**Introduction**

# **PKU Spring Semester CVDL homework**

Comparison between traditional method and deep Learning method

ID : 2102010644

Name : LEE DONG WOOK

College：Department of Engineering

We make a model classify Plant Seeding Using two method which are Traditional Method and Deep Learning method respectively. By comparing two method, check out each method of features and distinctions.

## **submission**

Traditional Method must have feature extraction and classifier. Deep Learning method must have network, optimizer, augmentation, reguralization. Check each charateristic property by comparing each method and model.

## **Dataset**

We use dataset of Kaggle (<https://www.kaggle.com/comp>etitions/plant-seedlings-classification ) trainset images are 4750, testset images are 794, and there are 200-600 images per class. Classes are consisted of 12 classes. Image pixel is all consisted of (3, 1899, 1900). More description is described in data\_analysis.ipynb in github. (https://github.com/dwsmart32/DL\_Model\_collect/blob/master/Plant\_Seeding/data\_analysis.ipynb)

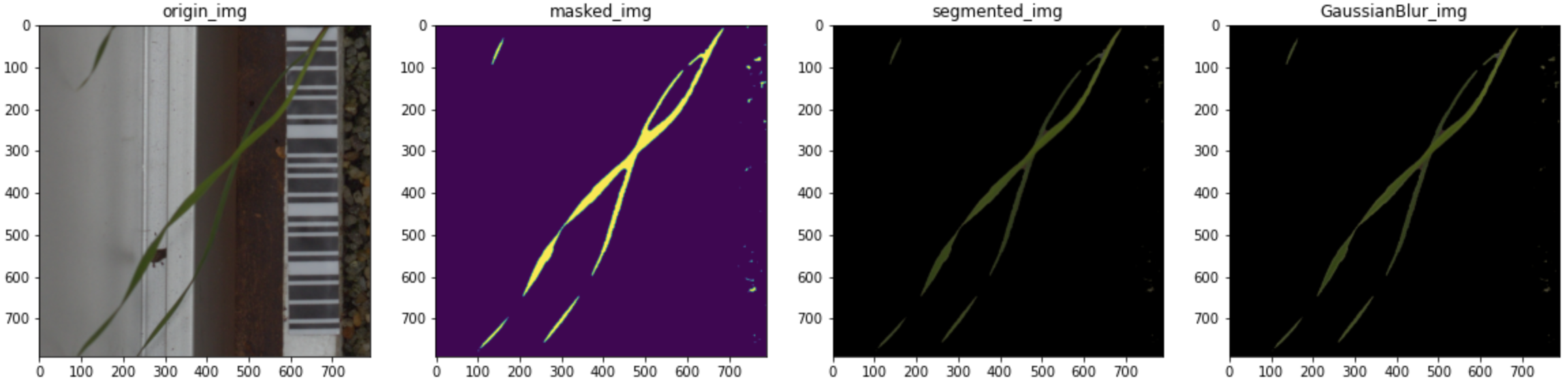
# **Traditional method**

At first, features and descriptor can be calculateed by using SIFT function in cv2 module. As Image below, because of Background, it can be seen that discriptors are not dotted on plant which is really important for classification. So deleting background method is used to high the performance.

텍스트, 전자기기이(가) 표시된 사진

자동 생성된 설명

Mask are calculated by using HSV features of each image. Black color is setted except for a plant area. Gaussian Blur are applied to make image sharpen. Performance Difference is compared in each method.



## **SIFT & K-mean Clustering with SVM**

All descriptor are caculated by SIFT method . All descriptor is vetorized and center point are calculated by using K-mean clustering method in condition which is that num\_cluster is 60 and image resize in (128, 256) . Support Vector Machine(SVM) model is used as training model. Linear model and Gaussian kernel are used for comparing. For exact comparing in same condition, seed is selected as 42 during Kmeans and svm training.

