Human Violence Detection Using Deep Learning Techniques

Human Violence Detection Using Deep Learning Techniques

Arun Akash S A, Sri Skandha Moorthy R, Esha K, Nathiya N

Division of Mathematics, School of Advanced Sciences, Vellore Institute of Technology, Chennai, Tamil Nadu 600127, India.
E-mail: nadhiyan@gmail.com

Abstract. The world's average annual fatality rate from human violence is 7.9 per 10,000 people. Most of this human violence takes place in an isolated area or of sudden. The information delay here is a major impediment in stopping these acts. To thrive on this issue, the detection technique is used in this study. Detecting moving objects from CCTV is one of the most effective computer vision algorithms. CCTV cameras are now in every streets which are extremely helpful in solving cases. Some techniques of deep learning are used as computer vision to predict and detect the action, properties from video. In real-time police reach violent destinations and start checking CCTV cameras, and investigate to proceed further. This study is deliberately designed to detect violent acts from CCTV cameras. The Inception – v3 and Yolo – v5 models detect the violent act, the number of persons involved, and also the weapons used in the situation. The study consists of these deep learning models, which are used to form a video detection system. This model can be used in real-time as an application programming interface (API) or software. The study results showed the proposed model achieves an accuracy of 74%.

Keywords: Closed-circuit television (CCTV), Human violence, deep learning, Machine learning, Transfer learning.

1. Introduction

The technological improvement in video and image processing has been exceptional due to the importance of finding the contents for various applications which includes recognizing actions and objects they use like knives or guns [1]. The recognition of finding the actions from video streams has been improving in recent years only because of the rise of human violence happening in our daily life. The footage of this surveillance is usually detected manually. There are millions of cameras placed around the world, even if the percentage of human-violence may be low but still, the dangers can happen at any place. This is to estimate the current situation of human-violence systems and about deep learning techniques and methods involved in it. This technology is to detect the objects, movements, and activities they perform and with the data, we can merge the technology to detect human-violence activities which happen every day [2].

The deep learning algorithms are used to detect violent activity automatically. This algorithm involves various stages such as object detection, action detection, and video classification [3]. We are in the aim of creating a system that is used to detect violent activities without the presence of humans. With the help of transfer learning the Google Net –Inception – v3 an image classification model and Yolo – v5 object and face detection model are used to recognize the human violence, objects present in the video [4]. In this study, the machine learning pre-trained model Inception – v3 is used. It is ahead of the basic structure of an Inception v1 and v2 C.V. models. Inception – v3 models are trained on the image net datasets and it also has an elaborated information to retain inceptions to top layers. In

object detection, the Yolo – v5 model has a good accuracy rate with a lower error rate. More than 80 different labels have been detected by Yolo – v5 which has better accuracy than its predecessor Yolo – v4.

2. Methods and Materials

Recognizing individuals by a visual surveillance system helps us to identify security-related problems throughout the world [5, 6]. The deep learning model is used to identify the number of persons involved in a strange video or in a human violence-related video. Human violence detection activities are done using deep learning models and to detect objects various object detection models are used. The given video frame will be resulted as violent or non- violent video and store the frame into the database. The detection of violence using a high population and high dimensional data can be thrived by using the faster learning algorithms and classification techniques identifies the violence, then alert the system [3]. With the help of improved Fish vector and also using the sliding window approach the detection of violence can be recognized more accurately then the previous IVF method with their spatio-temporal positions. This makes the violence detection significantly faster [7]. With the help of YOLO – v3-MT model and AS-CBAM, dilated convolution, 1x1 convolutional layers are encapsulated as single one to detect the objects accurately [4]. Motion detecting techniques to automatically detect the violence can be implemented in the appropriate datasets. Mosift algorithm is mainly used in detection of violence. With help of these the machine can indicate the real time events and stimulated fight events occurred [8]. With help of the VGG net and ResNet-50, the weapons like hunting refile, knives, pistols and revolvers can be detected. Images from of these weapons are collected from sources, trained and tested which results as a better detection model [1].

