Threat Detection in TSA Scans using AlexNet

Amartya Bhattacharyya
Department of Mechanical Engineering
University of California, San Diego
San Diego, California, USA
ambhatta@eng.ucsd.edu

Christine H. Lind
Department of Electrical and Computer Engineering
University of California, San Diego
San Diego, California, USA
clind@ucsd.edu

Rahul Shirpurkar
Deptartment of Mechanical Engineering
University of California, San Diego
San Diego, California, USA
rshirpur@ucsd.edu

***Abstract*—To help address passengers' concerns of both privacy and safety, it has been proposed to free TSA (Transportation Security Administration) scanning and analysis of human intervention by making use of machine learning algorithms. In this project, AlexNet, a convolutional neural network is implemented on a set of preprocessed images extracted from full body TSA scans. The project is, for the time being, limited to 2 regions of the body - the upper chest and the upper back. The log losses obtained were 0.1221 and 0.0088 respectively for the 2 regions, suggesting the algorithm is accurate.**

I. Introduction

After the infamous bomber in 2009, who exploited the flaw in the airport security measures. At the time TSA only used Metal detectors to screen travelers for concealed weapons. After the event two new types of scanners were introduced:

1) Millimeter-wave scanner
2) Backscatter - X-ray scanner

Of which the former uses radio waves to search for hidden weapons or devices.These are the full-body scanners travelers encounter at U.S. airports. The ones people are made to stand in with their feet apart and hands above their heads. Experts agree they shouldn't worry anyone.

For the past 8 years various court sessions have been held opposing the use of body scanners. Two of the main reasons brought up were health and privacy. Health is immediately ruled out as the millimeter wave does not harm the body with extended exposure.The privacy concern was dealt with by making it the duty of the officer behind the screen to voluntarily delete the images. In spite of these measures stats show people prefer pat-downs over the scanners. Another drawback of the present system is the over straining of the TSA officers. Due to the drop on the total number of inspectors from a range of 46 thousand in 2006 to 42 thousand in 2016(LA Times). Officers are forced to perform duties of over 12 hours. This is in spite of the fact that total no of travelers have increased. The total checkpoint volume has increased from 708 Million in 2006 to 740 million in 2016.

Our aim was to replace the officer behind the screen with a program to solve the above mentioned issues. Eliminating privacy concerns and improving on the human error.

II. Related Work

The most successful proposed method for threat detection suggests dividing the body into regions and then searching these specific regions for foreign objects (Guimares and Tofighi, 2018). The paper cited uses a convolutional neural network to divide the body into specific regions for threat detection. Jaccard et al. (2016) use a convolutional neural network to successfully detect small threats in X-ray images of cargo. Krizhevsky et. al (2012) used the convolutional neural network AlexNet to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2012 contest into 1000 different classes. They achieved a winning top-5 test error rate of 15.3%, compared to the 26.2% error rate achieved by the runner up. Because of this we chose to divide the body into regions and use AlexNet as our convolutional neural network.

III. Dataset and Features

Original Dataset

The dataset was obtained from the Passenger Screening Algorithm Challenge on Kaggle (source). The original dataset consisted of four types of files: calibrated object raw data, projected image angle sequence (.aps), combined image 3D, and combined image angle sequence. As described in the Kaggle challenge and displayed in Figure **??**, the body is divided into 17 different regions for threat detection. Each file in the dataset is labeled with a binary encoding specifying the presence of a threat in each region. The threat distribution of the full dataset is shown in Figure **??**.

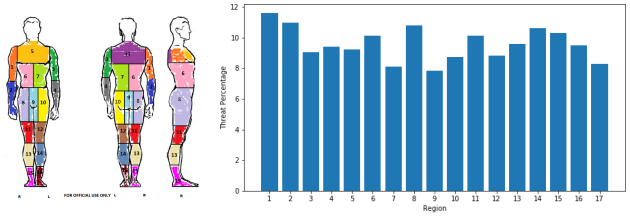

Fig. 1: Regions used for threat detection (left) and distribution of threats in the original dataset(right).

