

# Threat Detection using Machine Learning in Public Places

Vansh Gupta\*, Manya Kakkar<sup>†</sup>, Jai Garg<sup>‡</sup> and Dr. Sanjay Kumar<sup>§</sup>

Electronics and Communication Engineering Department,

Dr. Akhilesh Das Gupta Institute of Technology & Management\*, New Delhi, India

**Abstract**—Airports are subject to tremendous demand and at-tempt to meet a variety of interconnected performance objectives. Airport aspects, such as safety and security, also include inside baggage detection of dangerous items. This subject is irrespective of the size and importance of the airport. A large proportion of aviation safety is the identification of risk objects with X-ray baggage scanning pictures. Currently, most screens still depend largely on human specialists to discover potentially dangerous items manually. Presently, automated learning, which enables computers to find solutions to problems themselves, is the most fascinating branch of artificial intelligence. The study introduces a threat object identification system in ML, such as Open-CV and Keras with Tensor Flow using the convolutions of neural neural networks and specific libraries. The themes discussed include the creation of the neural network, data increase strategies utilised, testing and detection rates utilising X-ray images

**Index Terms**:- Computer vision Deep learning, automatic de-tection, threat object detection, machine learning, airport safety.

## I. INTRODUCTION

We may argue that machine learning is the most fascinating artificial intelligence branch. With computers being able to learn how to solve issues alone, machine learning has pro- duced several breakthroughs that formerly appeared virtually unattainable. Computers can understand human speech, recog- nise an object or detect the face of the human person and the car can also be used by computers.

The interaction of the human-machine system has also been strengthened by the development of robots that can imitate human behaviour[1]. But every robot is restricted and only one domain may be specialised. However, technical progress continues, and a computer is attempting, like a growing kid, to develop and expand its knowledge area alone.

While now one of the technologies most sought after and is expanding very fast in many sectors, it is fairly difficult to change artificial intelligence from one level to another and until the grand dream of people takes time to develop robots like humans[2].

Today, imaging gadgets have gone to a new level by provid- ing better functionality and broadening their range. For many picture-related activities, automatic image content processing is helpful. The processing of a picture in computer systems entails transmitting information at the pixel level to people with the same images.

With the growth of computer power, mechanical education takes a shorter time to scan all the data necessary to learn.

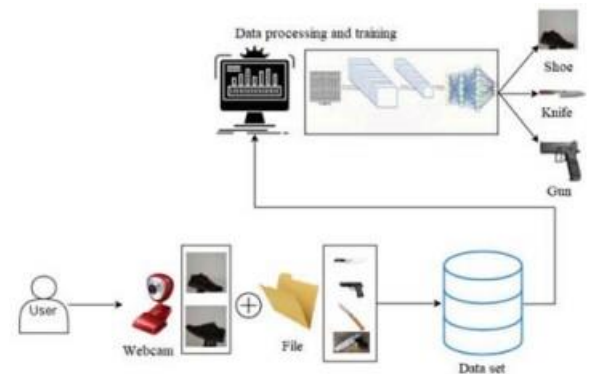


Fig. 1. System architecture

Systems can predict items presented once all pictures have been trained through the study procedure in the article.

This smart object identification system is on a simple camera level, but can be implemented on a very large array of devices that have a number of advantages as well. An example would be items identified and shown on a monitor during the check line at airports.

## II. PROPOSED METHOD

The system presented is one for the recognition at the airport of hazardous objects. The overall architecture of the system is shown in figure 1 and the stages are performed to achieve the ultimate outcome.

It must be trained to achieve the final level and the system of properly recognising the things. The workout is conducted utilising profound learning algorithms applied to the data collection. There are two techniques to load the data set:

- The first way is to activate the webcam using a code and capture the picture with a key with a specific function on real- time computer view.

- The second approach is the loading of the data .

Once the data set has been prepared, a number of operations follow it to become a numerical form.

The training model was developed after getting the data to the necessary form. A variety of layers are used to filter, simplify and model frames in this model. A probability vector that frames the content of the picture in an initially generated image class is the output of this model or network.

## A. Implementation

### 1) Data Collection

OpenCV is a library dealing with computer vision in python, which is designed to make applications computational. This library contains an application based on the web camera of the computer. A reference to the computer's regular webcam was established once it was imported. He built an item for VideoCapture.

