y. Li, Z. Hao, and H. Lei, "Survey of convolutional neural network," *Journal of Computer Applications*, vol. 36, no. 9, pp. 2508–2515, 2016.

[26] P. udikandula, "Deep view on transfer learning with iamge classification pytorch available at: https://medium.com/@purnasaigudikandula/deep-view-on-transfer-learning-with-iamge-classification-pytorch-5cf963939575. [accessed: 21- mar- 2019]."

[27] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *International conference on artificial neural networks*, pp. 270–279, Springer, 2018.

[28] A. Cook, "Global average pooling layers for object localization, available at: https://alexisbcook.github.io/2017/global-average-pooling-layers-for-object-localization [accessed: 21- mar- 2019]."

[29] X. Xia, C. Xu, and B. Nan, "Inception-v3 for flower classification," in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, pp. 783–787, IEEE, 2017.