CNN based Passenger screening algorithm on TSA scan images data

Prachi A. Patki and Amruta Deshmukh
Dept. of Computer Science and Electrical Engineering,
University of Maryland, Baltimore County,
patki1@umbc.edu,amu1@umbc.edu

I. Abstract

Full body scan is a normal security procedure employed at the airports throughout the world. These body scanners use millimeter wave technology to get a 360° scan of the passengers. The scans can penetrate through clothes and can reveal objects attached to the body.[1] However, detecting objects that are attached to the body is a manual process and consumes time and decreases the competence. Due to this limitation, security queues at the airports usually end up being unnecessarily long and time consuming. Our project is an attempt to automate this process using machine learning and image processing techniques to reduce the time it takes to go through such security procedures. It will not only reduce the hours wasted in security queues but will also reduce the manual efforts involved in the process and up-hold the privacy of all those going through the scanners.

II. Introduction

The increase in the demand of air travels have increased drastically. In 2017, the global air traffic passenger demand has increased by over 7.5% compared to the earlier year and is said to increase by another 6% in the following year. With the increase of travelers, security forms a prime concern. The TSA (Transportation Security Administration) department of Homeland Security of USA works for providing security measures by detecting individuals in possession of threatening item. The state of the art surveillance system developed for airport security to detect and identify threatening objects, which are concealed on the human body has a huge advantage compared to conventional metal detectors which fail to detect nonmetallic objects such as plastic explosives and plastic guns. The TSA has challenged to provide Algorithm with improved accuracy in predicting the threats via scanned image which forms the base of our project. An example of full body scanned image with a threat present is shown in Figure 1. Our project involves accurately predicting the threat zone present in a fully scanned body image. Threat here is any abnormal object present on the clothes or hidden inside the clothes. With the help of this project, the wait times will be significantly decreased, improving the passenger experience.

III. Related Work

The applications of CNN are varied and stretch along different fields. It is used prominently in the clinical applications which server the purpose of understanding the causes of diseases and find a remedy over the same from various kinds of images (MRI, DiCom). We studied about the classification of CT brain images which contained data of patients with Alzheimers disease or Lesion or normal ageing. The reason behind choosing CNN on medical images as related work is the similarity of the medical image data and our data. Where the base object remains constant throughout, and the differentiating factor is an object or pattern with size very less as compared to base image. The research acquainted us with the fact that when CNNs are trained with a proper regularizer, they can achieve high performance on visual recognition tasks without depending on the handcrafted features. The work addressed in the [2] helped us understanding the version of CNN used for scanned images and thereby provided a base for implementing the project.

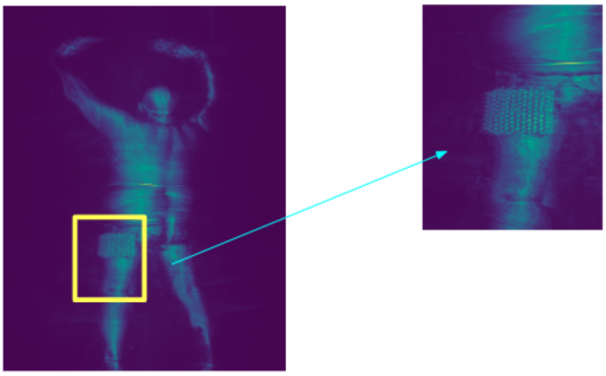


Figure 1. Image in the left shows an full body image form a certain angle of the total 3D image with a threat present in it. Image on the right shows a zoomed image of just the zone for a certain angle in which the threat is present.

IV. Method

While it is very easy for humans to identify places, things, environments, Computer face difficulty for the same task. But with the help of machine learning and Computer vision technologies, image recognition is possible. Image recognition implies resembling the working of human brain, thus is complex. For instance, to devise self-driving cars, the overhead is to differentiate between various instances of videos and photos. Solution for this is neural network which sacrifices ability to generalize for a feasible output. The convolutional network is preferred over conventional network as the computational overhead of conventional network makes the network less accurate. Filtering in convolutional networks make image processing computations manageable. The concept of convolutional neural network was best suited for the project as the project involved looking for low level features like curves and edges. These, features distinguished the abnormal object on/inside the clothes of passengers entering the security checkpoint.

