CNNabis

Final Report

Cong Yu Fang (1003145349), Jianing Li (1003416378)

2018.12.02

word count: 1986

penalty: 0

1 Motivation and Goal

It is common for tourists coming to Canada to keep a maple leaf as the souvenir of their visit. However, on Oct. 18, 2018, Cannabis is legalized in Canada [1]. Tourists are now more exposed to cannabis, so they have a greater chance of mistaking a cannabis leaf for a maple leaf and taking it back to their home countries, where possession of such a thing could be illegal. It will be very helpful for them to have a tool that helps them distinguish maple leaves and cannabis leaves.

Also, most tourists know nothing about their souvenir beyond the fact that it is a maple leaf. In fact, most of them aren't even aware that there are many types of maple leaves out there. We can enrich the experience of their visit by telling them which type of maple leaf theirs belongs to.

To address these needs, we present CNNabis, a CNN-based software that classifies 7 most common types of maple leaf in North America and Cannabis leaf. It tells the tourists if their souvenir is a cannabis leaf. If not, it tells the tourists what type of maple leaf it is.

2 Overall Software Structure

The software consists of the user interface, the data pre-processing module and the classifier model, as shown in Figure 1

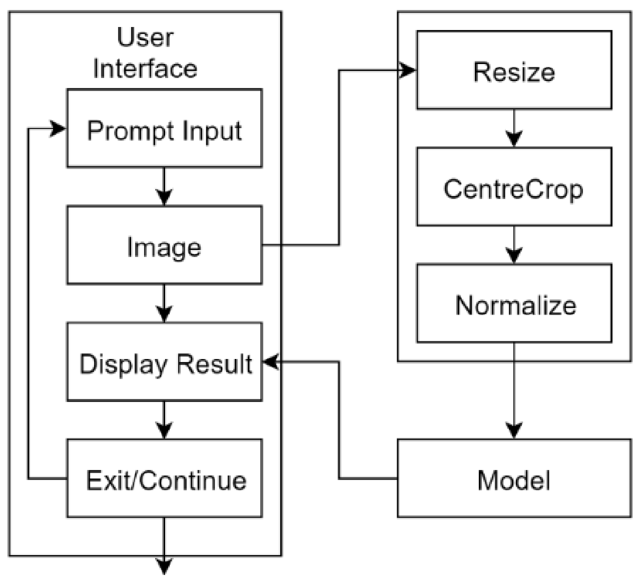


Figure 1: Overall Structure of the Program

The user interface instructs the user to click "Begin" to load the image for classification, upon which action a window pops out for the user to browse the file system for the image of the souvenir. Upon selecting an image, the software sends it through the pre-processing module and then into the classifier. The result of the classification, which consists of the two most probable classes for the image, is displayed on the user interface. The user can now click "Begin" again to classify another image or click "Exit" to finish using the software, as shown in Figure 2.

2.1 Data Pre-processing Module

Data pre-processing module consists of re-sizing and normalizing. The purpose of this module is to prepare the image for input into the classification model.

Figure 2: The user interface: it is built with the Python package Tkinter. It provides the user with means of operating the software and obtaining information from the software. In our case, the user is the tourist.

The re-sizing step re-sizes the image into a square, which is required for input into ResNet18, the model that we will discuss more later. It is followed by centre crop, which is trying to crop out the leaf in the centre. Normalization of colour is also used since it can remove the intensity values from the image while preserving color values [4].

2.2 Classifier

The classifier is a CNN-based model that takes in the pre-processed image and outputs the probability that the leaf in the image belongs to each of the 8 classes (7 types of maple leaves and cannabis). The most probable class is selected as the prediction, and the second most probable class is used as a secondary guess.

3 Data Collection

Data collection consists of downloading images, selecting images and pre-processing/augmenting images.

3.1 Downloading Images

Images of maple leaves and cannabis are collected by scraping Google images using the Python package google_images_download. To operate this package, we have to specify the search word (e.g., 'cannabis leaf') and a limit on the number of images to download (e.g., 1000). We downloaded around 1000 images of each class, totalling around 8000 images across all 8 classes.

3.2 Selecting Images

The downloaded images are generally of high enough resolution, but there are also some problems present in them, for example, the maple leaf images of the wrong type, drawings of maple leaves instead of real images, or photos of an entire maple tree showing from afar rather than a leaf. We manually filtered out these messy data, leaving us with around 130 images of each class, totalling 1052 images.