In this study, we use both Inception – v3 and Yolo – v5 model for a single agenda, i.e., to detect the violence activities taking place and also recognize the objects involved in it. The reason to select two different models is to have better results. Inception v3 model has great accuracy rates in classification and Yolo v5 model results better accuracy in object detection. In the previous works these two models were used to again different agenda but in this project we combined both (as API) as to attain our agenda. Different data sources were collected to bring out efficient results.

2.1. Deep Learning

It's a whole intersection point of artificial intelligence, where artificial intelligence (AI) is the key base platform for deep learning. The term derives the nature of studying everything in detail form for analysis and conclusion. Humans can do a handful of operations and analyses at a certain time only, if the quantity is less means the data is less, imagine something we humans want to get some detail to report for 100000 text which is highly impossible and takes huge time. For these issues, we can build a model to understand and absorb the learning information after which the model can do some operation to give the required results and conclusion. Deep learning is the operation done with machine learning techniques that makes the supervised and unsupervised function effective. It talks about the model construction and the inner architecture. We can access these techniques using a python programming language which is open-source software. These techniques have a mathematical concept behind them and change all the theoretical matter to numerical concepts to understand all the relationships between all the information. Python has various open-source functions where all the mathematical operation changes into py function and it can be used via packages.

2.2. Packages

A unit word that contains py function which can be mathematical, statistical, word processing, or binary action. These packages reduce the time to construct the model and architect the networks, to install the packages we use !pip command.

NumPy: A starter for a python code; it contains all the basic functions which perform numerical manipulation and access binary data.

Google –Drive: A cloud software can be floating external hardware for python. To access the data and store data, google drive has a package that contains the function to bind the cloud server and python work-base together.
Pandas: If your working data is in a structured form that needs to be added to the constructed model, pandas will help them to the convection process.
OpenCv: A computer vision package that helps in reading images, converting video into data frames, and also saving video or images in any format.
FastAPI: A web framework that helps in developing RESTful API in python. With the help of this, the developed model can be implemented on any website as API.

2.3. Neural Networks
Neural networks exactly imitate the process of a neuron. This neural has Input layer, hidden layer and output layer. Neural networks work with several processes of layers that are known as the perceptron. This technique are used in various fields such as forecasting and detection systems.

2.4. Transfer Learning
This technique reduces the huge computational knowledge with pre trained modeling. So, using deep learning models is a common thing to do with pretrained for challenging models [9]. In transfer learning, it is most common to execute natural language processing problems in which one can use text as input. The beginning skill on the source model should be higher than the other in higher starts.

2.5. Pre-trained Model
Constructing a new high-performance model is a little time-consuming job and adding the mathematical function into networks is hard work, we know by transfer learning we can append the features from one model to another model. Some so many open-source models will have the same motivation and techniques, so we can take the required feature function for building our model.

2.6. Yolo – v5
It is a sequence-based entity detector, which has a single flow through the neural networks. The main object of this model is to learn the object boxes on their own after one epoch of train data and produce a high speed in training and testing the given information. The networks have three main layers (Figure 1).

2.6.1. Back Structure
In this block, we will absorb the features of the pictures, basically it is convo layer in neural networks, and it uses CSPNET which is a type of convo layer. This is divided into 2 structure layers, in one section it will start convo layer working like padding and max pooling, padding is used to extract the feature from of images by adding extra binary data at the edges of the images, this will be helpful and acts during the loss of the information, padding is to minimize the pixels size of the images without the loss of the feature's information.

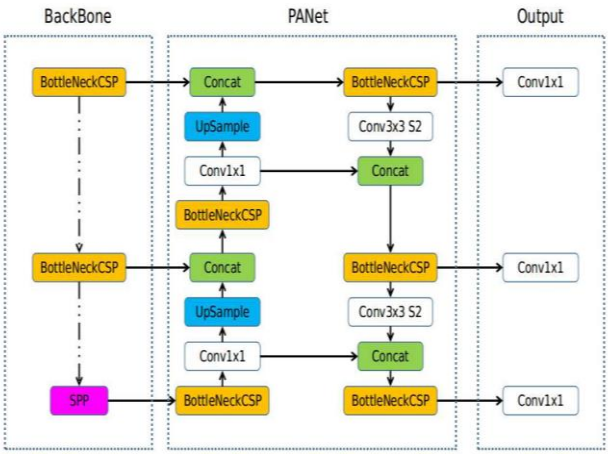

Figure 1. Architecture of Yolo – v5 model.