The object's parameter is the room index we refer to, as seen in figure 2 for our index 0. This change enables us to obtain the desired outcomes by providing much more information on the computer. Data may comprise various information, such as the distance at some point from the framed item or the camera. The motion of the camera in an image or image transformation sequence in black and white tones is another characteristic of computer vision.

The processor will perceive the pictures in numeric arrays and the noise component of any values in that array. This is the maximum that computer vision can accomplish. An example of picture coding is shown in Figure 3.



Fig. 2. Image encoding

The data vector is now ready to fit the model function, which also creates a matrix that has the binary values that match the labels of each data set. First, in the Num Py library we have the 'concatenate' feature, and then in the 'keras.utils' 'to categorical' feature (figure 5).

```
77 def incarcare_imagini(calea_de_baza):
78     images = []
79     path = os.path.join(calea_de_baza, '*.png')
80     for cale imagine in glob(path):
81         image = preprocessing.image.load_img(cale imagine, target_size=(width, height))
82         x = preprocessing.image.img_to_array(image)
83         images.append(x)
84     return images
```

Fig. 3. The image processing function

2) Structure of the CNN CNN used consists of:

The fundamental unit of these sorts of neural networks is two convolutionary layers. A converting layer conducts a sequence of linear filtering operations in the matrix, using

```
96 data = np.concatenate((data_type_1, data_type_2, data_type_3, data_type_4), axis=0)
97 data = data/255
98 labels = np.concatenate((label_type_1, label_type_2, label_type_3, label_type_4), axis=0)
99 from keras.utils import to_categorical
100 labels = to_categorical(labels, num_classes=len(class))
```

Fig. 4. Example of The concatenation functions

image processing techniques, resulting from the input picture or image from the preceding layer, also known as the feature map. • Two levels of pooling – replace the area value in the feature map with the area statistics. Within the system, the grouping feature MaxPooling was employed. This substitutes a highly defined region with the maximum area, leading to a smaller map while keeping the most essential information. Four drop-out layers - The drop-out layer is extremely useful to prevent overpackage in neural networks. This layer assumes that some layer weights are disregarded and that several networks with features are simulated with fewer. • A single layer of flattening — after certain convolutionary layers it must be possible to utilise completely linked layers. The tensor derived from the convolution layers therefore becomes a 2D vector. • The dense layer is the standard layer of an MLP network, with weight nodes and compensations 3 dense (compact) layers (bias).

The grouping layers (we set the size of the window for each layer) and the drop-out layers (we define the ignition rate for each layer) normally occur in a sequence of convolution layers. The flattened layer is then displayed. Then we have fully linked layers (the number of neurons is defined for every layer).

A table describes the architecture and number of layers of the neural network:

TABLE I. THE ARCHITECTURE OF THE NEURAL NETWORK

Layer	Neurons/ Filter number	Filter dimensions	Activation functions	Ignorance rate
Convolutions	16	3 x 3	ReLU	-
Pooling	-	2 x 2	-	-
DropOut	-	-	-	20%
Convolutions	32	3 x 3	ReLU	-
Pooling	-	2 x 2	-	-
DropOut	-	-	-	20%
Flatten	-	-	-	-
Dense	1024	-	ReLU	-
DropOut	-	-	-	20%
Dense	512	-	ReLU	-
DropOut	-	-	-	20%
Dense	nr. of class	-	Softmax	-

It must initially be compiled prior to training the model. The "compile" function is used to compile. Three parameters are required for this function: a loss calculation tool, an optimization tool and a measuring technique. It will now be

assembled after defining the model generation function. In order to training the model, it requires the data acquired in the form of an image, the labels (in the form of a binary-value matrix) and the time (epoch). The model itself was trained by use of the fitting feature.

### III. RESULTS

Three types of items such as weapons, cords and grenades have been selected to observe the performance of the program. The following findings have been achieved after the testing of the data sets comprising of these three classes, as shown in the Table II.

TABLE II. OBTAINED RESULTS

Test	Totally trained samples	Number of samples trained per class	Classes	Recognized samples from 150	Recognized samples from 150(%)	Data processing time (seconds)
1	150	50	guns	68	45.33%	10
		50	knives	132	88%	
		50	grenades	63	42%	
2	660	220	guns	98	65.33%	320
		220	knives	111	74%	
		220	grenades	62	41.33%	
3	1242	370	guns	149	99.33%	600
		430	knives	138	92%	
		442	grenades	82	54.66%	

The first exam was conducted in 10 seconds in a really short period, but was not very excellent, save for the class with knives. The second exam had roughly five and a half minutes of somewhat longer processing time. The performance was not particularly encouraging, but rose in the case of firearms. It took 10 minutes for the third test. Although the non-processing time has doubled, all three courses have achieved considerably better outcomes, even with a 100 percent ratio of weapons classes.