V. **Implementation**

A. Dataset

The data set contains body scan images produced by scanning devices and is acquired via Kaggle. The total number of images present in the dataset are 1247 (each image has 16 angles and can be divide into 17 zones) including training and testing data. Each aps image has a dimension 512x600x16. Data set is available in three different formats. These formats are as follows

- .a3Daps = combined image angle sequence file (41.2MB per file)
- .aps = projected image angle sequence file (10.3MB per file)
- .a3d = combined image 3D file (330MB per file)

We decided to go ahead with .aps files ((10.3MB per file) which is feasible for computation. Each aps file has 16 images equally spaced at angle 22.6^o around the axis covering 360^o around the person. The result of this is an image file that, when played back plane-by-plane, appears like the object is spinning on the screen. The Figure in **??** is the representation on single aps file

Along with this we had zone wise threat labels in the data set.Each image had 17 labels representing the presence of any object in the corresponding body zone.The dimensions defining 17 zones are provided by TSA.Below is the snippet from csv file of labels for one aps image.

B. Data-processing

In order to enhance image features and remove noise. Several image processing techniques were applied on the data

1. RGB to Grey Conversion:
Originally the images in the dataset were in the form of RGB images. RGB to Greyscale conversion and contrast adjustments are done to extract features obtained by luminance difference

2. Thresholding:
To enhance the human figure out of the images we applied constant threshold values on numpy arrays of images.A threshold for pixel is kept 50 as all the pixel values less than 50 are nothing but background noise.

2. Image Augmentation:
Since we had limited number of images in dataset(1247) in total. We used image augmentation techniques by shifting an image horizontally.

C. Approach

1.Feeding the data into CNN :
We had to consider to major factors while feeding the data into CNN, zones and the angles

- Zones: First approach was to give full images as an input to CNN. As the threat object is very trickily hidden under the clothes of a person. There was a minor difference between safe image and image with threat when looking at images. The model was unable to tell the difference looking at the full images. Therefore we decided to go for the option to feed the model with zone wise segments of images. After receiving an input image , it is segmented into 17 pieces as shown in Figure 3. Following is the representation of one of the segments.
- Angles: Initially, all the 16 angles were considered shown in fig 10 were considered for each model. As this was taking long time for computation we decided to consider three angles (i.e. angle taking front view,back view and side view for each zone). The thought behind doing this is, these three angles are primary contributors in detecting any threat present in that zone[3].
- Data Arrangement: Next task was to find a way to coordinate angles and zones As we didnt have the labels zone wise separated in the given labels. We separated labels zone wise and divided a single CSV labels file into 17 different CSV files.To get the zone information, we first created a set of dimensions to crop the full image into 17 segments. After that we wrote separate python program which give output as multidimensional array that contains all the file ids are present in the dataset and each id points to 17 arrays representing 17 zones. Each zone is further pointing to its Numpy array and its respective labels. The dataset arrangement flow is as shown in Figure. 6.

D. Model Architecture

- Transfer learning Approach with Fine tuning: As we had limited number of training data, we decide to use a pretrained imageNet Model with Vgg16 Architecture. In this model the initial layer were frozen to extract basic features (edges, shapes) and last two layers were fine-tuned according to the actual task. Unfortunately, we were not able to achieve good accuracy and sensitivity with this approach because of lack of knowledge about constraints from previous models that are to be taken care of.
- Randomized Weights:The CNN model implemented is based on the examples provide by TFLearn on CIFAR10 dataset[4]. In this model the zone wise segments of a full image are given as an input to the first layer of convolution followed by max-pool layer. The max pool layer output is again passed through convolutional layers twice and then second max pool layer. This max pooled output is provided to the fully connected layer which actually works as a binary classifier to give final output as 'threat detected' or 'safe'. The classifier is using TFLearn which is wrapper around Google's Tensor flow library with a simpler API. The learning rate for the model is 10^{-3}.The model architecture is as shown in Fig 7.