Custom Datasets

As people reported achieving the best results using only projected image angle sequence files, and as these files were significantly smaller than the other types of image files, only these files were used. Each .aps file consists of 16 two

dimensional frames equally spaced in angle as shown in Figure **??**. Since the frames are not labeled separately, it is not necessarily known what frame or frames a threat will be visible in. However, as the entirety of regions 5 and 17 (Figure **??**) are visible in the first and ninth frames (Figure **??**) respectively, it is known that threats in these regions will be visible in these frames. We thus chose to focus on these regions using only the corresponding frames. This was done by constructing a balanced dataset of corresponding frames for each region. Due to the threat distribution in the original dataset (Figure **??**), this resulted in two datasets, one of 212 images total for Region 5 and one of 190 images total for Region 17.

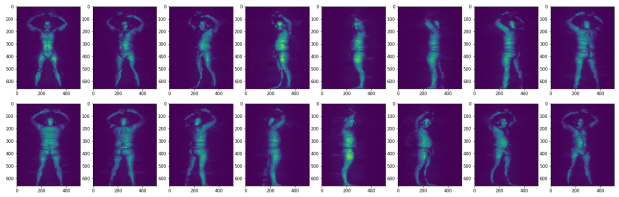

Fig. 2: The 16 frames of a sample .aps file. Threats in Region 5 will be visible in Frame 1 (top leftmost image). Threats in Region 17 will be visible in Frame 9 (bottom leftmost image).

Data Augmentation

All Images: In order to improve the classification results, all images went through an image preprocessing pipeline as displayed in Figure **??**. The final process consists of the following steps:

1) Grayscale Conversion. This was done so histogram equalization could be performed.
2) Local Histogram Equalization. This was done to enhance the contrast of the image.
3) Crop. This was done to exclude regions of non-interest and prevent other possible threats from interfering.
4) Smoothing. Performed on input for region 5 only. Due to height differences, faces were often included in images for region 5. It was found that the edges of the face would interfere with threat detection, so these images were smoothed with a gaussian blur.
5) Zero-centering.
6) Normalization.

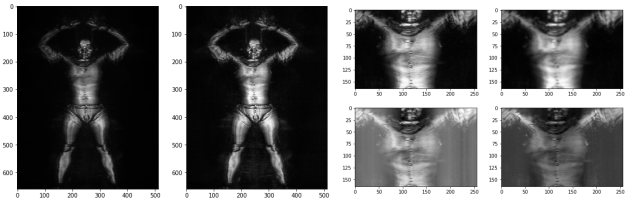

Fig. 3: The steps of the image preprocessing pipeline (from left to right) for a sample image: grayscale conversion, local histogram equalization, crop, smoothing, zero-centering and normalization. Smoothing was only used on Region 5.

Several image filters were experimented with in steps four and as step seven of the image preprocessing pipeline in an attempt to further optimize threat detection. As threats were manually seen to contain sharp edges, several edge enhancement filters were tested in step 4. These include unsharpened mask, laplacian, sobel and canny edge detection. Since principal component analysis (PCA) and whitening filters are often found to improve results of convolutional neural networks, these were tested as a potential step seven of the image preprocessing pipeline. The tested filters are displayed in Figure **??**. None of the filters mentioned were found to improve our results.

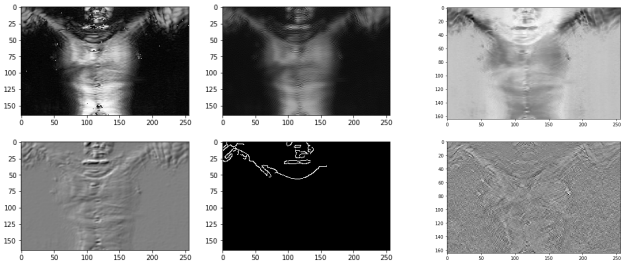

Fig. 4: Tested filters on a sample image. The image enhancement filters (displayed in the first two columns) are from from left to right: unsharpened mask, laplacian, sobel and canny edge detection. The last column displays PCA (top) and whitened image (bottom). None were found to improve our results.

Training Images: The images were split into separate training and validation datasets with a split ratio of 0.8. This resulted in 170/42 and 152/38 training/test images for Region 5 and Region 17 respectively. The training images were then further flipped randomly in the horizontal direction and/or randomly translated between -30 to 30 pixels in both the horizontal and vertical directions.

