Automatic Target Recognition for CT-based Airport Screening System

Chen Gu¹, Zhen Dong²

¹ Electrical and Computer Engineering, Marquette University, Milwaukee, Wisconsin, 53233, United States
² Electrical and Computer Engineering, Marquette University, Milwaukee, Wisconsin, 53233, United States

Abstract. To construct an automatic target recognition CT-based screening system for airport, this paper introduces a method by combining two hot technology, CT Image Processing and Machine Learning together. With the grey-scale features and the Histogram of Oriented Gradient (HOG) features extracted from the CT images, we can train different classifier (SVM, KNN) to recognize the targets (saline, rubber, clay) we want. By comparing the recall rate and precision rate of each classifier, we may find the best classifier for this project.

1. Introduction
NOWADAYS, CT-based target recognition screening system is playing an increasingly vital role in the airport security system. Compared with the traditional X-ray scanner, CT target scanner is more sensitive, comprehensive and faster. Furthermore, combining with the machine learning technique, the recognition rate of the CT scanner can be increasingly improved. There are two main steps for CT automatic target recognition, one is CT image feature extraction. Unlike the X-ray image, each CT image usually contains hundreds or thousands of slices, which make the whole extraction process much complex. The other step is the choice of classifier which is directly connected to the learning performance. As a result, we need to choose proper extraction algorithms and classifiers to reach an ideal recall and precision rate.

2. Feature Extraction
Image features are basic characteristics to distinguish different images, and image feature extraction is a process we change the characteristic of the image into numerical data. There are three main kinds of image features: grey-scale feature, texture feature, shape feature. In the project, we choose former two as our CT image feature extraction algorithms.

2.1. Grey Scale Feature
Grey-scale feature extraction is a classical algorithm in digital image processing, since this kind of feature only include the statistic information of grey-scale distribution, which means it is insensitive to size, orientation and has strong robustness. In the project, we use 6 kinds of statistic value:

1) Mean:

\[ \mu = \sum_{i=0}^{t-1} iH(i) \]

2) Variance:
\[ \sigma^2 = \sum_{i=0}^{l-1} (i - \mu)^2 H(i) \]

3) **Skewness:**

\[ \mu_z = \frac{1}{\sigma^3} \sum_{i=0}^{l-1} (i - \mu)^3 H(i) \]

4) **Kurtosis:**

\[ \mu_k = \frac{1}{\sigma^4} \sum_{i=0}^{l-1} (i - \mu)^4 H(i) - 3 \]

5) **Energy:**

\[ \mu_W = \sum_{i=0}^{l-1} H(i)^2 \]

6) **Entropy:**

\[ \mu_E = -\sum_{i=0}^{l-1} H(i) \log_2[H(i)] \]

Where \( H(i) \) is the grey-scale level of each pixel. For a CT image, the grey-scale level is from 0 to 255, and 0 usually represents the background which is no use for our feature extraction, so in the calculation we only set \( i \) from 1 to 255. After the extraction, we got a 1005*6 feature matrix.

### 2.2. Histogram of Oriented Gradient Feature (HOG)

In feature extraction, we not only need the statistic information, but also texture information from the image, since two images with same statistic value may be totally different things. To solve this, we use HOG feature because the appearance and the shape of an image can be described clearly by the edge directions. For HOG algorithm, we first divide the image into small connected region called cells, then for the pixels in each cell, a histogram of gradient direction is compiled and the concentration of these histograms is the descriptor. For better accuracy, we can contrast-normalize the local histograms by calculating a measure of the intensity across a larger region of the image named block, then we can use this value to normalize each cell. As the figure shows below, after HOG feature extraction, a boundary was drew around the target which include the edge direction information as figure.1. shows:

![Figure 1](image-url)

**Figure 1.** A slice after HOG extraction
3. Classifier

In the choice of classifier, the SVM and KNN that we chose are both binary classifiers. Since after analyzing and understanding the data, we found that the data samples can be simplified into two types which are “target” and “non-target”. Therefore, the classification question can be easy to achieve.