The other part of CSPNet flows the information and binds it to the transition layer, which contains the outline map of the information's features. The data passes through only one way because it uses the feedforward neurons, here the process is learned the old weights by using the various nullifying gradient descent function and this function works with separated phrases in dense layers. Only one part can use this technique and we cannot add imaginary weights to the part 1 layer.

2.6.2. Model Junction

To connect the Head and Backbone this layer does that job, it plays the role of a bridge to join the input and output layers. PANet architecture helps to access the breaking point of the images and plays a vital role in the spatial information convention. The weights can be properly aligned and it can be segmented pixel vice. Bottom-up Path Augmentation is the first layer of the planet. While heavy access to the data, may lose the features, there is a chance of spatial data to decrease, and pixel range cannot be detected properly in the next layer, accuracy may also dropdown. To reduce the previous problem the PANet gives the solution of joining the enriched features in a structured manner with the connection of localization data. The layers should be more in number to avoid the long sensitive spatial data. The lower layer has a connecting line segment that joins to exchange the weights and creates shortcut paths.

In the next part of the layer, the shortcut leads is Adaptive Feature Pooling, where all the spatial data will reduce its pixels, These higher-level data will be compressed and the function will take place in both the layers, the pooling method is max pooling there, it will get the max weights and divide by the average of it, this a one by one process where each element in the weight box will go through this process. The fully connected layer is to prepare the data for the final prediction process, this will check and control the segmentation process, and the parameters may affect the spatial data and control flow. The rate of distribution of spatial data is the same in all three layers.

2.6.3. Model Head

The head plays the role to find the prediction answer and the output will be in the matrix form, we need to get it has probabilities values for humans to understand it, it gives the bounding boxes to understand the detected areas. For this feature pyramid network (FPN) is allied in the structure. The FPN is pyramid-style layer construction. It is like long short-term memory (LSTM) the process will go from downwards to upwards, this pathway structure maintains the increase and decreases the flow of spatial segmentation. The detector will have the absorbed features maps and will have a designer role in the output prediction, while the images in feature extraction the area with the object will be identified as objectless detection and it also has an imaginary boundary box. This FPN is faster than all recurrent neural network (RNN) and LSTM.

Table 1. Description of Inception - v3 model.

Category	Packet Size	Input Size
Convolution	3x3/2	299x299x3
Convolution	3x3/1	149x149x32
Convolution with padding	3x3/1	147x147x32
Pool	3x3/2	147x147x64
Conv	3x3/1	73x73x64
Conv	3x3/2	71x71x80
Conv	3x3/1	35x35x192
3 x Inception	Module 1	35x35x288
5 x Inception	Module 2	17x17x168
2 x Inception	Module 3	8x8x1280
Pool	8x8	8x8x2048
Linear	Logits	1x1x2048
Softmax	Classifier	1x1x1000

2.7. Inception - v3 – Human Violence Detection

In this study, Inception - v3 is used as the base model. Inception - v3 is a version of the Google Net model developed by Google for image recognition and classification. Inception – v3 consists of 42 layers and a lower error rate than its predecessors. It has 5 convolution layers, 1 Conv padded layer, 3 Inception layers, and 2 pooling layers with the activation output as 'softmax'. It takes the input dimension as 229x229x3. After passing through two layers it gives the output as the dimension of 8x8x2048. Using the 'softmax' as the classification activation function, it gives the final output as 1x1x1000 (Table 1).

2.8. Confusion Matrix

After building up the model and getting the required result, we need to find whether our model is giving a good result or not. For that we can use a confusion matrix to get the accuracy and the confusion matrix shows the results rate of the models trained [10].

