UC San Diego
Electrical and Computer Engineering
JACOBS SCHOOL OF ENGINEERING

Flowers Classification via Deep Learning

Bo Chen, Jun Liu, Jiacheng Sun, Jiong Liu

Group 40

UC San Diego
Electrical and Computer Engineering
JACOBS SCHOOL OF ENGINEERING

Abstract

Due to time restrictions or computational restraints, it's not always possible to build a model from scratch about image classification problem in physics, we will compare the performance of methods and try to find a pre-trained way for this kind of problem.
The problem we are trying to solve is given an image of flowers, we want to determine which type it is (daisy/ dandelion/ rose/ sunflower/ tulip). To solve this classification problem, we plan to use different kinds of approaches including machine learning (Support vector machine[1]/ Random Forest[2]) and deep learning methods (Custom CNN/ VGG[3]/ ResNet[4]).

Dataset and Features

We use a dataset contains 4323 images of flowers from kaggle[5]. The data collection is based on the data flickr, google images, yandex images. We can use this dataset to recognize plants from the photo. It's over 200 megabytes and contains over 4000 images of flowers. It includes 5 types of flowers so we have 5 classes or 5 labels for this classification problem. Each image has a different resolution, like Example(1),(2) below.
The raw input we use for training is the image after resizing, usually a 128x128x3 matrix for image pixels, like (b) Resized below. While for some machine learning method(RF, SVM), we flatten the image into a list of row pixels.
In order to get a better result for CNN and VGG, we do data augmentation methods such as randomly flipping, shifting and zooming the images, and applying ZCA method. This augmentation is helpful for feature weights extraction.

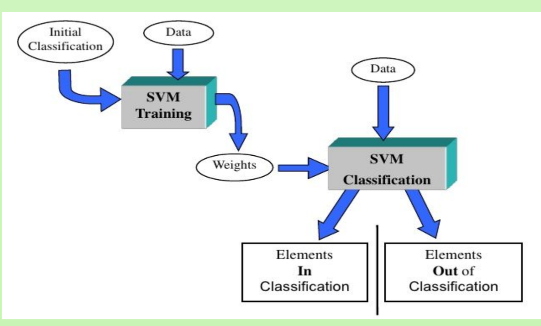
Models

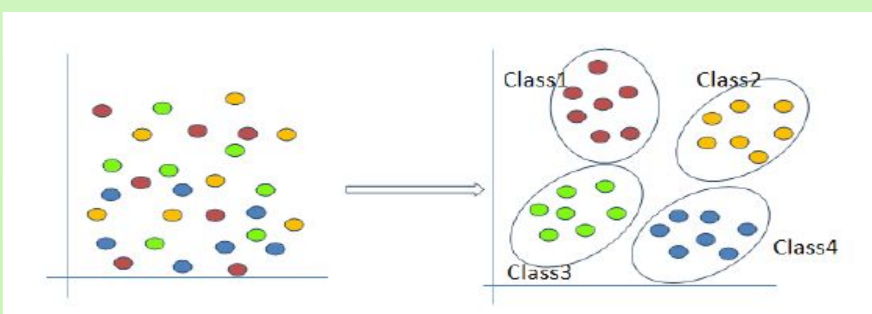
Several approaches have been applied to this problem

1. Traditional **Machine Learning** method

- Support Vector Machine

  SVM can efficiently perform a nonlinear classification by mapping their inputs into high-dimensional feature spaces.





- Random Forest

  RF is a conditional weighted method only uses important features to do the classification. Gini impurity is chosen as the metric to split the root.

$$I_G(p)=\sum_{i=1}^{J} p_i \sum_{k\neq i} p_k=\sum_{i=1}^{J} p_i(1-p_i)=\sum_{i=1}^{J}(p_i-{p_i}^2)=\sum_{i=1}^{J} p_i-\sum_{i=1}^{J} {p_i}^2=1-\sum_{i=1}^{J} {p_i}^2$$

  where J is the number of classes, and p is the portion of data that belongs to class i. The dataset keeps splitting based on Gini index until all the nodes cannot be splitted any more or a specific threshold is reached.

