6-class image classification

EE4146
Group 22

Name	SID	Role
Wong Ka Ho	54816829	Build code and write report
Wong Ho San	55696272	Write report and research
Yeung Man Hin	56221880	Write report and research
Chiu Tsun Hei	55683725	Write report

Abstract

A Convolutional Neural Network (CNN) is a deep learning algorithm that is able to identify or differentiate objects. With the powerful image processing and deep learning capabilities, It is commonly used in image recognition. In this project, CNN model will be used for image classification, while densenet and data augmentation will be adopted to improve the accuracy of image classification results.

Introduction

Image classification is a supervised-learning in the region of machine learning by training the model with a set of labeled training data. The model aims to map the input (images) to the output (categories) based on the training data provided.

In this project, we are attempting to train a model to solve the image classification problem and classifying the images into six classes : building, forest, sea, street, mountain and glacier. In order to classify the image, the convolutional neural networks(CNN) is used. The csv file contains the corresponding labels of each training image and each of them only contains a single label.

However, using the basic training model may obtain a less accurate result in classifying the image. In this project we are going to try different methods to obtain a higher accuracy of classification for each class of image such as data augmentation and residual network.

Methodology

Data augmentation

For improving accuracy of the image classification, we need to perform data augmentation. Due to the fact that more data used for training will get better results. Data augmentation can provide more training data. For example, we can change the dataset slightly like flips, translations, and rotation etc. The training model will recognize that augmented data is different data but not the same as before.[1]

In convolutional neural networks(CNN), If the network can classify the image stably even if the view of the object is changed, we can call it immutability. More precisely CNN is immutable for translation, viewpoint, size, illumination etc.[2] Back to our dataset, we have a batch of dataset shot in limited scenes. But our goal is applying the trained model in different situations. For example, In different directions, position, brightness, scaling ratio etc. Thus, we can train the CNN model through extra synthetic data to predict those situations.

In our model result, I found that the model has a high chance to misclassify “glacier” and “mountain” . We think the reason for this phenomena is that machine learning will find the most significant feature to classify the images. In our dataset, the most significant feature of the “glacier” and “mountain” is “glacier” mainly in white and mountain is not. In addition there is a very important feature affecting the landscape of the photo and which is the season.

Humans can easily classify the photo for which season but CNN does not understand that the landscape can contain different situations like clouds, snow, or wetness. The network will label the mountain on cloudy days with glaciers. To solve the above problem, we tried to adopt conditional GANs which can convert the image to different areas.[3] For example GANs can convert the summer view photo to a winter view.

Example of GANs

Feature extraction

Feature extraction is a process that extracts a feature which is beneficial to image classification.[4] Then, describe the image to the designed model by considering the extracted feature. For example color and shape of image. Thus, the model can classify the images by considering the extracted features. Feature extraction can build up an informative dataset to help the model get decisions easily with accuracy and completeness of the dataset. In the following, we will use wide resnet and densenet to perform feature extraction.

ResNet

ResNet can build up a very deep network through the residual network structure. ResNet solved the common problem, Vanishing gradient, in CNN by build up a shortcut connection between output and input of each residual block.[5]

DenseNet

The main idea of densenet is to create short paths from early layers to later layers. The architecture of densenet is combined with many dense blocks. [6]The target of dense blocks is to decrease the number of feature maps. Thus, dense blocks can do the dimensional calculation while merging the features of each channel. Due to the designation of dense block, DenseNet can train easily under feature and vanish transferring efficiently.

Image classification

Support vector machine

We mainly adopt a support vector machine to act as an image classifier. The basic idea of SVM is to find a separation plane to divide two sets of data with maximal interval.[7] One of the advantages of SVM is that the user can switch the kernel of SVM easily. Thus, we tried to use different kernels of SVM to perform image classification. In conclusion, kernel rbf is better than linear kernel in this image classification mission. Thus, we will adopt the rbf kernel in the following result analysis.

