华南理工大学

South China University of Technology

The Experiment Report of *Deep Learning*

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

Author:
Qichen Huang

Supervisor:
Mingkui Tan

Student ID:
201920142806

Grade:
Graduate

November 10, 2019

Face Classification Based on AdaBoost Algorithm

***Abstract*—Face Classification is a problem about determining whether it contains a human face or not. We present details of AdaBoost algorithm and NPD feature of images, and use them to deal with Face Classification problem. Final accuracy of our AdaBoost model reaches 95%, which is a satisfying result.**

I. Introduction

IMAGE Classification is a classification problem that it tries to figure out which class an image belongs to. Face Classification is a specific task among numerous Image Classification problems. Its objective is to classify an image whether it contains a human face or not. In this experiment, we solve this problem with AdaBoost algorithm, short for Adaptive Boosting, which is an ensemble technique that attempts to create a strong classifier from a number of weak classifiers. Because image pixels data is unsuitable for common classification models, we first extract the Normalized Pixel Difference(NPD) feature of images before feeding them into AdaBoost model.

Our motivation is to 1) understand AdaBoost further, 2) get familiar with the basic method of face classification, 3) learn to use AdaBoost to solve the face classification problem and combine theory with actual project, 4) experience the complete process of machine learning. We carry out this experiment on 1000 provided images, half of them contain faces while the others do not.

II. Methods and Theory

AdaBoost is one of the most famous representatives of Boosting algorithms, a serial ensemble method in which all base classifiers are trained one by one. Since the key factors affecting performance of AdaBoost algorithm is the accuracy and diversity of base classifiers, AdaBoost learns every base classifier based on samples distribution modified according to previous one, which enable model to pay more attention on mispredicted data. The details of AdaBoost algorithm is presented in Table I.

In the algorithm, $D_{1\ldots T}$ is samples distribution at each iteration, ϵ_t and α_t are error rate and weight of each base classifier separately.

Normalized Pixel Difference(NPD) is a kind of feature extraction method for image, which is efficient to compute and has several desirable properties, including scale invariance, boundedness, and enabling reconstruction of the original image. The NPD feature between two pixels in an image is defined as $f(x,y) = \frac{x-y}{x+y}$, where $x, y \geqslant 0$ are intensity values of the two pixels, and $f(0,0)$ is define as 0 when $x = y = 0$. Then, we compute NPD feature for every pairs of pixels in an image and gather them together, which forms NPD feature of the image.

TABLE I: AdaBoost Algorithm

Input:	training set $D = (\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)$
	base classifier $\mathcal{B}$
	number of iteration T
Steps:	
1:	$D_1(\boldsymbol{x}) = \frac{1}{m}$
2:	**for** $t = 1, 2, \ldots, T$ **do**
3:	$h_t = \mathcal{B}(D, D_t)$
4:	$\epsilon_t = P_{x\sim D_t}(h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}))$
5:	**if** $\epsilon_t > 0.5$ **then break**
6:	$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
7:	$D_{t+1} = \frac{D_t(\boldsymbol{x})}{Z_t} \times \begin{cases} \exp(-\alpha_t) & \textit{if } h_t(\boldsymbol{x}) = f(\boldsymbol{x}) \\ \exp(\alpha_t) & \textit{if } h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}) \end{cases}$ $= \frac{D_t(\boldsymbol{x})\exp(-\alpha_t f(\boldsymbol{x}) h_t(\boldsymbol{x}))}{Z_t}$
8:	**end for**
Output:	$H(\boldsymbol{x}) = \mathrm{sign}(\sum_i^T \alpha_t h_t(\boldsymbol{x}))$

III. Experiments

A. Dataset

The dataset of this experiment consists of 500 human face images with 250×250 pixels and 500 other images including animals, vehicle, etc, with 32×32 pixels. After mixing up, two thirds(667) of the total images are divided into training set and the rest into validation set.

B. Implementation

The experiment steps of AdaBoost algorithm are as follows:

1) Load images data and convert it into grayscale images with size of 24 * 24.
2) Process data set data to extract NPD features. Save the feature data in a file for future utilization.
3) divide the data set into training set and validation set.
4) Write all AdaboostClassifier functions based on the reserved interface in ensemble.py. The following is the guide of fit function in the AdaboostClassifier class:
    a) Initialize training set weights ω, each training sample is given the same weight.
    b) Train a base classifier, as which we use sklearn.tree library DecisionTreeClassifier.
    c) Calculate the classification error rate ϵ_t of the base classifier on the training set.
    d) Calculate the parameter α according to the classification error rate ϵ.
    e) Update training set weights ω.
    f) Repeat steps 4.2-4.6 above for iteration, the number of iterations is based on the number of classifiers.
5) Predict and verify the accuracy on the validation set using the method in AdaboostClassifier and use classification_report of the sklearn.metrics library function writes predicted result to *classifier_report.txt*.

In this experiment, the weights of samples ω are initialized in Uniform Distribution. The *max_leaf_node* parameter in DecisionTreeClassifier is initialized as 3 and the number of base classifiers is initialized as 8.

Final result of classification output in *classifier_report.txt* is presented in Table II.

TABLE II: Classifier Report

	precision	recall	f1-score	support
class nonface	0.95	0.96	0.95	172
class face	0.96	0.94	0.95	158
accuracy			0.95	330
macro avg	0.95	0.95	0.95	330
weighted avg	0.95	0.95	0.95	330

IV. Conclusion

In this paper,we introduced Image Classification and Face Classification roughly and presented the details of AdaBoost algorithm and NPD feature of images. Then, we pictured the experiment that implements Face Classification by AdaBoost with NPD feature extracted from images. As is shown in Table II, the final accuracy rate of total validation data is 95%, which is a satisfying result.