An Analysis of Concealed Object Detection Using Decision Tree and Random Forest Algorithms

Pallavi Shintre[1], Shweatlana Mahapatra[1], Shweta Vincent[2*], Om Prakash Kumar[3]

[1]Manipal Institute of Technology, MAHE
[2]Manipal Institute of Technology, MAHE
[3]Manipal Institute of Technology, MAHE
**Corresponding author E-mail:shwetavincent@gmail.com*

Abstract

This paper presents a comparative study of the Decision tree algorithm and Random Forest algorithm, both using Haar wavelet transform to classify a concealed object as a threat or not. This finds its applications in airports and railway stations where passenger security is a major concern. A sub-millimeter wave image of a person having a concealed weapon on his thigh has been treated as the test dataset. The Haar wavelet transform along with the aforementioned algorithms is applied on the image to classify the patches in the images as threat or no threat regions. It is found that the Random Forest algorithm outperforms the Decision tree algorithm in terms of accuracy of detection as well as number of false positive generation.

*Keywords***:** Decision tree algorithm, Random forest algorithm, Haar wavelet transform, Millimeter waves

.

1. Introduction

Security at airports and railway stations has been on an all-time high for the past few years due to reports of malicious passengers carrying objects which could harm other passengers or the flight operations. These security procedures could be extremely long and tedious. In the interest of airport and railway safety, substantial amount of research has been carried out to devise methods to automatically detect objects such as weapons which are concealed under a passenger's clothes.

Lying between microwaves and infrared waves in the EM spectrum, millimeter waves have a frequency range of 30 GHz to 300 GHz. They exhibit a wavelength of 1 mm to 10mm. Millimeter waves find their applications in RADAR as they form a narrow beam width which can be used to detect objects in precise locations and at closer distances. Microwaves do not suit such an application where the objects which are close need to be detected. In order to reduce the beam width of microwaves, a larger antenna would be required, and this feature restricts the use of such a RADAR in compact systems. Thus millimeter waves are preferred over microwaves in the application of concealed object detection.

Submillimeter waves lie in the IR region of the EM spectrum and have a frequency range of 0.3 THz to 3 THz. These waves also find their application in detection of objects at short distances.

Active or passive wave scanners can be used to create millimeter or submillimeter waves. A passive type of scanner is used in [1] to detect threats and thefts. The advent of Machine Learning has revolutionized the manner in which this could be carried out. Submillimeter waves have been studied and used for detection of concealed objects in [2]. The proposed method in [2] claims to outperform the existing methods for concealed object detection. It was found that the use of tree sets for classification of the objects yielded an efficiency of 94%. Neural networks have also been explored in [3] for target identification and classification using millimeter waves. A fast threat detection algorithm has been proposed in [4] using millimeter waves. It has also been proven in [5] that fast algorithms for detection of multiple metallic objects can be implemented using millimeter waves.

This article focuses on the usage of an image of a person carrying a concealed object which needs to be detected. The decision tree algorithm is used for detection of the concealed object. Further this paper describes the usage of the Random Forest algorithm along with Haar filters for image classification which is able to detect the hidden object with a greater accuracy.

Section II of this article describes the theoretical background of the image processing techniques and machine learning algorithms used in this project. Section III of this article highlights the process carried out for the detection of the concealed object. Section IV of this article presents a detailed analysis of the results obtained and their discussion. Section V of this article concludes the paper and throws light on the future scope of work in this regard.

2. Theoretical Background

2.1. Haar Filters

Haar filters are used for image compression. The Haar wavelets rely on averaging and differencing values in an image matrix such that a sparse matrix with most entries being 0 is produced. The creation of a sparse matrix leads to lesser storage space and smaller file sizes [6]. MATLAB provides an in-built function for performing Haar compression.

2.2. Patches

A Patch is a possible threat region. It is a portion of the image that is considered for detecting threats. Patch formation is done to simplify the process of threat detection by considering smaller portions of the image for processing. Haar filters are implemented on these patches.

2.3. Decision Trees

Decision Trees are a type of Supervised Machine Learning where the data is continuously split according to a certain parameter. The tree can be explained by two entities, viz. decision nodes and leaves. The leaves are the decisions or the final outcomes and the decision nodes are where the data is split. Decision trees could be of two types namely, Classification Decision Trees which give a categorical output or Regression Decision Trees which give a continuous set of data. In our application for the detection of concealed objects, the Classification Decision Trees have been used.

2.4. Random Forests

The Random Forest algorithm is an extension of the Decision tree algorithm. It consists of having a number of decision trees to decide upon which is the best option for the classification of a particular set of data. The denser the forest, the better is the accuracy of the result. Random forests can also be an ensemble of learning or classification algorithms for a particular set of data.

