# Automatic Target Recognition(ATR) for CT-based Airport Checkpoint Screening

Cassandra Lundberg
*Department of Computer Science*
*Georgia State University*
Atlanta, Georgia
United States
cassandral1999@gmail.com

Nicholas Albano
*Department of Computer Science*
*Georgia State University*
Atlanta, Georgia
United States
nalbano1@student.gsu.edu

***Abstract*—This paper introduces the construction of an automatic target recognition system to detect particular targets (saline, rubber, clay) in ct-scanned images. The system utilizes methods such as feature extraction of statistical information taken from the images and histogram feature extraction of pixel intensities for each scanned image. With these features extracted, different models can be trained on the image information to identify these targets. Each model than then be evaluated by comparing the accuracy, precision, and recall rate for detecting targets. The best model is used for predicting targets in other, unseen images.**

***Index Terms*—introduction, data visualization, feature extraction, model evaluation, conclusion**

## I. Introduction

With safety and security being the primary concern at airports, the issue arises with time constraints from manual luggage checks. An automatic target recognition system helps to quicken the process of bag checks by sending each bag through a computer tomography(CT) scanner machine. As the X-ray mechanism revolves around the luggage, the CT-scanner records data and compiles a detailed tomogram or slice of the bag. Multiple slices of the bag are constructed into a single CT-scanned volumetric image. This allows for a more accurate representation of the three-dimensional objects in the bags and gives ease to the automatic target recognition system to evaluate the slices. The ATR system can then extract information from each slice of a CT-image such as the dimensions, mean, standard deviation, variance, skew, and kurtosis of the image. More information can be extracted from the images using the histograms of the image gray-scale pixel intensities. As this information now explains the composition and contents of each bag, a model classifier can be trained to recognize targets with the given information. However the question arises of which model performs the best to detect particular targets when given a data-set describing the objects located inside luggage. The model classifiers analyzed in this paper are the Support Vector Machine, Logistic Regression Classifier, AdaBoost Classifier, and K-Nearest Neighbors Classifier. Each of these classifiers use supervised machine learning techniques to classify a target variable and can be hyper-tuned to improve performance.

## II. Data Visualization

The data-set used for constructing this automatic target recognition system consists of 1005 training and 467 testing CT-images.Each CT-scanned image is considered a volumetric image, meaning it has three dimensions. The x dimension explains the length, the y dimension is interpreted as the height, and the z dimension or width is the number of slices for each image.

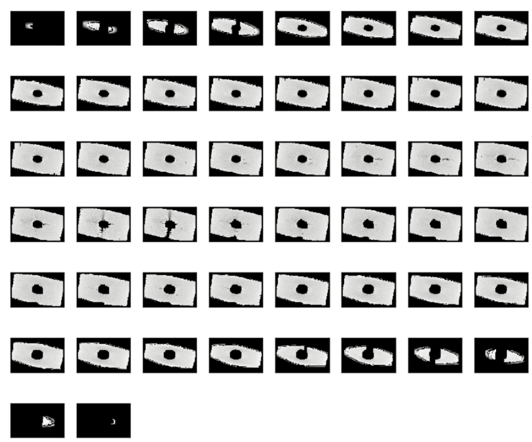

Fig. 1. Slices of first CT-image in the data-set

As shown in “Fig. 1”, this CT-scanned image has 50 slices and the total dimensions are (89,140,50). Each slice exists in a gray-scale format which helps to extract features.

## III. Feature Extraction

Feature extraction is the process of reducing the number of resources, in this case pixels, required to describe a large set of data. In an image, each pixel is considered a feature which entails a specific amount of information about that image. Since CT-scanned images are in gray-scale, each feature is a numerical value ranging from 0-255. Zero indicates the darker pixels or background and 255 indicates the lightest pixels or a piece of an object. There are two types of features that can be extracted from these gray-scale images; the statistical features and the histogram features.

### A. Statistic Features

Since each image is already in gray-scale, statistical information can be extracted from the gray-scale distribution. The statistical features that are extracted from the gray-scale distribution of an image are insensitive to size and orientation of an image and seem to already be very robust. Because of this, image processing does not need to be done before computing these features. However, since the zero gray-scale pixels indicate the background noise of an image and can calculate imprecise statistics, the range of pixels is set from 1-255. This helps to eliminate the background noise.