3.3 Pre-processing and Augmenting Images

Like the pre-processing module in the final software introduced above, this pre-processing module also resizes, centre-crops and normalizes the images. However, since 1052 images is certainly not enough data to train the model, we also have an augmentation stage, where we create more images out of one single image to effectively amplify our dataset.

The choice of amplification tools is important. Some tools might impede the training of the model by damaging useful features while others won't. For instance, we are using random affine transformation, which includes rotation, stretching and shearing. Rotation is helpful since the user could place the leaf in any orientation; stretching and shearing is helpful since they mimic a different shooting angle.

4 Model Structure

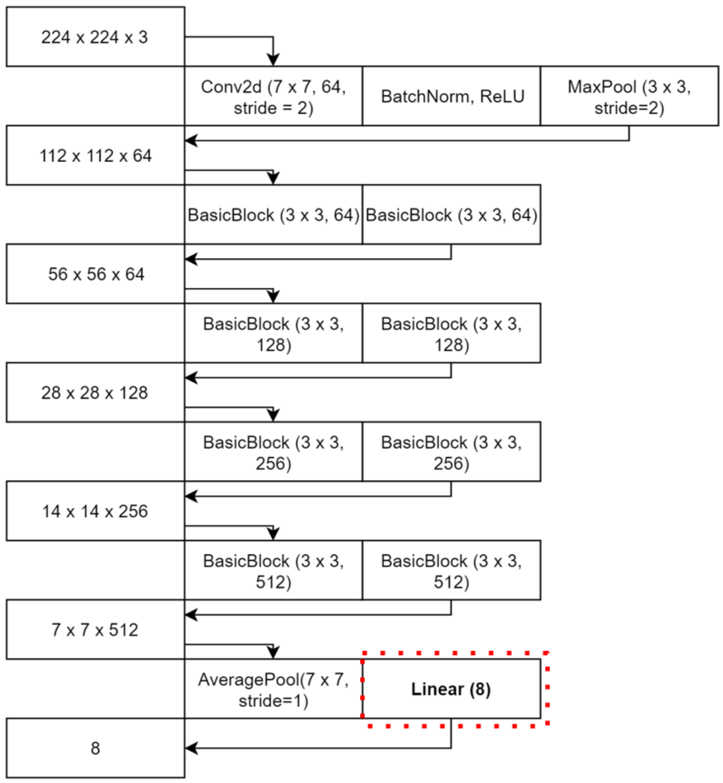
The model in our software is a CNN-based model with 17 convolution layers and 1 fully connected layer. It is transferred from a pre-trained ResNet-18 model, but we replaced its fully connected layer with our own, which has 8 outputs as desired.

We chose to use a convolutional neural network for this image recognition/classification problem since it is the most widely used model for such tasks due to its capability to scan across an image to look for a specific feature and its efficiency with parameters.

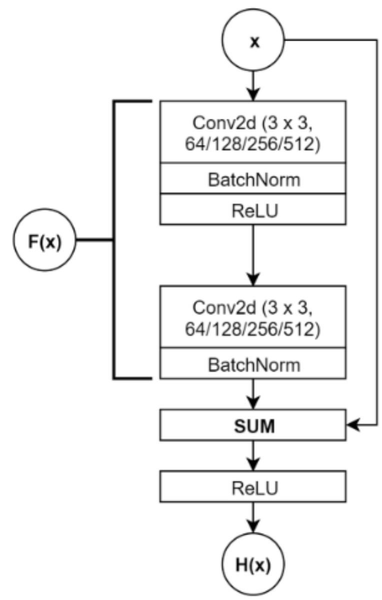
At first, we built our own model, which consists of 2 convolutional layers and 3 fully connected layers. However, this model plateaus at a validation accuracy of around 30%, only a little bit more than twice the chance, which is 12.5%. We reasoned that our mode isn't powerful enough for the task. Some different types of leaves could look very similar to each other (e.g., big leaf maple and sugar maple), thus requiring the model to be able to detect subtle features, but on the other hand, images of leaves of the same type could vary quite a bit in image background, shooting angle or even colour of the leaf itself (depending on the season when the photo was taken). Therefore, we went for a deeper model.

The model we went for is a ResNet-18. Its structure is shown in Figure 3a, with the last fully connected layer replaced with our own 8-output linear layer. The advantage of a ResNet over a generic deep convolutional neural network is that if we add convolution layers naively, the deeper layers will tend to approximate an identity function, so the depth of the model doesn't help with performance, and it can even result in degradation of the model performance. The way ResNet deals with degradation can be seen in the structure of a basic block [3], which is shown in Figure 3b. Instead of optimizing the overall non-linear function H(x) directly, ResNet optimizes the residual function, which is the difference between the overall function and the identity function (F(x)=H(x)-x). This design of ResNet allows the model performance to increase alongside its deepening.