### Background without deleted

Images which are not deleted its Background are used. Number of Cluster is 60 for K-mean Clustering, C is 1, gamma is 0.008. Random state is 42. so that make a higest performance.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class name | Precision |  | Class name | Precision |
| Black-grass | 0.19 |  | Black-grass | 0.30 |
| Charlock | 0.16 |  | Charlock | 0.20 |
| Cleavers | 0.20 |  | Cleavers | 0.24 |
| Common Chickweed | 0.18 |  | Common Chickweed | 0.23 |
| Common wheat | 0.12 |  | Common wheat | 0.00 |
| Fat Hen | 0.18 |  | Fat Hen | 0.16 |
| Loose Silky-bent | 0.21 |  | Loose Silky-bent | 0.19 |
| Maize | 0.00 |  | Maize | 0.29 |
| Scentless Mayweed | 0.26 |  | Scentless Mayweed | 0.35 |
| Shepherds Purse | 0.00 |  | Shepherds Purse | 0.00 |
| Sf - Cranesbill | 0.13 |  | Sf - Cranesbill | 0.26 |
| Sugar beet | 0.35 |  | Sugar beet | 0.24 |
| Total(%) | 20.61 |  | Total(%) | 23.12 |

### Background deleted

Images which is deleted its Background are used. Nuber of Cluster is 60 for K-mean Clustering, C is 1, gamma is 0.008. Random state is 42. Hypreparameters are also tuned as well.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class name | Precision |  | Class name | Precision |
| Black-grass | 0.17 |  | Black-grass | 0.38 |
| Charlock | 0.55 |  | Charlock | 0.61 |
| Cleavers | 0.37 |  | Cleavers | 0.48 |
| Common Chickweed | 0.46 |  | Common Chickweed | 0.49 |
| Common wheat | 0.44 |  | Common wheat | 0.60 |
| Fat Hen | 0.50 |  | Fat Hen | 0.67 |
| Loose Silky-bent | 0.63 |  | Loose Silky-bent | 0.71 |
| Maize | 0.24 |  | Maize | 0.60 |
| Scentless Mayweed | 0.54 |  | Scentless Mayweed | 0.60 |
| Shepherds Purse | 0.32 |  | Shepherds Purse | 0.47 |
| Sf - Cranesbill | 0.70 |  | Sf - Cranesbill | 0.73 |
| Sugar beet | 0.44 |  | Sugar beet | 0.59 |
| Total(%) | 51.10 |  | Total(%) | 60.42 |

## **Hog with SVM**

Hog function in cv2 module is used for getting feature descriptor. Image is also resized in (128, 256) and changed in gray scale. SVM model is used as training model. Linear model and Gaussian kernel are used for comparing. For exact comparing in same condition, seed is selected as 42 during Kmeans and svm training.

### Background not deleted

Images which are not deleted its Background are used. Nuber of Cluster is 60 for K-mean Clustering, C is 6.6, gamma is 0.015. Random state is 42. Hyperparameter is tuned by a lot of steps so that make a higest performance.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class name | Precision |  | Class name | Precision |
| Black-grass | 0.12 |  | Black-grass | 0.29 |
| Charlock | 0.22 |  | Charlock | 0.28 |
| Cleavers | 0.13 |  | Cleavers | 0.29 |
| Common Chickweed | 0.27 |  | Common Chickweed | 0.28 |
| Common wheat | 0.09 |  | Common wheat | 0.50 |
| Fat Hen | 0.17 |  | Fat Hen | 0.21 |
| Loose Silky-bent | 0.17 |  | Loose Silky-bent | 0.24 |
| Maize | 0.25 |  | Maize | 0.56 |
| Scentless Mayweed | 0.31 |  | Scentless Mayweed | 0.40 |
| Shepherds Purse | 0.06 |  | Shepherds Purse | 0.00 |
| Sf - Cranesbill | 0.32 |  | Sf - Cranesbill | 0.32 |
| Sugar beet | 0.21 |  | Sugar beet | 0.19 |
| Total(%) | 20.50 |  | Total(%) | 26.99 |

### Background deleted

Images which is deleted its Background are used. Nuber of Cluster is 60 for K-mean Clustering, C is 6, gamma is 0.015. Random state is 42. Hypreparameters are also tuned as well.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class name | Precision |  | Class name | Precision |
| Black-grass | 0.36 |  | Black-grass | 0.39 |
| Charlock | 0.57 |  | Charlock | 0.71 |
| Cleavers | 0.46 |  | Cleavers | 0.65 |
| Common Chickweed | 0.53 |  | Common Chickweed | 0.65 |
| Common wheat | 0.28 |  | Common wheat | 0.54 |
| Fat Hen | 0.53 |  | Fat Hen | 0.68 |
| Loose Silky-bent | 0.67 |  | Loose Silky-bent | 0.70 |
| Maize | 0.51 |  | Maize | 0.63 |
| Scentless Mayweed | 0.55 |  | Scentless Mayweed | 0.67 |
| Shepherds Purse | 0.37 |  | Shepherds Purse | 0.53 |
| Sf - Cranesbill | 0.81 |  | Sf - Cranesbill | 0.84 |
| Sugar beet | 0.58 |  | Sugar beet | 0.72 |
| Total(%) | 54.18 |  | Total(%) | 67.36 |

# **Deep Learning Method**

## **Model**

We used transfer learning for saving time.

