International Journal of Combined Research & Development (IJCRD)
eISSN:2321-225X;pISSN:2321-2241 Volume: 11; Issue: 4; April -2022

OBJECT DETECTION AND CLASSFICATION ON X-RAY BAGGAGE SCANNING DATASET USING NEURAL NETWORK

Harsh Shekhar — *harshshekhar24@gmail.com* — *Vellore Institute of Technology*

Dr. Rajkumar Soundrapandiyan — *rajkumars@vit.ac.in* — *Vellore Institute of Technology*

Chiranjeev Kaur Bullar — *chiranjeevkaurbhullar@gmail.com* — *Vellore Institute of Technology*

ABSTRACT

Our aim for this project was to create an automatic system for security baggage scanning at the airports by training and testing three different neural network algorithms on x-ray baggage image dataset which would include various firearm component which would be detected by the algorithms and then these algorithms would be compared on various parameters. These parameters were compared to conclude the best algorithms out of the three for x-ray image dataset scanning.

KEYWORDS

X-ray, Dataset, CNN, FRCNN, YOLO v2, R-FCN, True Positive, False Positive, True Negative, False Negative, Accuracy, True Positive Rate, False Positive Rate, Precision, F-score

1. INTRODUCTION

As everyone would have experienced on airports that there is a huge delay while security check due to manual checking of X-Ray baggage scans which is time taking as human mind takes more time to capture the image and analyze it thoroughly to draw conclusions. For some baggage they have to cross check it again and again as they are not sure of their decision, all this causes a lot of delays in the process. Our objective is to minimize the delay as well as increasing the precision of the decisions by creating an optimized automated system for baggage scanning using neural networks which would give more efficient results, would be able to identify if there is any harmful or illegal object present in the baggage or not.

For implementation purpose we have used an x-ray baggage image dataset and implemented it on three neural network algorithms namely- FRCNN, YOLO v2 and R-FCN, which would detect various components like – guns, knife, pliers and wrenches. After testing we compared the algorithms using various parameters, we have divided these parameters as primary parameters and secondary parameters. The primary parameters (True Positive, False Positive, True Negative and False Negative) are the ones which we observed manually through the testing images and secondary parameters (Accuracy, True Positive Rate, False Positive Rate, Precision, F-score) are the one which are calculated using the primary parameters. Using these parameters we finally concluded the best algorithm out of the three for such application.

2. SYSTEM ARCHITECTURE

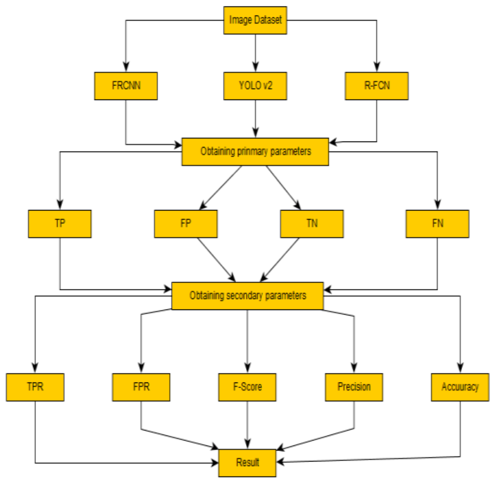


Figure 1 - System Architecture

Figure 1 shows our system architecture and our whole process of implementation.

We started by obtaining a suitable image dataset of x-ray baggage scans and then we trained and tested the three neural network models, after the testing was completed we manually observed the primary parameters and using these primary parameters we calculated the secondary parameters and finally by comparing these parameters we moved towards our result and conclusion.

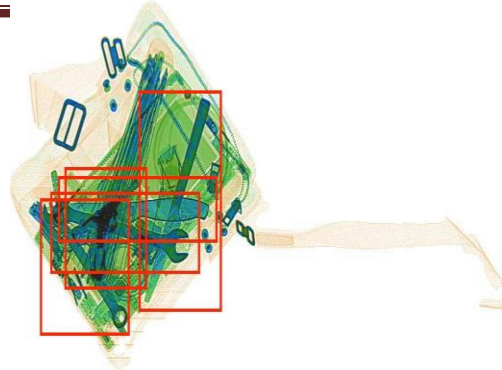
The implementation details of all the three algorithms are given below -

2.1 FRCNN

Faster RCNN takes image as input and applies various convolution layers on it to obtain feature maps, from which the region proposals are obtained using RPN. These region proposals are then provided to the classifier after going through ROI pooling which is basically applying max pooling but only on the region of interests rather than the whole image.