However, we only have 1052 images. Even after augmentation, training such a deep model from scratch will almost definitely result in bad performance and over-fitting. Therefore, we use transfer learning, where we train on top of a pre-trained ResNet. Previous work has used transfer learning with ResNet to classify 500 species of plants [5]. However, the pre-trained ResNet was trained on ImageNet 1000-class classification task [2], meaning it is a relatively general-purpose classifier, so making it fit to the task of classifying leaves and recognize the subtle features of leaves would necessarily require altering the convolution layers. Therefore, we allow all parameters in the model to vary, not only those in our self-added fully connected layer.



(a) Structure of ResNet18 (last linear layer modified): The input image first goes into a convolution layer with kernel size 7x7 and the associated batch normalization, ReLU and max pooling layers. It then passes through a series of pairs of Basic Blocks, before going into an average pooling and a fully connected layer, which we replaced with one of our own with 8 outputs.



(b) Structure of Basic Unit

Figure 3: Structure of our model

5 Training, Validation and Testing

The training of the model consists of the following steps:

- First, the 1052 images are split into training, validation and test set at an 8:1:1 ratio.
- Then, the training set is used to train the model, which again is a pre-trained ResNet-18 model with its fully connected layer replaced with our own. Details of the training is stated below as shown in Figure 4.
- We used a batch size of 64.
- The images are loaded batch by batch during training and are pre-processed and augmented as they are loaded, which again consists of re-sizing, centre-cropping, normalizing and random affine transformation.
- The loss function used is cross entropy loss
- The optimizer used is Adam optimizer, with a learning rate of 0.001. We also used learning rate decay, with a learning rate half life of 7 epochs.

- The model is trained for 25 epochs, at the end of which we observe saturation. The validation set is checked after each epoch and the test set is checked after all epochs.

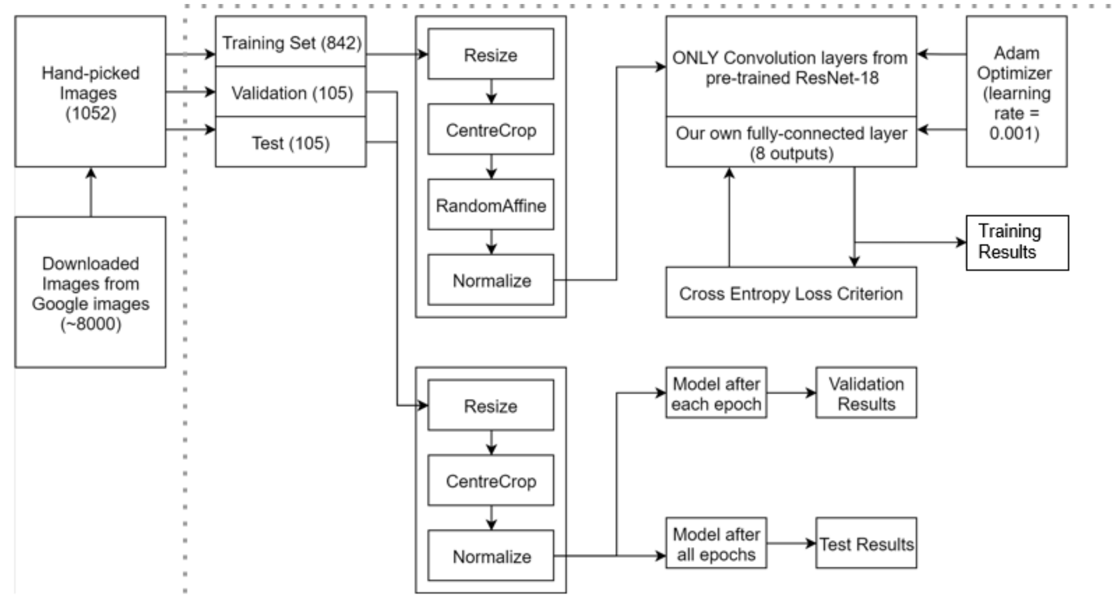


Figure 4: Training, Validation and Testing Procedure

The best result we obtained shown in Figure 5a is a training accuracy of 94.3%, a validation accuracy of 85.7% and a test accuracy of 83.8%. Note that we allow all parameters to vary in the model. We also tried to complete the training with first our own shallow model (Figure 5b), then with a non-pre-trained ResNet (Figure 5c), finally with a pre-trained ResNet with frozen convolution layers (Figure 5d). All of these trials returned worse results than our best model, proving the superiority of a deep pre-trained model, as well as the necessity to allow all pre-trained parameters to change and adapt to the radically distinct task of classifying leaves.