3.1. SVM

3.1.1. Background and Theory

The support vector machine was first proposed by Cortes and Vapnik in 1995. It showed many unique advantages in solving small sample, nonlinear and high-dimensional pattern recognition, and was applied to other machine learning problems such as function fitting. The basic idea of SVM is to define the optimal linear superface and to attribute the algorithm for finding the optimal linear hyperplane to a convex programming problem as figure. 2. shows. Based on the Mercer kernel expansion theorem, by constructing a nonlinear mapping $\Phi$, the sample space is mapped to a high-dimensional or even infinite dimensional mapping space, so that the linear learning machine can be applied to the feature space to solve the high nonlinear classification and regression problems in the sample space. The SVM method is proposed from the optimal classification plane in the case of linear separability. The most common classification is to require the classification line to accurately separate the two types of samples and maximize the distance between the two types.

3.1.2. SVM in MATLAB

In this project we directly used the SVM classifier in the statistics and machine learning toolbox in MATLAB. The function “fitcsvm” is what we used in our project. Here, we input our training data and training label. At the same time, we need to change some parameters to make our classifiers show better results. First, we need to standardize our data, Scaling the raw data to between 0 and 1 based on the ratio to better match the normal distribution. Then, we need to choose our kernel function. For grey-scale feature, we use rbf function; for the HOG feature, because of the high dimension, we use linear function. Finally, we have a 10-fold cross-validation for our model to get the accuracy which shows the performance. Figure.3. shows the whole training process.
3.2. KNN

3.2.1. Background and Theory
As figure 4. shows, the KNN algorithm works by selecting one sample and selecting the k samples closest to the sample. Among the k samples, the positive and negative samples are the proportions of the selected samples. If there are more positive samples, the selected samples are positive; if there are more negative samples, the selected samples are negative.
3.2.2. KNN in MATLAB
The function “fitcknn” is what we used in our project. Here, we input our training data and training label. At the same time we need to change some parameters to make our classifiers show better results. First, we need to standardize our data, it should be the first thing to do. Then, we have to define how much near samples we need to use. Moreover, we need to choose a kind of distance between neighbour and samples for the classifier to calculate. Finally, always remember to use cross-validation to show our model’s performance.

4. Result

4.1. Grey-scale feature with SVM

Recall Rate:0.43089 53/123
Precision Rate:0.83721 288/344
Saline Recall Rate:0.4878 20/41
Rubber Recall Rate:0.31818 14/44
Clay Recall Rate:0.5 19/38

As it can be seen from the figure, the recognition rate of the target (recall rate) is 43%, and the recognition rate of the non-target (precision rate) is 83.7%. For the specific target, clay has the best recall rate, rubber has the lowest recall rate.

4.2. HOG feature with SVM

Recall Rate:0.42276 52/123
Precision Rate:0.88372 304/344
Saline Recall Rate:0.43902 18/41
Rubber Recall Rate:0.38636 17/44
Clay Recall Rate:0.44737 17/38

As it can be seen from the figure, the recognition rate of the target (recall rate) is 42.2%, and the recognition rate of the non-target (precision rate) is 88.3%. For the specific target, clay and saline have the same best recall rate, rubber has the lowest recall rate.

4.3. Grey-scale feature with KNN

Recall Rate:0.7561 93/123
Precision Rate:0.31395 108/344
Saline Recall Rate:0.56098 23/41
Rubber Recall Rate:0.97727 43/44
Clay Recall Rate:0.71053 27/38
As it can be seen from the figure, the recognition rate of the target (recall rate) is 75.6%, and the recognition rate of the non-target (precision rate) is 31.3%. For the specific target, rubber has the best recall rate, saline has the lowest recall rate.

5. Conclusion
The project result we completed this time is not ideal. It can be seen from the results that the SVM is good for non-target object recognition, and KNN is good for target object recognition. So this is a reminder to us that no algorithm is perfect, and each algorithm has its own advantages and disadvantages. We will use additional algorithms later to combine their results, exploit the strengths of each algorithm, and abandon their deficiencies to get better results.

References
[1] https://www.mathworks.com/help/stats/fitcecoc.html
[2] https://www.mathworks.com/help/vision/ref/extracthogfeatures.html
[3] http://www.docin.com/p-240514695.html
[4] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.
[5] Zhang, Min-Ling, and Zhi-Hua Zhou. "ML-KNN: A lazy learning approach to multi-label learning." Pattern recognition40.7 (2007): 2038-2048.
[6] http://www.cqvip.com/read/read.aspx?id=46293110#