IV. Methods

AlexNet Structure

AlexNet is a convolutional neural network that consists of 15 layers total. These include five convolutional layers, each with a rectified linear (ReLu) activation function, two normalization layers, three pooling layers, two dropout layers and three fully connected layers. The first two fully connected layers use a ReLu activation function, while the last fully connected layer uses a softmax activation function. The detailed architecture of AlexNet is displayed in Figure **??**. Each type of layer functions as follows:

The convolutional layer: The convolutional layer computes the output of neurons connected to a local region in the input. The size of this local region is specified by a given filter size F, and one convolutional layer often has more than one and up to N filters.

Each neuron computes a dot product between a weight and a local region, the number of weights is calculated by element wise multiplication of the size of the filter (F, F) and the

depth of the input image. In order to reduce the number of parameters of the network, one assumes that the neurons of each depth slice share the same weight and bias. This reduces the number of parameters of the network and the forward pass of the convolutional layer is now computed as a convolution of the weight with the input layer.

The activation function used by the convolutional layer is the ReLu function, $f(x) = max(0, x)$, as this function trains a four layered neural network up to six times faster than an equivalent network that uses the traditional hyperbolic tangent function (Krizhevsky et. al, 2012).

The Normalization Layer: The normalization layers are inserted between convolutional layers and pooling layers and is used to generalize the network and reduce the overall error

The Pooling Layer: The pooling layer is applied after the convolutional layer, and its purpose is to downsample the spatial size of the network. This further reduces the number of parameters and computations done by the network, and as such it controls overfitting. The pooling method used by AlexNet is max pooling, which takes the largest output returned by a region of neurons in the previous layer.

The Dropout Layer: The dropout layers are inserted between the fully connected layers and discards a specified ratio of neurons, which again reduces the number of parameters and thus prevents overfitting.

The Fully Connected Layer: Neurons in the fully connected layer connect to all neurons in the previous layer, and as such this layer behaves like a regular neural network or a multi-layer perceptron. It can also be considered a convolutional layer with filter size equal to one, and the activation functions for this layer are thus, with the exception of the last fully connected layer, the same as for the convolutional layer. Since the output of the final layer returns a probability the softmax activation function, also known as the normalized exponential function, is used for the last layer.

AlexNet Implementation

An original AlexNet was implemented in Python following the theoretical structure outlined in the previous section and represented in Figure **??**. However, a pretrained AlexNet already trained on 1.2 million high-resolution images of the ImageNet challenge, as described by Krizhevsky et. al (2012), also exists in MATLAB. Since the first layers of a convolutional network detect low level details of images, such as edges, it is advantageous to use already pretrained layers to both speed up and improve the training process. Therefore, AlexNet was also implemented in MATLAB using layer transfer.

In MATLAB, the activation functions are separated into separate layers. In addition, the first and last MATLAB layers are just input and output layers. This results in a network with 25 layers total. In order for AlexNet to classify our threat-detection images into the binary threat or no threat classes, the last fully connected layer (which corresponds to the last three layers in MATLAB) needed to be modified and retrained. The number of classes of the last fully connected layer was set

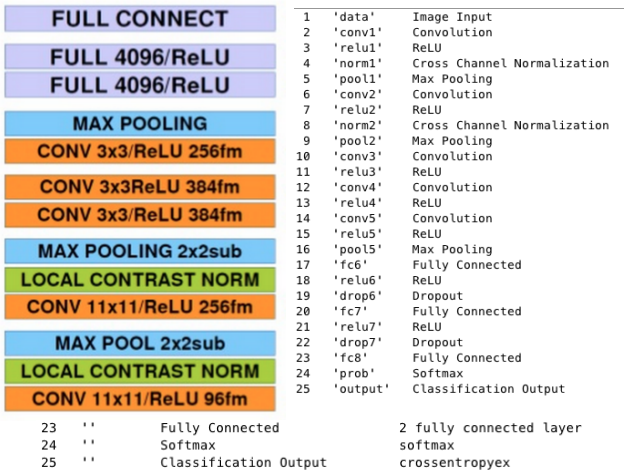

Fig. 5: Detailed architecture of AlexNet as implemented in Python (top left), AlexNet implemented in MATLAB (top right) and the modified MATLAB layers (bottom). The dropout layers (not shown in the top left) are inserted after the first and second fully connected layers. Filter size is shown as NxN, the number of filters as Nfm and the stride as NxNsub for a number N.

to 2 and this layer was trained on our datasets. The original MATLAB implementation, as well as the implementation of the modified layers, are displayed in Fig **??**.