### B. Statistical Features Calculated

- x,y, and z dimension of each image
- Maximum, Middle, Minimum dimension of each image
- Median
- Mean

$$\mu = 1/n \sum_{i=0}^{n} iG(i) \tag{1}$$

- Standard Deviation

$$\sigma^2 = \sqrt{1/n \sum_{i=0}^{n} (i-\mu)^2 G(i)} \tag{2}$$

- Variance

$$\sigma^2 = 1/n \sum_{i=0}^{n} (i-\mu)^2 G(i) \tag{3}$$

- Skewness

$$\mu_s = 1/((n-1) * \sigma^3) \sum_{i=0}^{n} (i-\mu)^3 G(i) \tag{4}$$

- Kurtosis

### C. Histogram Features

Since each image has different dimensions and a different number of slices, we can use histograms to find the same length descriptor. By taking the histograms of an image, the number of pixels with a particular intensity can be found for each image. An unnormalized histogram can be denoted as:

$$h(r_k) = n_k \quad for \quad k = 0, 1, 2, ...L-1 \tag{5}$$

where nk is the number of pixels with intensity rk of an L-level image. The subdivisions of the intensity scale are called histogram bins. These histogram bins are taken as the descriptor features of an image. However, these unnormalized histograms need to be normalized since the components are estimates of the probabilities of intensity levels occuring in an image. The normalized histograms is denoted as:

$$p(r_k) = h(r_k)/MN \quad = n_k/MN \tag{6}$$

where M and N are the number of image rows and columns. In a normalized histogram the sum of p(rk) is always 1.

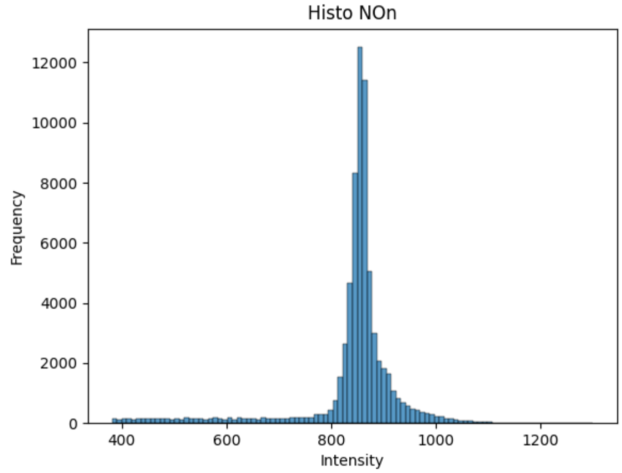


Fig. 2. The Unnormalized Histogram of an image

### D. Histograms Visualized

As seen in “Fig. 2”, the majority of the pixel intensities of this image fall in the range between 850 and 900. The zero pixel intensities are also excluded from the histogram to handle the background noise and artifacts of an image. The shape of this unnormalized histogram resembles a Gaussian ’bell’ curve where the Gaussian distribution shows the cluster of pixel intensities grouped together. In order to extract features from the histogram of an image, the histogram needs to be normalized to evenly distribute the probabilities of intensity levels occurring in an image.

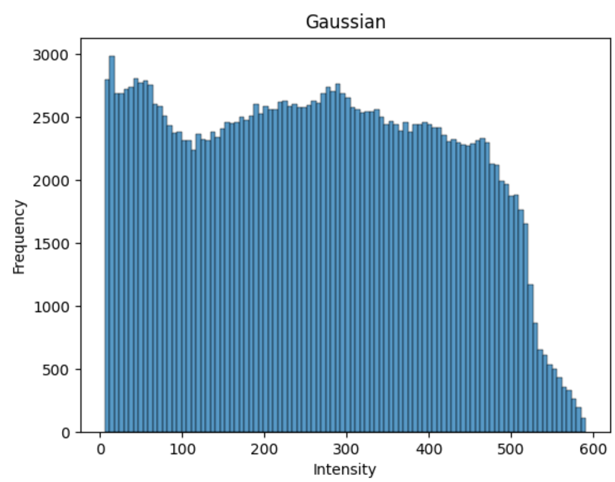


Fig. 3. The Normalized Histogram of an image

In the histogram visualization of “Fig. 3”, the Gaussian distribution of pixel intensities are now much more evenly spread throughout the histogram. This allows the histogram bins, holding the value of frequency in intensity scale to be taken as features describing the appearance of an image.

## IV. Model Evaluation

With each image having both gray-scale statistical and histogram pixel intensity features to distinguish the composition, different model classifiers can be evaluated to detect particular contraband objects. Each of the four models undergo an analysis with the image data to assess the accuracy rate, precision rate, and recall rate in determining the best. These models are first evaluated on the training data with 80% of the images for training and 20% of the images for testing. Once the best model is chosen, the model can be validated with the testing images in a blind evaluation.