- K-nearest Neighbors

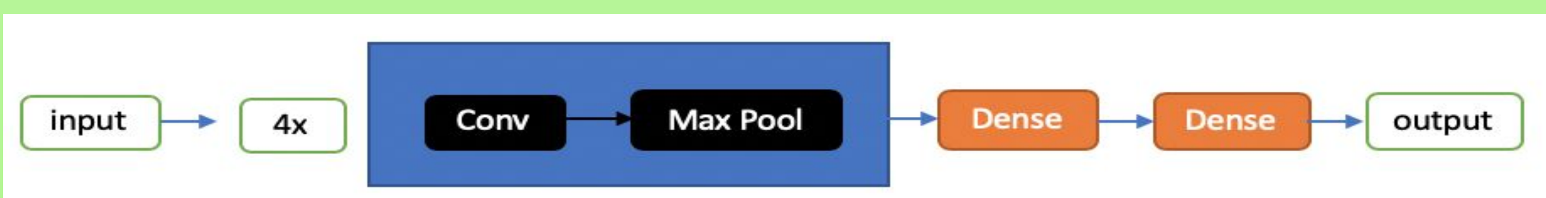
  It runs through the whole dataset computing distance between the given point x and each training observation according to Euclidean distance, then estimate the conditional probability for each class.

$$D_H=\sum_{i=1}^{k}|x_i-y_i|$$
$$x=y\Rightarrow D=0$$
$$x\neq y\Rightarrow D=1$$

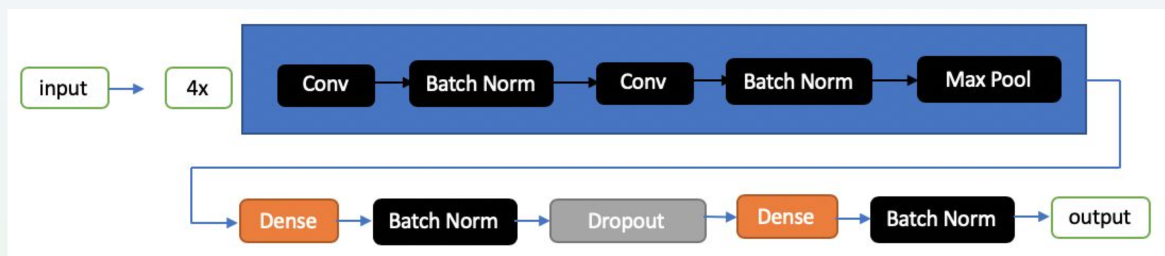
2. **Deep Learning** Method

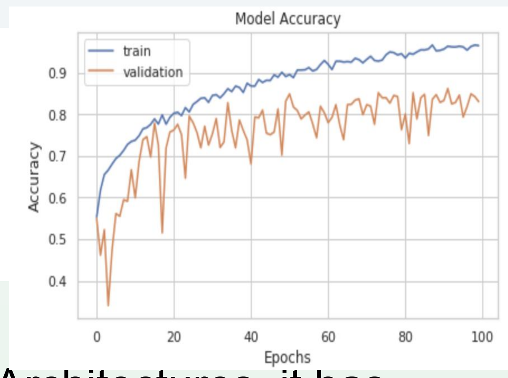
- Naive Neural network: A simple single layer Neural network
- Custom CNN I & II

  Convolutional Neural Network is to apply a convolution operation to the input and pass the convoluted input(layer) to the next layer and repeating this process. We make our own custom CNN based on the architecture below.