We implemented the FRCNN code on Google CoLab platform as it provided us with online GPU on its servers, but the session time provided to us was of only 10 hours within this 10 hours we were able to train on an image dataset of 500 images for 500 epochs consisting of 50 iterations each thus undergoing a total of 25000 rounds of training.

We trained and tested on only 3 out of the 4 components and the pliers component was left for this algorithm as the number of images for training



were too less and training only 500 images for 4 components would have resulted and in much more drop in accuracy for all the components.

Once the training was completed testing was done on 200 images and the parameters were noted for each component separately.

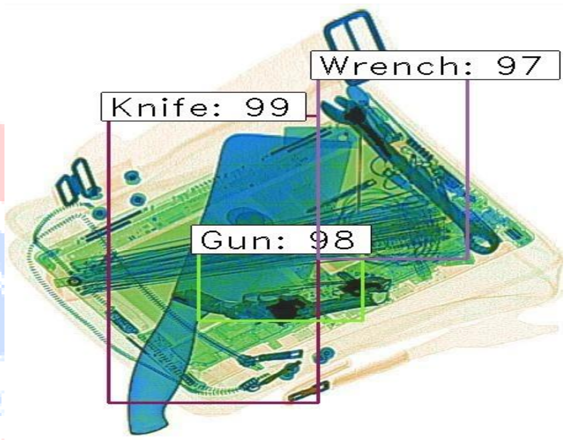


Figure 2 – Sample output of FRCNN

Figure 2 shows a sample output of our FRCNN code implementation , here we can clearly see a gun, a knife and a wrench detected successfully with their labels and percentage of match with the class.

2.2 R-FCN

The R-FCN algorithm is similar to FRCNN code where an image is taken as input then CNN layers are applied to get feature maps. The difference is that the RPN layer is absent in R-FCN and in place of RPN, ROI pooling is done alongside with generation of score maps which would map the class vote to each of the features detected and thus as per the vote scores the objects would be classified.

We trained the R-FCN code on Kaggle Platform similar to Google CoLab, it is a platform which

International Journal of Combined Research & Development (IJCRD)
eISSN:2321-225X;pISSN:2321-2241 Volume: 11; Issue: 4; April -2022

provides us with free online GPU for our training and testing purposes.

Kaggle also provided us with 10 hours of session time for training; within this time period we were able to train our system for 800 images and for all the 4 components.

The training was completed in 4 stages –

Stage 1 – 240 epochs

Stage 2 – 480 epochs

Stage 3 – 960 epochs

Stage 4 – 1920 epochs

And each epoch consisted of 4 iterations.

Once the training was completed we implemented our model on 200 images.

Figure 3 – Sample output of R-FCN

Figure 3 represents a sample output image of R-FCN algorithm where we can observe various components detected in red boxes.

2.3 YOLO v2

Yolo v2 algorithm takes image as the input and applies convolution layers followed by max pooling layers for feature extraction by decreasing the resolution of the image and increasing the depth of the image.

The input image is divided into squares and on each of those square grids the algorithm predicts five bounding boxes each of them with different aspect ratios. After obtaining the bounding boxes, the algorithm predicts the center of the box and then calculates the confidence score of having any object in that square along with the probabilities of which class the object belongs to, in our case we have set the threshold of the confidence score and probability both at 0.3.

We have implemented the YOLO v2 algorithm on Google CoLab platform which provided us with 10 hours of session time just like the other algorithms and in these 10 hours we were able to train 1000 images which is the maximum among the three which clearly indicates YOLO v2 is the fastest among the three. We trained for 50 epochs but the execution went under early execution at 36 epochs due to no decrease in the loss function since the last three epochs.

The training was conducted for all the four objects and after the training was completed we tested the model for 200 images.