Model : Resnet 50

Params size (MB): 97.49

Estimated Total Size (MB): 1415.87

## **Optimizer**

3 kinds of Optimizer which are SGD, Adam, Adagrad are tried. But Adam was the most efficient. So, Adam has been chosen in this model. We also set learning rate as 0.002 by tuning hyperparameter.

Optimizer : Adam (lr : 0.002, betas : (0.5, 0.99))

## **A**

## **ugmentation**

Firstly, I use resizezing method. Since Resnet50 network is based on (224,224) image. We make input images to (224, 224) size.

Secondly, We make all pictures normalize. All dataset of pixel mean value is 0.3 and standard variation value is 0.1. Using these values, we make all pixels value normalize for enhancing speed and performance.

Thirdly, We also use Centercrop, GaussianBlur, Flip, but performace is rather worse. So only resizing and normalization is seleted as augmentation . Details are in data\_analysis.ipynb file.

Resize : (224, 224)

Normalizez(RGB) : ((0.3, 0.3, 0.3), (0.1. 0.1, 0.1))

## **Regularization**

We use weight decay for avoiding weight differs dramatically and set gamma value as 0.1.

Weight decay(gamma) : 0.1

## **Other Specification**

For Hyperparamter tuning torch.manual\_seed(10) is used. CrossEntropyLoss is also used as loss function. epochs value is 30. Batch size is 32 and output Dense Layer is 12. Experiment is done in condition which is AMD Ryzen 5 3600 CPU, NVDIA GeForce GTX 1660 SUPER.

## **Performance and Result**

Accuracy and Loss of test and valid data is below

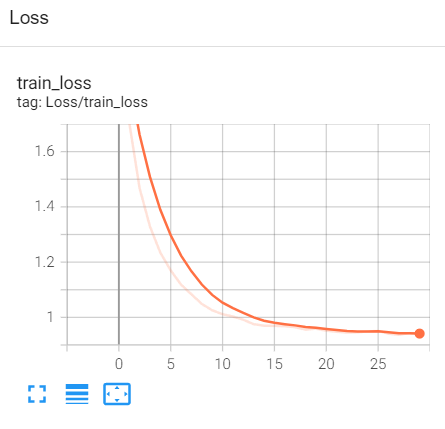


Figure 2 Loss of traindataset

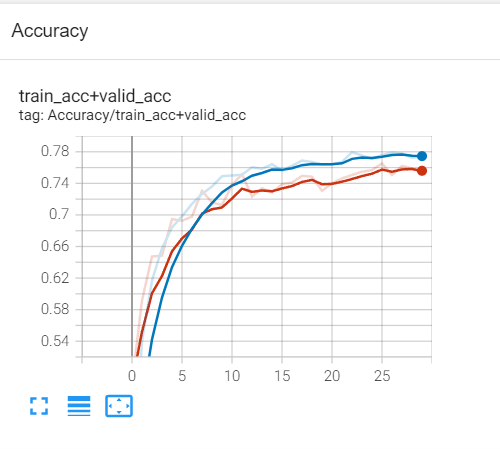


Figure 3 Accuracy of datasets

Train dataset loss : 0.94 (at epoch 30)

Train dataset acc : 0. 78

Valid dataset acc : 0.77

Testdata set is tested and result is recored in sample\_submission.csv file. Submission name is DongwookRhee.

Test dataset acc : 0.70780

Spend time in training : 570 (sec)

# **Conclusion**

# **References**

1. Pytorch\_tutorial:https://pytorch.org/tutorials/beginner/blitz/cifar10\_tutorial.html
2. tensorflow\_tutorial:https://www.tensorflow.org/tutorials/images/classification
3. vlfeat开源library: https://www.vlfeat.org/
4. sklearn开源library: https://scikit-learn.org
5. opencv开源library: https://opencv.org/
6. dwsmart32 github : https://github.com/dwsmart32/DL\_Model\_collect