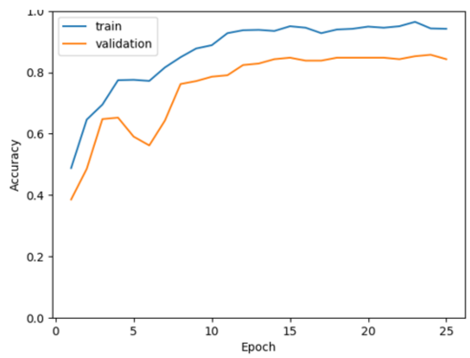
Figure 6 is the confusion matrix showing the predictions of the model on the test set. Note that aside from the generally high accuracy of the model, there is no cannabis leaf predicted as a maple leaf, at least in the small test set we run the model on. This is important since predicting a cannabis leaf as a maple leaf can put the user into trouble.

It is also notable that even in our small test set, some instances of silver maple are predicted as big leaf, box elder, cannabis or Norway maple. This is understandable since the incorrectly predicted silver maple images are obviously the more confusing ones. For instance, the one predicted as Norway maple has leaves that are possibly too small to be recognizable, and the one predicted as box elder is a pre-mature silver maple leaf, which looks more round thus resembles a box elder leaf.

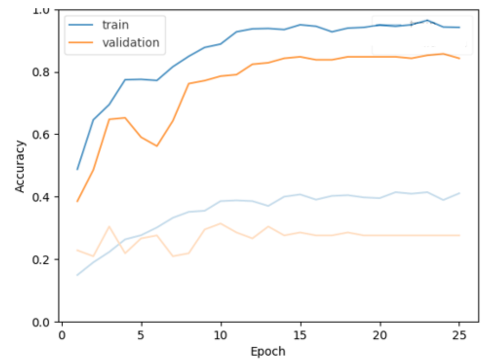
6 Ethical Issues

An ethical issue with the project is as follows: There will almost definitely be occasions where the software makes incorrect predictions, and if the error is of the type of mistaking a cannabis leaf for a maple leaf, it would put the user in disastrous situation.

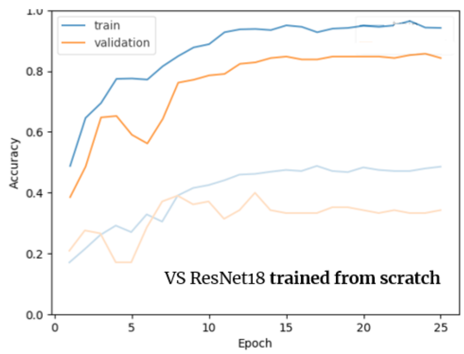
As creators of the software, we can declare in Terms and Conditions that we are not responsible in case the prediction of the software makes the user guilty of illegal drug possession. However, the burden on us here is



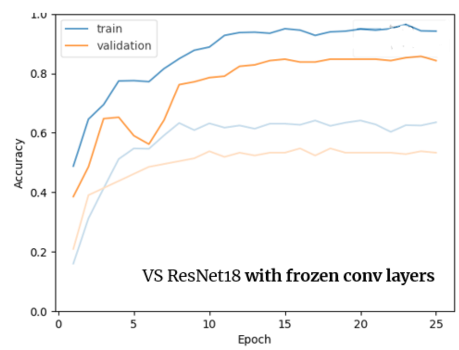
(a) Training and validation accuracy of the best model we achieved (Pre-trained ResNet18 and parameters are allowed to vary)



(b) Training and validation accuracy of the best model and our shallow model (2 convolutional layers followed by 3 linear layers)



(c) Training and validation accuracy of the best model and the ResNet18 model trained from scratch (non-pre-trained ResNet18)



(d) Training and validation accuracy of the best model and pre-trained ResNet18 with frozen convolutional layers

Figure 5: Performances of Models

an ethical one instead of a legal one. Although we want the software to help the tourists, we are also putting the users at risk. What if the user already decided not to keep a cannabis leaf because he or she isn't sure about what it is, but the software tells him or her that it's a silver maple leaf so he or she keeps it? Our software could indeed save many tourists from unknowingly committing a crime, but is it ethical to do this at the cost of sacrificing some other users?