### A. Choosing the Classifier

During the evaluation of Support Vector Machine (SVM), AdaBoost Classifier, Logistic Regression Classifier, and K-Nearest Neighbors(KNN), the cross validation accuracy score was found using five folds for a better measure of prediction accuracy. The accuracy for each fold was averaged along with the standard deviation.

```
Clf: Support Vector Machine
Accuracy Mean: 0.74
Standard Deviation: 0.01
Clf: Logistic Regression
Accuracy Mean: 0.72
Standard Deviation: 0.02
Clf: AdaBoost
Accuracy Mean: 0.64
Standard Deviation: 0.03
Clf: K-Neighbors
Accuracy Mean: 0.79
Standard Deviation: 0.02
```

Fig. 4. Average Cross Validation Accuracy and Standard Deviation for each model

As shown in "Fig. 4", the comparison of each accuracy mean and standard deviation results in KNN having the best accuracy. However, as it is known that accuracy alone cannot prove the efficiency of a model on a data-set, the classification reports of each model are also gathered, which entails the precision, recall, and f1-score.

The results in "Fig. 5" show that KNN has the best scores for recall. In this particular type of system to detect targets, the recall score has more importance over precision score. Recall, in this scenario, explains the percentage of targets detected out of the true number of targets detected while precision explains the percentage of non-targets detected out of the true number of non-targets. During an airport checkpoint screening, the system should prioritize the targets(1,2,3) being detected rather than the non-targets(0). As it is known, there is a trade-off between precision and recall where as the precision increases, the recall decreases. Although KNN has the best recall compared with the other three models, the recall score is still not the best. Only 59% of rubber targets(2) are being detected out of the true number of rubber targets. This means almost half the time, the system fails at detecting rubber in luggage. Because of the precision and recall trade-off, the model's parameters can be hyper tuned to support recall rather than precision. This will increase the detection rate of each of the targets.

Support Vector Machine Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.77 | 0.95 | 0.85 | 146 |
| 1.0 | 0.00 | 0.00 | 0.00 | 14 |
| 2.0 | 0.77 | 0.37 | 0.50 | 27 |
| 3.0 | 0.50 | 0.29 | 0.36 | 14 |
| accuracy | | | 0.76 | 201 |
| macro avg | 0.51 | 0.40 | 0.43 | 201 |
| weighted avg | 0.70 | 0.76 | 0.71 | 201 |

Logistic Regression Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.79 | 0.94 | 0.86 | 146 |
| 1.0 | 0.33 | 0.07 | 0.12 | 14 |
| 2.0 | 0.82 | 0.52 | 0.64 | 27 |
| 3.0 | 0.50 | 0.29 | 0.36 | 14 |
| accuracy | | | 0.78 | 201 |
| macro avg | 0.61 | 0.45 | 0.49 | 201 |
| weighted avg | 0.74 | 0.78 | 0.74 | 201 |

AdaBoost Classifier Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.74 | 0.73 | 0.74 | 146 |
| 1.0 | 0.11 | 0.14 | 0.12 | 14 |
| 2.0 | 0.50 | 0.15 | 0.23 | 27 |
| 3.0 | 0.31 | 0.64 | 0.42 | 14 |
| accuracy | | | 0.61 | 201 |
| macro avg | 0.41 | 0.42 | 0.38 | 201 |
| weighted avg | 0.63 | 0.61 | 0.60 | 201 |

K-Nearest Neighbors Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.88 | 0.89 | 0.89 | 146 |
| 1.0 | 0.71 | 0.86 | 0.77 | 14 |
| 2.0 | 0.84 | 0.59 | 0.70 | 27 |
| 3.0 | 0.56 | 0.71 | 0.63 | 14 |
| accuracy | | | 0.84 | 201 |
| macro avg | 0.75 | 0.76 | 0.75 | 201 |
| weighted avg | 0.84 | 0.84 | 0.84 | 201 |

Fig. 5. Classification Reports for each Model

### B. Hyper-tuning the Parameters

Hyper-parameter tuning is the process of searching for the ideal model architecture. To find the best parameters for KNN, a grid search cross validation was done with ten folds to determine the most optimal combination of n neighbors, the leaf size, and power parameter to use Manhattan distance or Euclidean distance. The best parameters were using 1 nearest neighbor, 1 leaf size, and the Manhattan distance. After the parameters for KNN were hyper-tuned to these specifications, the recall rates for each target efficiently improved.

The classification report in "Fig. 6", shows how the recall rates for rubber(target 2) increased from 59% to 74% and for clay(target 3) the rate increased from 71% to 81%. However, the saline(target 1) decreased from 86% to 79%. The system

```
K-Nearest Neighbor
Accuracy:  0.8756218905472637
Classification Report:
              precision    recall  f1-score   support

         0.0       0.92      0.91      0.91       146
         1.0       0.69      0.79      0.73        14
         2.0       0.83      0.74      0.78        27
         3.0       0.75      0.86      0.80        14

    accuracy                           0.88       201
   macro avg       0.80      0.82      0.81       201
weighted avg       0.88      0.88      0.88       201
```

Fig. 6. Classification Reports for K-Nearest Neighbors after Hyper-parameter tuning

seems to have more difficulty detecting saline objects in particular. Other methods may need to be done to address the detection of different saline objects.

