Font Classification Using Multiple Classifiers

Project Proposal of Group 7

Chen Hanzhao
School of Mathematics,
Peking University
1500010703

Li Hongcheng
National School of Development,
Peking University
1600011374

Liu Taide
School of Electronic Engineering
and Computer Science, Peking University
1500012923

I. Introduction

Image classification is one of the basic problems discussed in the field of computer vision and is a traditional type of pattern recognition problem, which means we can apply most of the pattern recognition methods we have learned/will learn on this problem. Through this project, we hope to reach a clearer and deeper understanding of the content introduced in this course.

Font classification is similar to other multi-class classification problems, such as digits recognition, scene classification, etc. We get an image of a character, and the goal is to tell which font it is. The characters may be printed or handwritten, and the images are scanned from a variety of devices or generated by computer. These factors make the solution more complicated.

II. Dataset

The dataset we plan to use is from UCI Machine Learning Repository, you can find the dataset and its description here: http://archive.ics.uci.edu/ml/datasets/Character+Font+Images

It contains 745000 instances of 153 character fonts. This scale is suitable for both traditional methods and deep neural networks. But the data is stored in CSV format which is not easy to read by programs, and therefore extra pre-processing of converting it to binary file is needed.

This dataset is also used in a machine learning textbook, which provides codes of many kinds of machine learning methods. But the pre-processing codes it provides seem too complicated and the methods part is also based on popular libraries. Therefore, we will figure out this project from scratch and promise not copying from Git.

III. Preprocessing

Most of the characters in the dataset are special characters in Unicode rather than digits and English letters. For example, mathematical operators like '$\neq$','$\int$','∂', geometries like '$\Diamond$', others like EUR symbols, trademark symbols, etc. Too many characters, especially those that seem similar to each other, will not only bring prohibitively long training time and rather costly hardware requirements, but also deficiency of accuracy. Therefore, we would only use the ANSI characters and use some augmentation methods like reshape, re-sample, padding and crops to expand the dataset, if needed.

The scale of one character is also an important feature. For example, there is a clear difference between the scale of character 'f','i' of monospaced font and non-monospaced font. But the characters in the dataset are all re-sized to the same scale. Therefore, We will use the originally provided size to recover these characters.

IV. Algorithms

Image classification is usually divided into three stages in traditional modelsfeature extraction, encoding, and classification. The first two stages are relatively irrelevant to pattern recognition, thus we would simply use the HoG or SIFT [Lowe et al.(2004)] descriptors (depending on computing time) as image features, and Fisher Vector [Snchez et al.(2013)] as encoding methods. These methods are widely and elaborately implemented in many popular CV or ML libraries, and we will use them directly in order to focus more on the classification part. The encoding methods will affect whether the feature is easy to separate, and consequently the comparison between using Fisher vector as encoding and simply flatting the vectors should be tested.

As for the classification stage, we would like to try most of the methods introduced in the course, including but not limited to Bayesian inference, multi-class SVM [Hsu et al.(2002)], decision tree (or random forest [Liaw et al.(2002)]), and clustering. Undoubtedly, to acquire best comprehension of these methods, we need to carry out very detailed process of feature selection, parameter setting as well as outcome evaluation. For instance, when we apply SVM, the weight allocated to regularization term will certainly affect outcome, for there exists a trade-off that the decision surface will be unreasonable given a small weight whereas the sample will be underfitted given a big one. Likewise, decision tree sometimes manifests itself as very sensitive to selection of features, thus different sets of features will be considered (, and we will also apply random forest, since it has comparatively robust property in this matter.) All the attempts that we will exert need to be evaluated by multiple indicators, which we will present later.

As we all know, CNN is the state of the art in the field of computer vision, as ResNet [He et al.(2015)] beats human in image classification in the dataset of ImageNet, and DeepLab with dense CRF [Chen et al.(2018)] reaches a pretty high accuracy in scene segmentation. Based on these, we will surely not doubt that CNN beats traditional methods in most problems. However, does the improvement of accuracy worth the extra costs that CNN brings, such as longer training time, and higher computation costs? Obviously, our font classification problem seems to be such a simple problem compared to those mentioned above that DCNN like ResNet is far more complicated and the accuracy increment it brings will not be worth the extra costs it imposes. Therefore, we would only use a little modified LeNet and ResNet-18 to do the comparison.

V. Evaluation

The main aspect of evaluation that we consider is accuracy. Given that the number of classes is relatively high, the top-k accuracy (if available) should be a more proper evaluation. Precision and recall (F1-score) will be also considered if the dataset is relatively biased. The computation costs, such as training time, inference time, model size, etc., will also be compared, because one may focus on different aspects within distinct application scenarios.

The dataset does not provide the test dataset. Therefore, we would randomly choose one tenth of the data as test dataset and one tenth as validation dataset (used only in CNN).

References

[Lowe et al.(2004)] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91-110.

[Snchez et al.(2013)] Snchez, J., Perronnin, F., Mensink, T., & Verbeek, J. (2013). Image classification with the fisher vector: Theory and practice. International journal of computer vision, 105(3), 222-245.

[Zheng et al.(2016)] Zheng, L., Yang, Y., & Tian, Q. 2016, arXiv:1608.01807

[Liaw et al.(2002)] Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R news, 2(3), 18-22.

[He et al.(2015)] He, K., Zhang, X., Ren, S., & Sun, J. 2015, arXiv:1512.03385

[Chen et al.(2018)] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. 2018, arXiv:1802.02611

[Hsu et al.(2002)] Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. IEEE transactions on Neural Networks, 13(2), 415-425.