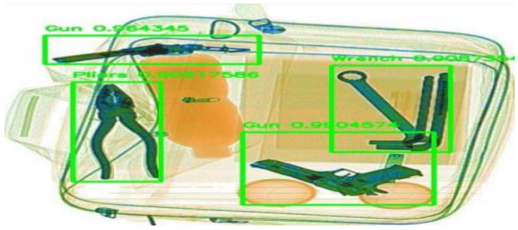


Figure 4 – Sample output of YOLO v2

Figure 4 represents a sample output of YOLO v2 where we can see two guns, one wrench and one plier is detected in green boxes with labels as their class name and probabilities of matching.

2.4 PARAMETERS

We have divided the comparison categories as primary categories and secondary categories –

2.4.1 PRIMARY PARAMETERS

Primary parameters are the one which we observed manually from the tested results.

TP: True Positive, which was determined by how many correct predictions the algorithm made, i.e. how many components were correctly detected.

TN: True Negative, which is determined by how many negative predictions were done correctly, i.e. if any particular component is absent in the image then the system, should not detect the component in that image.

FN: False Negative, which is determined if any component is left undetected in any image.

FP: False Positive, which is determined if any component is wrongly detected or a blank space is detected as one of the component.

2.4.2 SECONDARY PARAMETERS

These are the parameters which are calculated from the primary parameters.

TPR: True Positive Rate is given by the following formula –

TPR = TP/ (TP+FN)

It is also known as Recall it determines out of the present components, how many components were predicted correctly.

FPR: False Positive Rate is given by the following formula –

FPR = FP/ (FP+TN)

It determines out of all the blank spaces how many were wrongly detected as a component.

Accuracy: Accuracy is calculated by the following formula –
Accuracy = TP+TN/ (TP+FN+TN+FP)

It determines the ratio of total number of correct predictions of presence and absence of components to the total number of cases.

Precision: Precision is calculated as –

Precision = TP/ (TP+FP)

It determines how many detections made are correct.

F-score: It is the harmonic mean of Recall and Precision. Hence the formula is,

F-score = 2* Precision * Recall / (Precision + Recall)

Higher the value of F-score more perfect is the Precision and Recall.

3. RESULTS AND DISSCUSSION

We have tested each of the algorithms for 200 images. For analyzing the results we have calculated the various parameters for each component to be used for comparison analysis between the algorithms.

The comparative analysis would be done in tabular form for each component.

3.1 GUN COMPONENT

Table 1 - Gun Component parameters for all 3 algorithms

	YOLO v2	FRCNN	R-FCN
TP	277	131	59
FP	28	21	12
TN	48	103	113
FN	7	7	9
TPR	0.975	0.95	0.867
FPR	0.368	0.17	0.096
Accuracy	0.902	0.89	0.86
Precision	0.908	0.86	0.83
F-score	0.940	0.90	0.848

As we can observe in the above table TPR is highest in YOLOv2 i.e. 0.975 or 97.5% algorithm that means the algorithm was able to detect 97.5% of the samples correctly.

Accuracy, Precision and F-score are also observed highest in YOLO v2 algorithm, hence we can successfully conclude from these results that the YOLO v2 algorithm gave the best results for the Gun component.

3.2 KNIFE COMPONENT

Table 2 - Knife Component parameters for all 3 algorithms

	YOLO v2	FRCNN	R-FCN
TP	190	275	117
FP	18	7	15
TN	69	0	76
FN	27	97	11
TPR	0.875	0.74	0.914
FPR	0.206	1	0.164
Accuracy	0.852	0.725	0.965
Precision	0.91	0.975	0.886
F-score	0.892	0.841	0.899

As observed in the above table TPR and Accuracy is highest in R-FCN for the knife component that indicates R-FCN was able to detect most samples correctly.

The precision is highest in FRCNN i.e. 0.975 or 97.5% which implies that majority of the predictions made by the algorithm were correct, but there is a vast difference in its accuracy and precision as it had left lots of samples undetected which had hamper its accuracy. So, even though the algorithm made minimum error when it detected the component but it had left lots of samples undetected. Which has also affected its F-score which implies the TPR and the Precision is not that perfect.

The FPR observed in FRCNN is 1 or 100% that is because its TN value is 0 which means there was no image in which the Knife component was absent.

As observed the F-score is highest in R-FCN algorithm which implies its TPR and Precision are the most perfect.