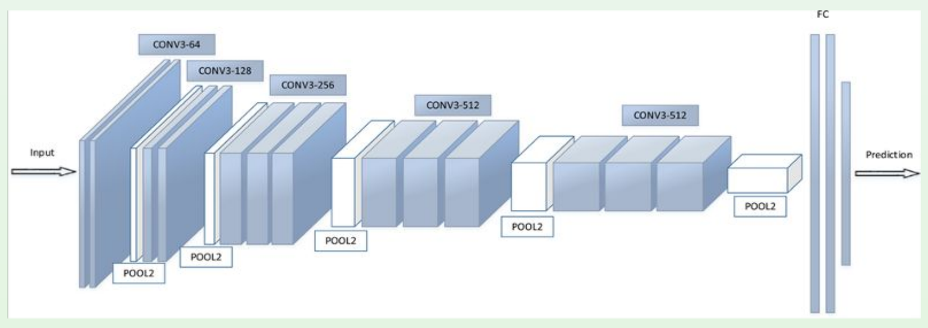
Models Continue

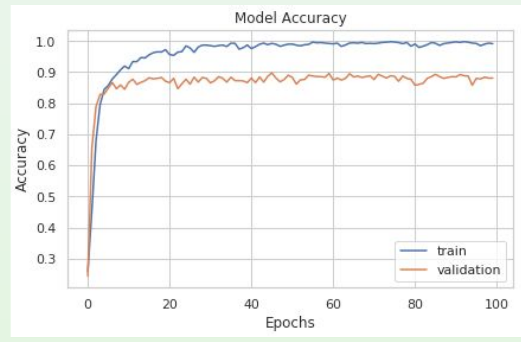




- VGG

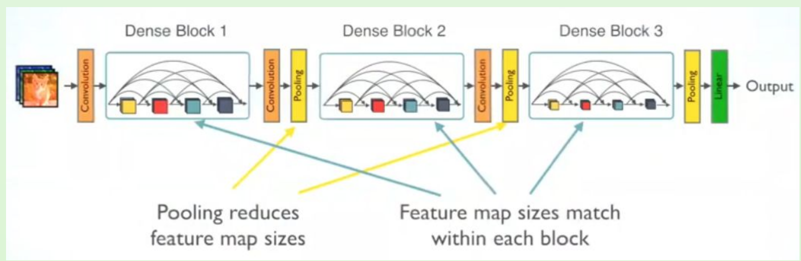
  Visual Geometry Group(VGG)[3] is a well-trained CNN Architectures, it has features like containing total 16 layers in which weights and bias parameters are learnt.

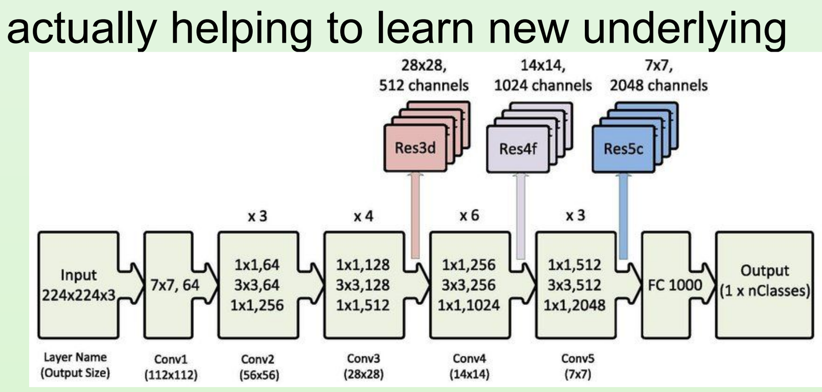




- Resnet & DenseNet

  By using residual blocks in the network, one can construct networks of any depth with the hypothesis that new layers are actually helping to learn new underlying patterns in the input data.[6]





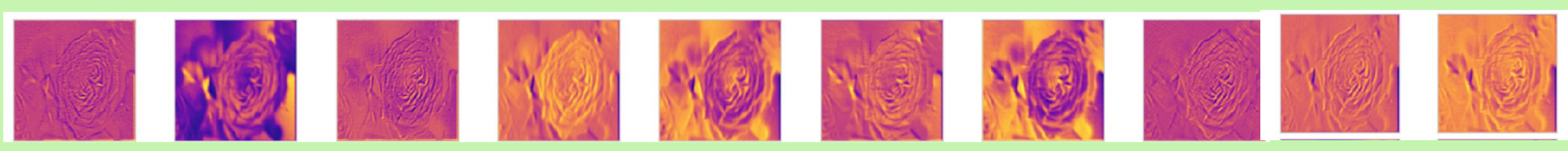
Results

Accuracy rate of test set is used to measure the difference between classification of the model and the flower label, loss is for training process. It's like all the results of machine learning method are around 0.5, while these of deep learning method are above 0.85. There are 4323 images totally.