E. Model Details

- Input Layer: The first layer involves inputting data the function 'input data' takes the arguments as shape of input

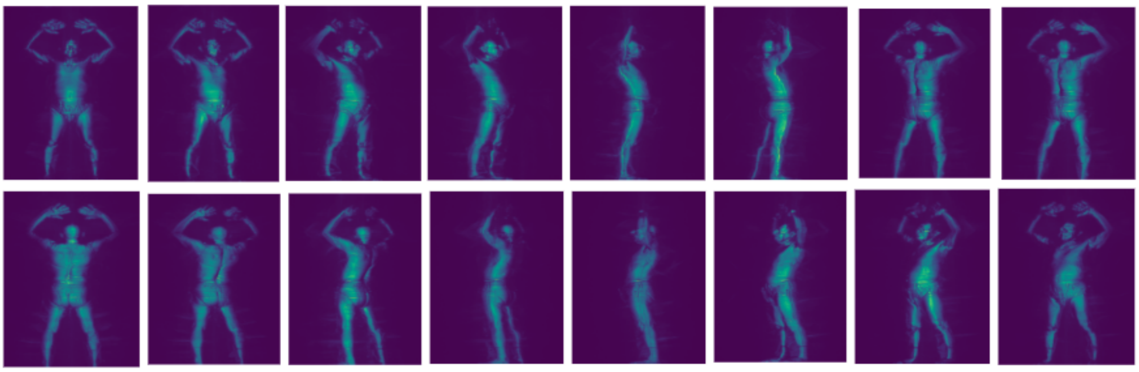


Figure 2. Representation an single .aps file for images from all the angles each separated by 22.5 degrees

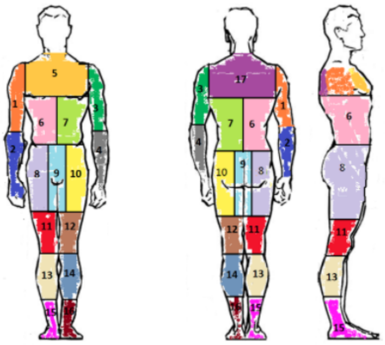
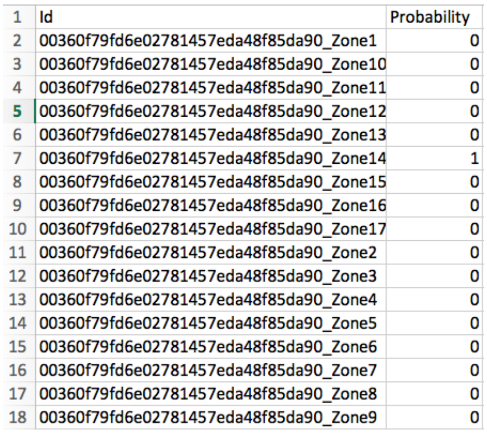


Figure 3. Zone wise threat labels [3]



1	Id	Probability
2	00360f79fd6e02781457eda48f85da90_Zone1	0
3	00360f79fd6e02781457eda48f85da90_Zone10	0
4	00360f79fd6e02781457eda48f85da90_Zone11	0
5	00360f79fd6e02781457eda48f85da90_Zone12	0
6	00360f79fd6e02781457eda48f85da90_Zone13	0
7	00360f79fd6e02781457eda48f85da90_Zone14	1
8	00360f79fd6e02781457eda48f85da90_Zone15	0
9	00360f79fd6e02781457eda48f85da90_Zone16	0
10	00360f79fd6e02781457eda48f85da90_Zone17	0
11	00360f79fd6e02781457eda48f85da90_Zone2	0
12	00360f79fd6e02781457eda48f85da90_Zone3	0
13	00360f79fd6e02781457eda48f85da90_Zone4	0
14	00360f79fd6e02781457eda48f85da90_Zone5	0
15	00360f79fd6e02781457eda48f85da90_Zone6	0
16	00360f79fd6e02781457eda48f85da90_Zone7	0
17	00360f79fd6e02781457eda48f85da90_Zone8	0
18	00360f79fd6e02781457eda48f85da90_Zone9	0

Figure 4. Dataset labels

tuple and other two inputs as output of data processing and data augmentation sub classes. Data processing subclass takes care of normalizing and centering of data, Data augmentation takes care of real-time data augmentation and performs operations as flip,rotation and blurring as shown in Fig. 8.