### C. Analyzing Performance

The overall performance of the KNN on the training data can be further represented in a confusion matrix, visualizing the number of predicted targets versus the number of true targets.

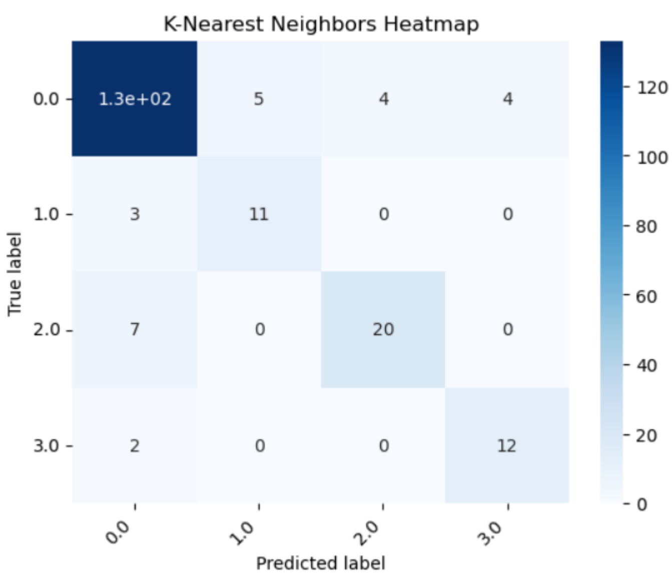


Fig. 7. Heatmap of the Confusion Matrix of KNN predictions

The heatmap shown in "Fig. 7" is a confusion matrix explains how each of the targets(1,2,3) being predicted either correctly or as non-targets. The diagonal represents the number of true target labels that were predicted correctly. This can be further seen as the target values can be grouped into one class to form a binary classification with targets being one class and non-targets being another class, as in "Fig. 8".

The final evaluation of the model on the training data shows the percent of false alarms in the system, the recall rate of the

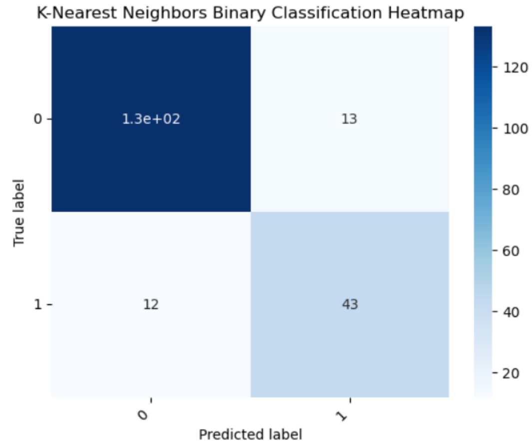


Fig. 8. Heatmap of the Binary Confusion Matrix of KNN Predictions

```
Accuracy:  0.8139303482587066
Percent False Alarm:  0.18606965174129353
Percent Total Detected:  0.6937145271118196
Percent Non-Targets Detected:  0.1353264015728631
Percent Saline Detected:  0.6363095238095238
Percent Rubber Detected:  0.6500925079448
Percent Clay Detected:  0.823992673992674
```

Fig. 9. Recall Rates of each target value

total targets detected, and the recall rate of each target detected, displayed in 'Fig. 9".

### D. Validation on Testing

Now that the KNN model has been completely evaluated on the training data, hold-out-testing is done on the testing data so validate how well the model performs on unseen data. The results shown below, display a total recall rate of 48.7% with the best detection of clay targets and the worst detection of saline targets.

Recall Rate:0.4878 60/123
Precision Rate:0.7936 273/344
Saline Recall Rate:0.34146 14/41
Rubber Recall Rate:0.54545 24/44
Clay Recall Rate:0.57895 22/38

## V. Conclusion

Using the K-Nearest Neighbor classifier detects object in the CT-scanned images with the best recall rate. This rate could be improved in future research to address the low saline recall rate. As this may be due to the different concentrations of saline, other measure will need to be done to increase the saline recall, in doing so increases the overall recall. To conclude, KNN performs the best to detect contraband in luggage.

## References

[1] Gu, Chen and Dong, Zhen. (2020). Automatic Target Recognition for CT-based Airport Screening System. Journal of Physics: Conference Series. 1453. 012111.10.1088/1742-6596/1453/1/012111.

[2] Wenling,Xue and Ting, Jiang, ”An adaptive algorithm for target recognition using Gaussian mixture models”,Measurement,Volume 124,2018,Pages 233-240,ISSN 0263-2241,https://doi.org/10.1016/j.measurement.2018.04.019.(https://www.sciencedirect.com/science/article/pii/S0263224118302951)

[3] C.Gonzalez, Rafael and E.Woods, Richard, ”Digital Image Processing,” Person. Hudson Street, New York, pp.189-191, 2018