7 Key Learnings

The greatest learning of the project is that state-of-the-art deep convolutional neural networks are very powerful. At first, we were almost certain that even a pre-trained deep model will have huge difficulties with the task, since leaves of different classes can look very similar sometimes but photos of leaves of the same class can vary a lot in the way they are taken. However, it turns out that the model can handle the classification task fairly well, achieving a validation accuracy almost 7 times as high as chance.

Therefore, if we were to start again, we would definitely try doing transfer learning with more models like ResNeXt-101, VGG16, Inception-v3, etc. we can also try the cutting-edge Deformable Convolutional Network (DCNv2). It consists of Deformable Convolution and Deformable Region of Interest Pooling, allowing the model to adapt to changes in the shape of the object [6]. This is highly helpful for our task since leaves can flex.

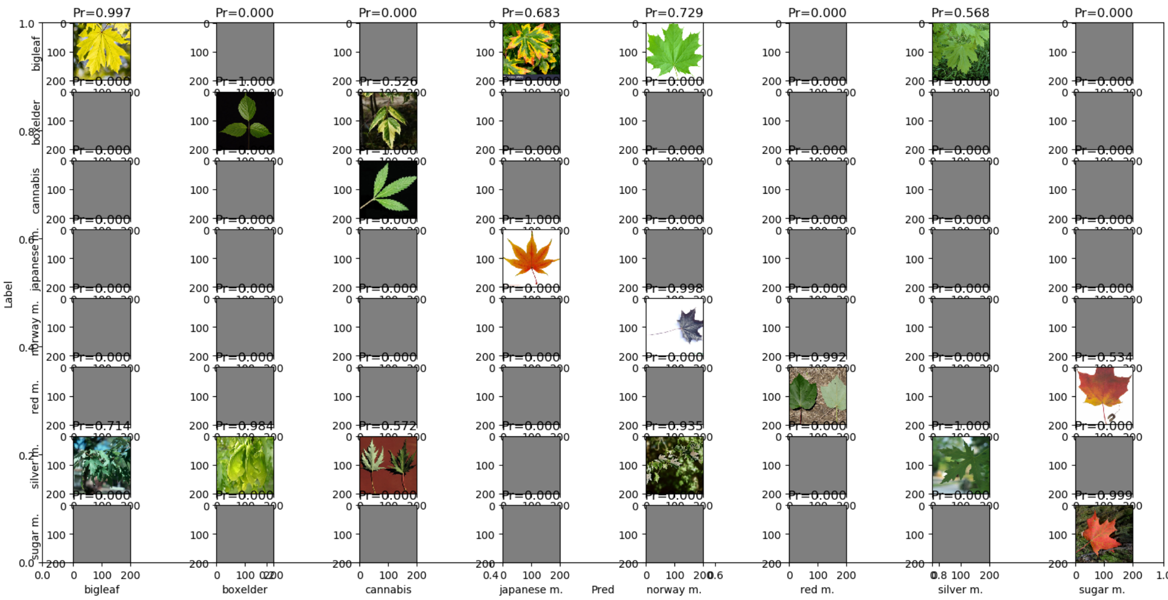


Figure 6: Visualization of Confusion Matrix. Pr represents the probability of the image being predicted as a certain class and each image shown in this matrix is the image with the highest probability in that spot. The vertical axis represents the ground truth; the horizontal axis represents the predicted results.

References

[1] *Cannabis legalization*. Oct. 2018. URL: `https://www.ontario.ca/page/cannabis-legalization?gclid=Cj0KCQiAuf7fBRD7ARIsACqb8w4VMMOR_mfV_Zfr0B5yiQQIbBzT6_T8wCJFkVRD1D3xuC1jAYfgQC8aAhO-EALw_wcB`.

[2] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee. 2009, pp. 248–255.

[3] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016). DOI: `10.1109/cvpr.2016.90`. URL: `https://arxiv.org/abs/1512.03385`.

[4] M. Vanrell et al. “Colour normalisation based on background information”. In: *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)* (Aug. 2002). DOI: `10.1109/icip.2001.959185`. URL: `https://ieeexplore.ieee.org/document/959185?arnumber=959185`.

[5] Vladislav Vorobey. *Building Image Classification Model Based on Pre-Trained Neural Network*. Nov. 2017. URL: `https://indatalabs.com/blog/data-science/building-image-classification-model-using-pre-trained-neural-networks`.

[6] Xizhou Zhu et al. “Deformable ConvNets v2: More Deformable, Better Results”. In: *Computer Vision and Pattern Recognition* (Nov. 2018). URL: `https://arxiv.org/abs/1811.11168`.