- Convolution and Max-pooling Layer: The first convolution layer takes the input of incoming tensor from the input layer, the number of convolution filters ans size of the filter. We have used 32 filters with size 3 each as shown in fig **??**. The maxpooled output is again passed through two layers of convolution followed by second layer of maxpooling. The number of filters in the second and third layer is kept double the size of number of filters in the first layers Convolutional neural networks use hierarchical features in their processing. The features in lower layers are primitive while those in upper layers are high-level abstract features made from combinations of lower-level features. Therefore, it is reasonable that the deeper layers should try to look into as much details as possible of the image data set.
- Fully connected layer: Number of units provide as an input argument to fully connected is ’128’. After the first fully connected layer, we have used dropout by factor of 0.5 to prevent over fitting of data. The final layer works as binary classifier which gives an output ’1’ for an image with threat and ’0’ for safe image
- Activation: In the initial layers network uses ReLu acti-

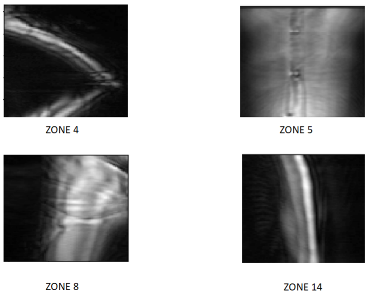
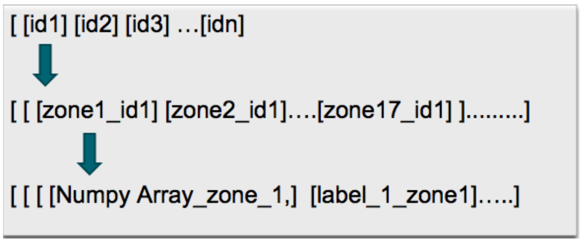


Figure 5. Segment for zone 14



[[id1] [id2] [id3] …[idn]

[[[zone1_id1] [zone2_id1]….[zone17_id1]]………]

[[[[Numpy Array_zone_1,] [label_1_zone1]…..]

Figure 6. Zone wise Data Arrangement

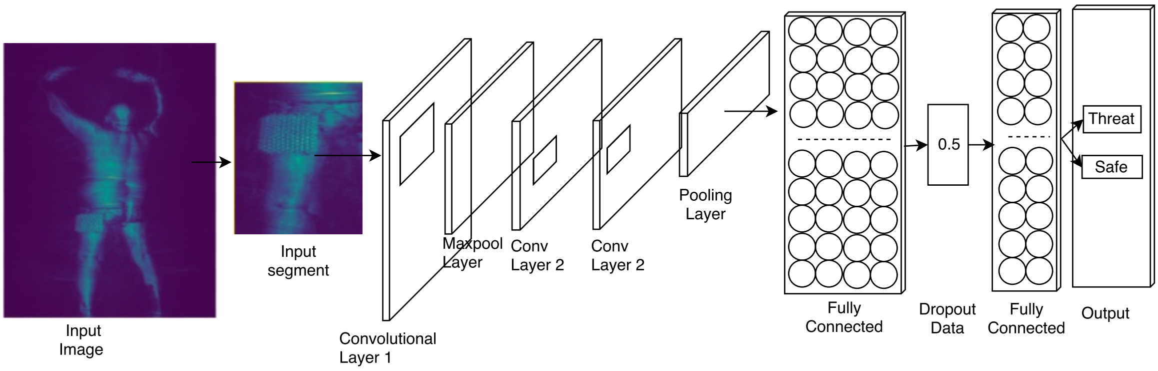
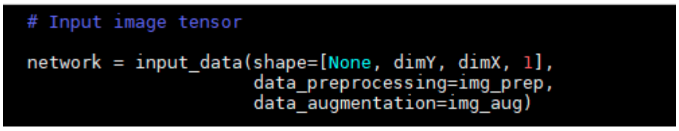