V. Results

The minimum losses achieved for each region were obtained with the pretrained AlexNet and are displayed in Table I and the losses obtained at each training step in Figure **??**. The minimum validation losses achieved for the two regions were 12.21% for Region 5 and 0.88% for Region 17. In comparison, the original AlexNet achieved a best loss of 35% and 6% for Region 5 and Region 17 respectively.

TABLE I: Minimum Losses

	Training	***Validation***
Region 5	0.0139	0.1221
Region 17	0.0006	0.0088

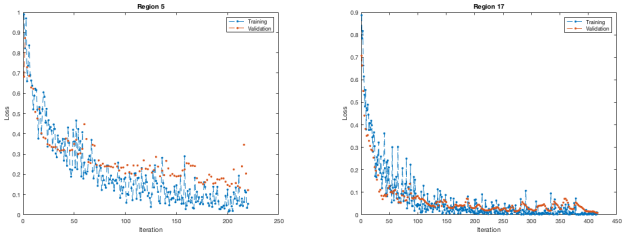

Fig. 6: Training and validation loss for Region 5 (left) and Region 17 (right). Orange represents the validation loss while blue is the training loss.

VI. Conclusion

To help address passengers' concerns of privacy, it was proposed that the scanning and analysis of these scans be freed of human intervention by making use of machine learning algorithms. In this project, AlexNet, a convoluted neural network is implemented on a set of preprocessed images extracted from full body scans. The project is, for the time being, limited to 2 regions of the body - the upper chest and the upper back. When the program was run on a validation set, the log losses obtained were 0.1221 and 0.0088 respectively for the 2 regions, suggesting the algorithm is very accurate. The average log losses reported on Kaggle for all regions is 0.0242%, and the average of our 2 regions puts us in the top 10. The results are, however, based on the synthetic data provided in dataset made available by the TSA. Although real world performance of the program was not tested, it is reasonable to expect similar accuracy in detecting presence of threats in real scans.

VII. Future Work

For sake of simplicity and because of constraints in time, the program was only run on inputs that were manually cropped from a single slice. To be able to identify threats from any part of the body, including curved parts of the body, such as parts of the thighs and calves, a new algorithm would have to be developed to automatically crop the images (one image for every slice- approximately 20 degrees of the 360 degrees body scan) of the scanned subject into the predefined regions, then pass them to the main program to identify presence of threats, and finally, combine the information from these 16 slices to affirm the presence of a threat on the entire body. Future work might also include development of the neural network portion of the program to identify the kind of threat detected using object detection strategies.

Acknowledgment

This project was done as a part of the machine learning course ECE 228: Machine Learning for Physical Applications at UCSD. The original dataset was obtained from the Kaggle Passenger Screening Algorithm Challenge.

References

[1] Martin, Hugo. "Treating TSA agents better might reduce airports' long lines." LA Times. Aug 15, 2016.

[2] Kaggle Passenger Screening Algorithm Challenge. https://www.kaggle.com/c/passenger-screening-algorithm-challenge

[3] Guimaraes, Abel Ag Rb, and Ghassem Tofighi. "Detecting zones and threat on 3D body in security airports using deep learning machine." arXiv preprint arXiv:1802.00565 (2018).

[4] Thomas W. Rogers, Nicolas Jaccard, Edward J. Morton, Lewis D. Griffin, "Automated X-ray image analysis for cargo security: Critical review and future promise", Journal of X-Ray Science and Technology, vol. 25, pp. 33, 2017.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. "ImageNet classification with deep convolutional neural networks." In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'12), F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.), Vol. 1. Curran Associates Inc., USA, 1097-1105.