2.5. Classifiers

Classifiers are used to calculate the positivity or the negativity of a term in a set with respect to the other terms in the set. In this case, we use classifiers to find the positivity of each of the terms in the positive patch dataset with respect to all the negative patches. This provides the most positive patch, therefore giving and outcome of a single best value patch. This method generally gives the final accuracy level which can be tabulated and can be used to conclude which algorithm is the best one.
The upcoming section of this article presents the experimentation carried out to classify the pixels of a test image as a threat region (for a concealed object) and the rest of the regions as a no threat region.

3. Concealed Object Detection using Decision Trees and Random Forests

For the application of the Decision tree and Random forest algorithms, the test data set was an image from [2]. This consisted of a person with a concealed object tied to his thigh. The test data image used in our article is shown in Figure 1.

Fig 1: Image used for testing: Person with concealed object on his thigh (Adapted from [2])

Initially a 5 x 5 median filter is applied to the image shown in Figure 1 for the purpose of denoising the image. After enhancement of the image, a learning model was used for naming each pixel 0 or 1 based on whether it has a pixel value of threat or no threat. After enhancement, feature extraction needs to be performed in order to make the image data compatible for the learning model. This is done using a technique called patch extraction. Patch is an n*n matrix centered at the (n/2, n/2) th pixel. The image is divided into patches centered at pixels, 2*2 pixels apart. Patch sizes of 9*9, 19*19 and 39*39 are extracted on each pixel. These pixels that turn out to be the centers of patches are called active pixels. The Haar wavelet transform is used to calculate the threat value at each pixel using equation (1).

$$Threat\ value = (Pixel\ value * Haar value) - (Pixel\ value * Haar value') \tag{1}$$

The outcome of this is a matrix with of threat values for each patch considered in the image. This forms the patch dataset T. If the value of the threat is positive, then the pixel becomes a part of the positive patch set T_p or else it becomes a part of the negative patch set T_n. The number of pixels in the final datasets of T_p and T_n are annotated as nn and np.

After the creation of the positive and negative threat patch data sets, the next step of the process is to fine tune the results by considering the patch dataset of T_n and comparing each pixel value with a threshold value. If the pixel value is lesser than the threshold, then the pixel remains in the set T_n or else it gets moved to the set T_p.

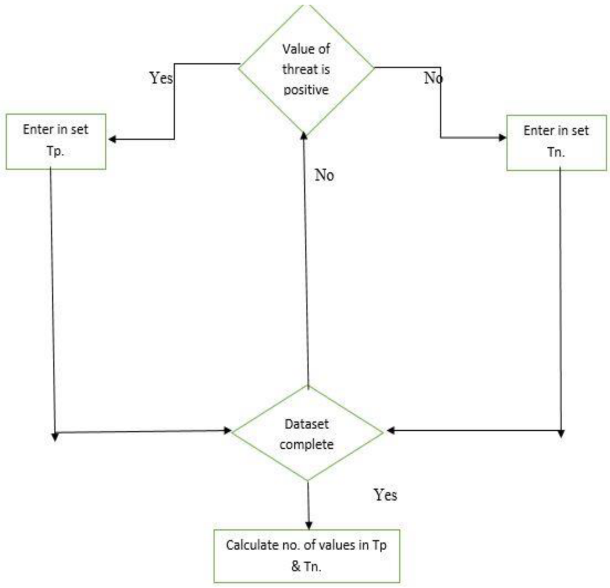

Fig 2: Flowchart depicting the creation of the positive and negative patch datasets.

4. Results and Discussion

This section of the article presents the results of the application of the Decision tree algorithm along with Haar wavelets and the Random Forest algorithm with the Haar wavelets.

Figure 3 depicts the positive and negative patch data sets detected using the Decision Tree algorithm after the Haar wavelets are applied to the dataset.

Fig 3: Patch detection using Decision tree algorithm

Fig 5: Patch detection using Random Forest algorithm

Figure 4 illustrates one of the decision trees generated by MATLAB software for classifying the image pixels in the image as positive or negative threat patches.

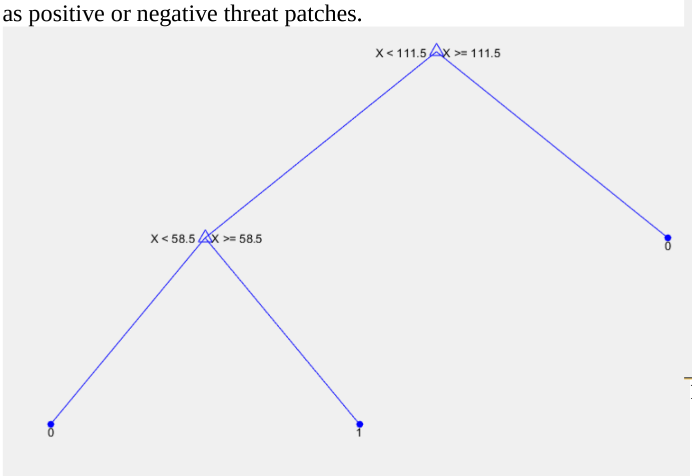

Fig 4: Sample Decision tree generated for patch classification

It can be observed from Figure 3 that, the Decision tree algorithm gives extremely coarse boundaries along the concealed object on the thigh of the person. This is a major drawback of the algorithm and hence renders it of less usage in case of objects which might be very sharp and small in size.