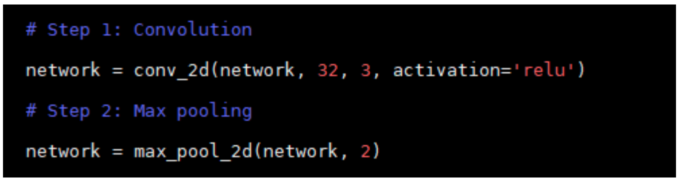
Figure 7. Deep CNN model architecture of our model to detect threats in the Image.



```
# Input image tensor

network = input_data(shape=[None, dimY, dimX, 1],
                     data_preprocessing=img_prep,
                     data_augmentation=img_aug)
```

Figure 8. Input Tensor



```
# Step 1: Convolution

network = conv_2d(network, 32, 3, activation='relu')

# Step 2: Max pooling

network = max_pool_2d(network, 2)
```

Figure 9. Convolution and Max-pool Layer

vation. After studying about these activation in detail we realized that ReLu in the initial layers will make less dependent between features. In the output layer we have used Softmax activation to give a probabilities of two occurring classes.

VI. **Results**

A. *Accuracy of the model*

The train and test data ratio in the dataset is 80:20. The accuracy is plotted according to zones. At the present we have not evaluated all the 17 zones. We are considering eight major zones where threat probability is more that the other zones. Also we are considering three primary angles for each zone which are front, back and respective side of the zone. The plots below show accuracy respective to the zones. The accuracy is calculated by considering true positives and true negatives predictions on test images. The model most accurately predicted the threat present in the zone 7 which was area around the abdominal region. Whereas we observed decrease in accuracy in the zone 8 and 10 in the areas around hips. After looking at the training data we realized that because of the presence of pockets and other cloth lines present on the bottom-wear of the subjects, it is ambiguous to tell if it is cloth pattern or threat object. The average accuracy is 87.20%

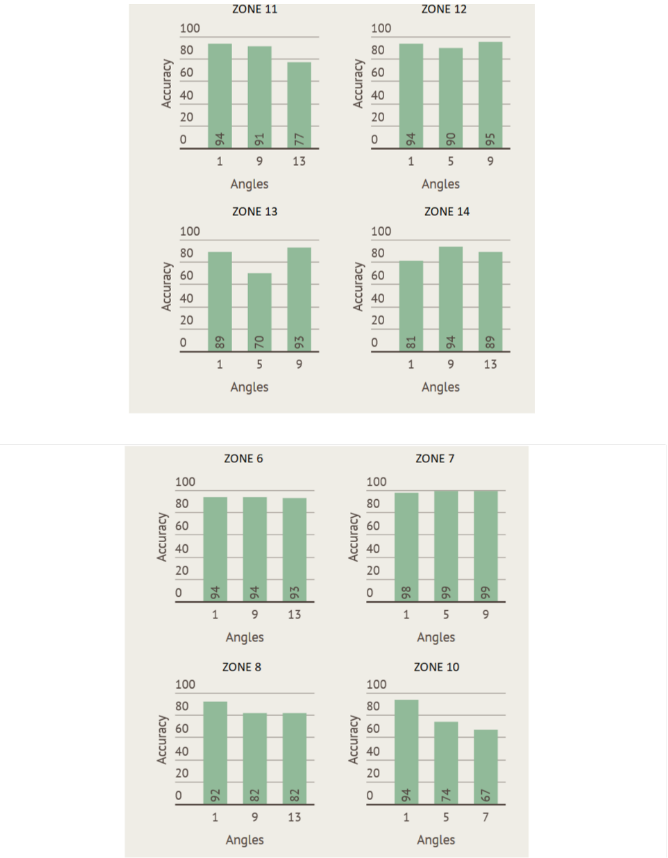


Figure 10. Graph showing Zone wise accuracies for the implemented model

B. *No. of epochs vs. Accuracy*

We have trained the model for different number of epochs in the range 1 to 500. the model acquired the best accuracy at 100 epochs. The figure 11 shows the changes in accuracy w.r.t. increase in the number of epochs