### IV. CONCLUSION

We may thus infer that the performance in identification of the application also rises, albeit not always, by increasing the training data set. The increase in the data collection also results in increased processing time. Several hazardous things were chosen as identification models, including a knife, a pistol, or a grenade. Articles which an aircraft passenger would not be allowed to have.

In the future, it will enhance the system in order that the computer is simulated on an airport to recognise and notify the unpacked objects in your luggage. For this, the kind of items not permitted should be increased and the network improved to improve precision.

### REFERENCES

- [1] K. Dautenhahn, "Socially intelligent robots: dimensions of human-robot interaction", *Philos Trans R Soc Lond B Biol Sci.*, Volume 362, Issue 1480, pp. 679-704, 2007.
- [2] A. Rathi, "The impact of Artificial Intelligence", A critical review of opportunities risks of AI adoption, 2019.
- [3] E. Go'mez, C. Castillo, V. Charisi, V. Dahl, G. Deco, B. Delipetrev, N. Dewandre, M.A' . Gonza'lez-Ballester, F. Gouyon, J. Herna'ndez-Orallo, P. Herrera, A. Jonsson, A. Koene, M. Larson, R. Lo'pez de Ma'ntaras, B. Martens, M. Miron, R. Moreno-Bote, N. Oliver, A.P. Gallardo, H. Schweitzer, N. Sebastian, X. Serra, J. Serra', S. Tolan, K. Vold, "Assessing the impact of machine intelligence on human behaviour: an interdisciplinary endeavour", *Proceedings of 1st HUMAINT workshop*, Barcelona, Spain, 2018.
- [4] B. Gesing, S.J. Peterson, D. Michelsen, "Artificial Intelligence in logis- tics", A collaborative report by DHL and IBM on implications and use cases for the logistics industry, 2018.
- [5] Y. Lu, L. Luo, D. Huang, Y. Wang, L. Chen, "Knowledge Transfer in Vision Recognition: A Survey", 2019
- [6] O. Russakovsky, H. Su, "ImageNet Large Scale Visual Recognition Challenge", *International Journal of Computer Vision*, 2014.
- [7] F. Chollet, "The limitations of deep learning", *Deep Learning with Python*, Section 2, Chapter 9, Essays, 2017.
- [8] C. Kone, "Introducing Convolutional Neural Networks in Deep Learn- ing", *Data Science*, 2019.
- [9] V. Nair, G.E. Hinton, "Rectified Linear Units Improve Restricted Boltz- mann Machines", *International Conference on Machine Learning*, 2010.
- [10] M. Ponti, E.S. Helou, P.J.S.G. Ferreira, N.D.A. Mascarenhas, "Image Restoration Using Gradient Iteration and Constraints for Band Extrapo- lation", *IEEE Journal of Selected Topics in Signal Processing*, Volume: 10, Issue: 1, pp. 71-80, 2016
- [11] L. Wang, W. Ouyang, X. Wang, H. Lu, "Visual Tracking with Fully Convolutional Networks", *IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [12] H. Nam and B. Han, "Learning Multi-domain Convolutional Neural Networks for Visual Tracking", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [13] D. Griffiths and J. Boehm, "A Review on Deep Learning Techniques for 3D Sensed Data Classification", *Remote Sens.*, Volume 11, Issue 12, 2019.
- [14] M.A. Ranzato, F.J. Huang, Y.L. Boureau, Y. LeCun, "Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition", *IEEE Conference on Computer Vision and Pattern Recog- nition*, 2007.
- [15] M. Blum, J.T. Springenberg, J. Wu'lfing, M. Riedmiller, "A learned feature descriptor for object recognition in RGB-D data", *IEEE Inter- national Conference on Robotics and Automation*, 2012
- [16] A. Coates and A.Y. Ng, "The Importance of Encoding Versus Training with Sparse Coding and Vector Quantization", *Proceedings of the 28th International Conference on Machine Learning*, ICML 2011, Bellevue, Washington, USA, 2011.
- [17] C. Farabet, C. Couprie, L. Najman, Y. Lecun, "Scene Parsing with Multiscale Feature Learning, Purity Trees, and Optimal Covers", *29th International Conference on Machine Learning*, 2012.