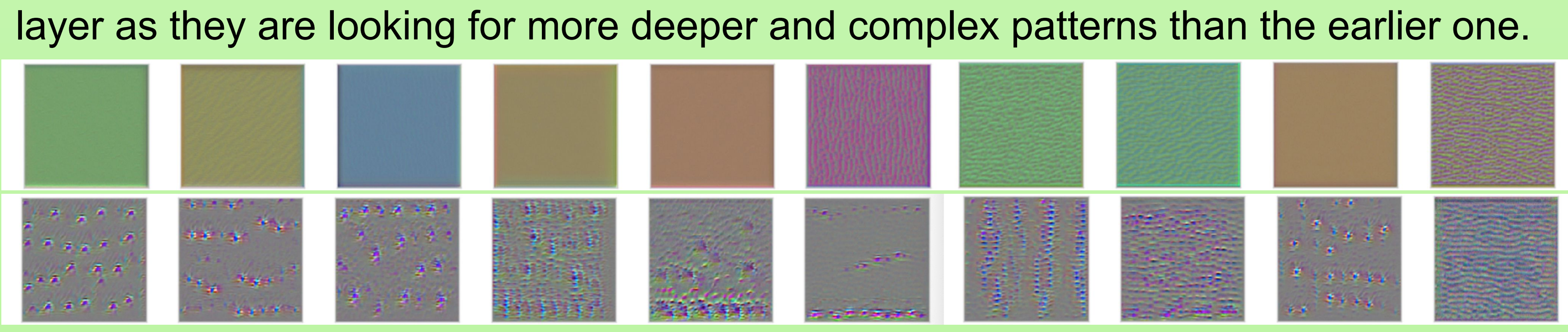
Model	Training Samples	Testing Samples	Training Accuracy	Training Loss	Testing Accuracy	Testing Loss
Support Vector Machine	3026	1297	0.55		0.55	
Random Forest	3026	1297	0.87		0.48	
K-nearest Neighbors	3026	1297	0.50		0.35	
Naive Neural network	3385	847	0.46	1.236	0.45	1.247
Custom CNN1	3385	847	0.94	0.160	0.81	0.612
Custom CNN2	3385	847	0.97	0.099	0.85	0.501
VGG	3026	1297	0.99	0.015	0.89	0.429
ResNet	3028	1295	0.93	0.197	0.88	0.470
DenseNet	3028	1295	0.98	0.053	0.87	0.818

Discussion

According to the results, the machine learning method is not the best choice for the image classification problem. Their accuracies are all around 0.5.
There are also some interesting findings about our Custom CNN when we train the model. Below is the visualization of the activation maps from first convolution layer. Different filters activate different parts of the image, like some are detecting edges, some are detecting background, while others are detecting just the outer boundary of the flower and so on.



Patterns from sixth convolution layer are more abstract than the simple one for first layer as they are looking for more deeper and complex patterns than the earlier one.



VGGNet has expected results, about 90% accuracy for 100 epochs, with smaller convolution kernel size, there are fewer parameters to train.
Applying ResNet and DenseNet in this classification project brings problem of overfitting. The best accuracy we made are 86.6% for DenseNet and 87.8% for ResNet.

Future

Clearly as the accuracy rate shows, deep learning model has some potentiality to improve. If we have more time working on this project, we are going to unfreeze the pre-trained ResNet and DenseNet layer, modify the following custom layers, change learning rates in order to try our best to solve this overfitting problem. We would also like to find whether loading new layers(top model) added in fine-tuning with proper weights and could give us a better accuracy or not.

Contact

- Jun Liu jul001@ucsd.edu
- Jiacheng Sun j5sun@eng.ucsd.edu
- Jiong Liu jil027@eng.ucsd.edu
- Bo Chen b6chen@eng.ucsd.edu

Reference

- [1] I. Steinwart and A. Christmann. Support vector machines. Springer Science and Business Media, 2008.
- [2] https://en.wikipedia.org/wiki/randomforest.
- [3] VGG 2014. https://arxiv.org/pdf/1409.1556.pdf.
- [4] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- [5] Dataset. https://www.kaggle.com/alxmamaev/flowers-recognition
- [6] Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.