VII. **Future Work**

The first task that we are planning to implement as a part of future work is building models to detect the threats in all the 17 zones which is implemented only for 8 zones currently. The integration of these 17 models using a con-

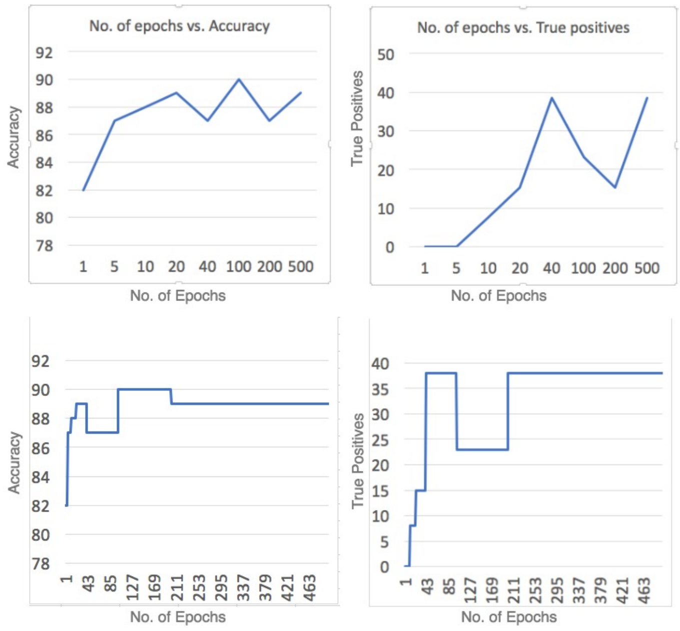


Figure 11. Graph showing the tren in which the accuracies change with increase in the numher of epochs on the left. A graph showing the percentage of true positives with increse in the number of epochs.

cept 'Multiple Instance Learning'. Where the labels will be assigned to bag of instances. A bag is labeled positive if at least one instance in that bag is positive, and the bag is labeled negative if all the instances in it are negative. After covering all the zones the model should be exposed to all the 16 angles which are provided as a part of single .aps file. Presently, we are considering three angles (i.e. angle taking front view, back view and side view for each zone) for each zone. We started with 'Transfer Learning Approach' to build this model. But later shifted to Randomized weights approach. Implementing the model with 'Transfer learning approach with fine tuning will be a good step to check the improvement in accuracy and sensitivity.We will explore 3C convolution to see if it is a good option for the model[3].

VIII. **Conclusion**

This report presented an implementation of Machine Learning model which takes an input as segments(zones) of scanned images and predicts if a zone has a threat present in it. Final prediction is a binary classification where '0' stands for safe images and '1' stands for an image with a threat. We have achieved an average accuracy of 87.2%.This accuracy is achieved on training the model with 100 epochs. This model can be taken as good starting point to developed an deep CNN integrated with all the zones and angles to achieve better Accuracy and Sensitivity which will make it of potential application in TSA scanning systems.

References

[1] M. E. Boris Kapilevich, "Detecting hidden objects on human body using active millimeter wave sensor," 2010.

[2] R. H. Xiaohong W. Gao, "A deep learning based approach to classification of ct brain images," 2016.

[3] E. L.-M. Subhransu Maji, Evangelos Kalogerakis, "Multi-view convolutional neural networks for 3d shape recognition."

[4] "Dataset description provided by tsa." [Online]. Available: https://www.kaggle.com/c/passenger-screening-algorithm-challenge/data