Therefore the need arose to test the application of the Random Forest algorithm on the same test dataset. Figure 5 illustrates the image of the patch classification obtained by applying the Random forest function of the MATLAB software on the test image.

As it can be easily noticed that the Random Forest algorithm provides much more intricate details of the patch classification performed. The boundaries of the classified pixels are finer in comparison to Figure 3 which depicts the Decision Tree classifier. This occurs due to the finer model of classification in Random forest algorithm by using an ensemble of Decision trees to classify the respective patches.

Figure 6 illustrates one of the 100 decision trees of the Random forest algorithm which was generated for patch detection using the MATLAB software.

Table 1 tabulates the percentage of false positives and false negatives generated by both the algorithms over the same dataset.

Table 2 depicts how the values of Table 1 have been obtained.

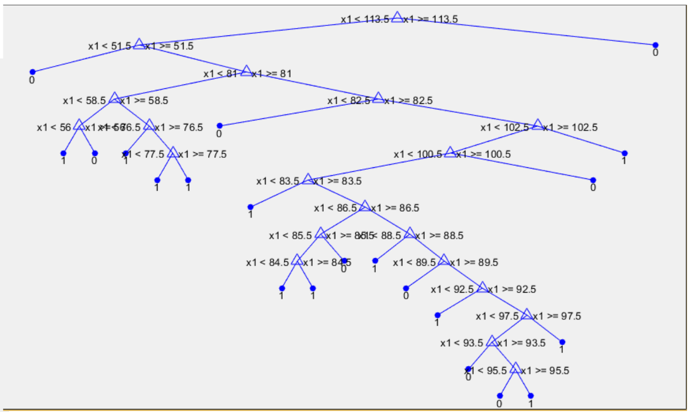

Fig 6: Sample Random forest decision tree generated for patch classification

Table 1: False positive and False negative percentage for both algorithms

Algorithm	False positives	False negatives
Decision tree	17.8%	4%
Random forest	18.8%	0%

Table 2: Details of numbers of False positives and False negatives in both algorithms

Parameter	Value
Total number of true positives	1921
Total number of true negatives	150839
Positives in the positive area of Decision tree	1260
Positives in the positive are of Random forest	1021
Negatives in the positive area of Decision tree	0
Negatives in the positive area of Random forest	90
Positives in the negative area of the Decision tree	27761
Positives in the negative area of Random Forest	26993

5. Conclusion and Future Work

This paper presented a simulation study of the superiority of the Random forest algorithm over the Decision tree algorithm in detection of concealed objects.

MATLAB was used as a platform for testing the claim that the Random Forest algorithm along with Haar wavelet transform gives a better outcome than a Decision tree along with the Haar transform. The methodology used here was, applying either one of the algorithms on each pixel and giving each pixel value a tag of 0 or 1, based on whether the algorithm states that the pixel value is within the threat range or not. Then patches were extracted out of the image and stored in a database. Haar filters were run on these patches and each patch was given a value. Thresholding was done on the Haar values of all the patches from the set and thus threat patches are separated in a positive dataset and a negative dataset. Each patch from the positive dataset is then completely made white or 1 and the remaining patches are made 0 or black. Thus, the output image is a black and white image with probable threats being white.

It was encountered that MATLAB took longer running time as it is not a dedicated machine learning or image processing software. Therefore, a platform such as Open CV would be more suitable for the testing of this application.

It was also found that using RGB images for the learning algorithms was a preferable choice as it gives more attributes for the algorithm to make decisions.

References

[1] N. E. Alexander, C. C. Andres, R. Gonzalo, "Multispectral mm-wave imaging: Materials and Images", SPIE, vol. 6948, pp 694803-694812, 2008.

[2] S. L. Tapia, R. Molena, L. P. Blanca, "Using machine learning to detect and localize concealed objects in passive sub millimeter wave images", Engineering Applications of Artificial Intelligence, Elsevier, vol 67, pp 81-90, 2018.

[3] S. Agarwal, A. S. Bisht, D. Singh, N. P. Pathak, "A novel neural network based image reconstruction model with scale and rotation invariance for target identification and classification for active millimeter wave imaging", Journal of Infrared Millimeter TeraHertz waves, vol 35, pp 1045-1067, 2012.

[4] I. Gomez, L. P. Blanca, R. Molena, A. Katsaggalos, "Fast millimeter wave threat detection algorithm", 23rd European Signal Processing Conference, pp 599-603.

[5] C. D. Haworth, Y. De Saint-Pern, E. Trucco, Y. R. Pettillot, "Detection and tracking of multiple metallic objects in millimeter wave images", International Journal of Computer Vision, vol 71(2), pp 183-196, 2007.

J. Shore, "On the Application of Haar Functions", IEEE Transactions on Communications, vol 21(3), pp 209-213, 1973.