3.3 WRENCH COMPONENT

Table 3 - Wrench Component parameters for all 3 algorithms

	YOLO v2	FRCNN	R-FCN
TP	53	33	26
FP	2	3	8
TN	145	142	110
FN	4	76	3
TPR	0.92	0.3	0.896
FPR	0.013	0.02	0.067
Accuracy	0.97	0.688	0.68
Precision	0.96	0.916	0.764
F-score	0.939	0.45	0.824

Results are clearly conclusive as YOLO v2 has best TPR, FPR, Accuracy, Precision and F-score as compared to the other two algorithms.

There is a vast difference in the accuracy and precision of FRCNN algorithm because though it was able to detect the components accurately with very less number of wrong detections but it left out a lot of samples undetected due to which it's accuracy was hampered greatly. And so is its F-score which has fallen even below 50%.

3.4 PLIERS COMPONENT

Table 4 - Pliers Component parameters for 2 algorithms

	YOLO v2	R-FCN
TP	60	8
FP	0	4
TN	142	120
FN	0	1
TPR	1	0.88
FPR	0	0.032
Accuracy	1	0.64
Precision	1	0.66
F-score	1	0.758

FRCNN is not present since it was trained for only 500 images and thus it couldn't be trained for all the 4 components thus, Pliers component was absent in training as well as testing of FRCNN.

For the YOLO v2 algorithm there were no wrong detection and neither was any Pliers left undetected thus its FN and FP values are 0 and since FP was 0 thus FPR also became 0%.

The Accuracy and Precision are 100% since there were no errors found in detecting the Pliers component using YOLO v2 algorithm.

So clearly, YOLO v2 algorithm gave best results for detecting Pliers component with an F-score of 100% which implies that it's TPR and Precision has achieved perfection.

4. CONCLUSION

We were successfully able to implement 3 algorithms namely – YOLO v2, FRCNN and R-FCN, on the online platforms. We had successfully trained all the algorithms and tested these algorithms on 200 images each.

We have accurately calculated all the parameters for each component namely, TP, TN, FP, FN, TPR, FPR, Accuracy, Precision, F-score.

And observing these results we can reach to the conclusion that out of the three, YOLO v2 is the best algorithm for detecting various firearms components in X-ray Baggage scanning, as in the training period YOLO v2 was the fastest to execute and was able to train on 1000 images which is the highest compared to the other two algorithms, and it gave the best results for most of the components except the Knife component where R-FCN had the best F-score but even in that case the difference between F-score of R-FCN and YOLO v2 was minimal.Hence, we can arrive on the conclusion that YOLO v2 is the best algorithm to implement on X-ray Baggage Dataset.

5. REFERENCES

1. M. Bas ̧tan, M. R. Yousefi, and T. M. Breuel, “Visual words on bag- gage X-ray images,” in Computer Analysis of Images and Patterns. Berlin, Germany: Springer, 2011, pp. 360–368.

2. D. Turcsany, A. Mouton, and T. P. Breckon, “Improving feature- based object recognition for Xray baggage security screening using primed visualwords,” in Proc. IEEE Int. Conf. Ind. Technol., Feb. 2013, pp. 1140 1145.

3. M. E. Kundegorski, S. Akçay, M. Devereux, A. Mouton, and T. P. Breckon, “On using feature descriptors as visual words for object detection within X-ray baggage security screening,” in Proc. Int. Conf. Imag. Crime Detection Prevention, Nov. 2016, p. 12.

4. D. Mery, E. Svec, and M. Arias, “Object recognition in baggage inspec- tion using adaptive sparse representations of X-ray images,” in Image and Video Technology. Cham, Switzerland: Springer, 2016, pp. 709–720.

5. D. Mery, V. Riffo, I. Zuccar, and C. Pieringer, “Object recogni- tion in X ray testing using an efficient search algorithm in multiple views,” Insight Non-Destructive Test. Condition Monitor., vol. 59, no. 2, pp. 85–92, 2017.

6. D. Mery, “Automated detection in complex objects using a tracking algorithm in multiple X-ray views,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops, Jun. 2011, pp. 41–48.

7. T. Franzel, U. Schmidt, and S. Roth, “Object detection in multi-view X- ray images,” in Pattern Recognition. Berlin, Germany: Springer, 2012, pp. 144–154.

8. N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), vol. 1. Jun. 2005, pp. 886–893.