Stochastic Gradient Descent

SGD is an algorithm that aims to minimize the risk function and loss function. The basic idea of SGD is to calculate the sum of real sample values and predict values from the dataset. Within the calculation all the samples are selected randomly. If the dataset contain many peak value, SGD could get a local optimal solution.[8]

Results

wide_resnet + SVM(original) :

```
svmclf = svm.SVC(kernel='rbf')
svmclf.fit(trainX, trainY)
predY_svm = svmclf.predict(valX)
print(predY_svm)
acc_svm = metrics.accuracy_score(valY, predY_svm)
print("rbf svm validation accuracy =", acc_svm)
```

```
[5 4 5 ... 1 4 3]
rbf svm validation accuracy = 0.9287619047619048
```

wide_resnet + SGD :

```
sgd_clf = SGDClassifier(random_state=11, max_iter=1000, tol=1e-3)
sgd_clf.fit(trainX, trainY)
predY_sgd = sgd_clf.predict(valX)
acc_sgd = metrics.accuracy_score(valY, predY_sgd)
print("Kernel PCA sgd validation accuracy =", acc_sgd)
```

```
Kernel PCA sgd validation accuracy = 0.9203809523809524
```

densenet + SVM :

```
In [39]: svmclf = svm.SVC(kernel='rbf')
         svmclf.fit(trainX, trainY)
         predY_svm = svmclf.predict(valX)
         print(predY_svm)
         acc_svm = metrics.accuracy_score(valY, predY_svm)
         print("rbf svm validation accuracy =", acc_svm)
```

```
[2 2 0 ... 3 5 1]
rbf svm validation accuracy = 0.932952380952381
```

densenet + SGD :

```
from sklearn.linear_model import SGDClassifier
sgd_clf = SGDClassifier(random_state=11, max_iter=1000, tol=1e-3)
sgd_clf.fit(trainX, trainY)
predY_sgd = sgd_clf.predict(valX)
acc_sgd = metrics.accuracy_score(valY, predY_sgd)
print("Kernel PCA sgd validation accuracy =", acc_sgd)
```

```
Kernel PCA sgd validation accuracy = 0.9154285714285715
```

By considering the kaggle result, we improved the accuracy score after adopting DenseNet and data augmentation.
Original : 0.78909
Apply DenseNet : 0.7909
Apply data augmentation : 0.79636

Conclusion

To conclude, this project can be divided into three main parts: data augmentation, feature extraction, and image classification. First, data augmentation can provide more data to train the model, which can let it obtain better results. When the convolutional neural networks(CNN) is immutable for changes in the dataset, including translation and illumination, the model which is trained with extra synthetic data can still predict regardless of the situations. Unlike human beings, CNN is unable to understand the weather information a landscape contains, thus conditional GANs have been adopted to help the model perform better in distinguishing “glacier” and “mountain”. Second, in the stage of feature extraction, features that are constructive to image classification, such as colors and shapes, have been extracted by using wide ResNet and DenseNet. The informative dataset can significantly help the model to get accurate decisions easily. Third, for the image classification part, the support vector machine(SVM) is used as an image classifier, which works by separating two sets of data with maximum interval. Kernels of SVM can be switched easily, after trying different kernels, kernel rbf is found to be better than linear kernel when it comes to image classification. Meanwhile, Stochastic Gradient Descent(SGD) can be used to minimize the risk function and loss function. Finally, as the results showed, the accuracy score is improved after adopting DenseNet and data augmentation, which increases from 0.78909 to 0.79636.

Reference

[1] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, 6(1), 1-48.
[2] Hussain, M., Bird, J. J., & Faria, D. R. (2018, September). A study on cnn transfer learning for image classification. In UK Workshop on computational Intelligence (pp. 191-202). Springer, Cham.

[3] Yi, X., Walia, E., & Babyn, P. (2019). Generative adversarial network in medical imaging: A review. Medical image analysis, 58, 101552.
[4] Nixon, M., & Aguado, A. (2019). Feature extraction and image processing for computer vision. Academic press.
[5] He, F., Liu, T., & Tao, D. (2020). Why resnet works? residuals generalize. IEEE transactions on neural networks and learning systems, 31(12), 5349-5362.
[6] Zhang, C., Benz, P., Argaw, D. M., Lee, S., Kim, J., Rameau, F., ... & Kweon, I. S. (2021). Resnet or densenet? introducing dense shortcuts to resnet. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 3550-3559).
[7] Chen, Y., Zhou, X. S., & Huang, T. S. (2001, October). One-class SVM for learning in image retrieval. In Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205) (Vol. 1, pp. 34-37). IEEE.
[8] Ketkar, N. (2017). Stochastic gradient descent. In Deep learning with Python (pp. 113-132). Apress, Berkeley, CA.