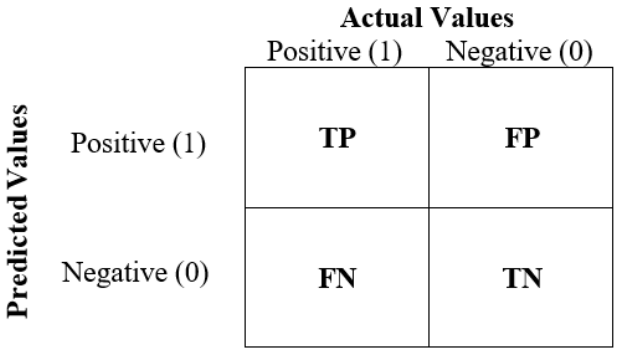

Figure 2. Confusion matrix.

Figure 2 shows that

- Predicted data are denoted as rows
- Actual data are denoted as columns
- The variable value can be either positive or negative
- True Positive: The actual data is positive but predicted as positive
- True Negative: The actual data is negative but predicted as negative

- False Positive: The actual data is negative but predicted as positive
- False Negative: The actual data is positive but predicted as negative

Using the obtained value, the accuracy of the modelling can be found using Equation (1). Recall tells the number of positive predictions that were made out of all positive predictions (Equation (2)). Error rate tells the incorrect prediction of the model in percentage (Equation (3)). If the error rate is less it is set to be the model is performing efficiently.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$Error\ rate = \frac{FP+FN}{TP+FP+TN+FN} \quad (3)$$

3. Data Analysis

3.1. Data Set

The data are in the .jpg format for training and testing our model. For the Yolo - v5 model we should have data of the detected objects in pixelated form. As it is supervised learning, label data is also needed. First, the images of objects like weapons, other objects are collected from different sources and some also from Kaggle. For person detection, the personal images and random faces of the persons from different sources were collected. Google images are also used to collect random person images [1]. To collect huge data with the required custom image sizes and quality, we are adding a chrome extension called imageeye (image downloader) which has a control function feature where we can adjust the size of the picture and it will download all the images in a full one google search entity. For better quality add a size range from 1300 - 2500 for both the height and width of the picture [11].

```
1 0.610259 0.572327 0.654481 0.813417
0 0.741745 0.251572 0.459906 0.461216
0 0.738208 0.733753 0.459906 0.469602
0 0.453420 0.466457 0.878538 0.895
0 0.540094 0.537736 0.794811 0.849057
0 0.903302 0.211139 0.146226 0.286144
0 0.900943 0.504528 0.160377 0.297010
0 0.927476 0.815121 0.135613 0.280711
0 0.750590 0.141414 0.128538 0.237246
0 0.612618 0.126020 0.121462 0.210081
0 0.632665 0.344251 0.104953 0.219136
0 0.363797 0.157714 0.147406 0.266223
0 0.484670 0.253699 0.110849 0.291577
0 0.198113 0.185785 0.141509 0.278900
0 0.057193 0.309841 0.088443 0.179293
0 0.170401 0.480079 0.114387 0.262601
0 0.323703 0.450196 0.147406 0.329609
0 0.492925 0.563386 0.146226 0.280711
0 0.458726 0.851342 0.113208 0.255357
0 0.645047 0.851342 0.132075 0.244490
0 0.782429 0.579686 0.130896 0.244490
0 0.786557 0.848625 0.125000 0.228191
0 0.139741 0.822365 0.154481 0.324176
```

Figure 3. Example of Yolo label data.

To label images, a software called labeling is downloaded in python, which will create a label with their pixel ranges. Use !pip install labeling command, which will download it in the Linux interface. Open the labeling software and add the folder which contains your images. Using the Rectangle Box command, mark the area of the object which you need to label. In our case, we have 4 labels, 2 labels of weapon images, and another 2 labels of people images. After labeling all the images the file is exported as XML, but the needed format of the output pixel file is .txt extension, labelling does not have the preferences to change it. For this XML to .txt process, a small python code script is executed,

which will change the file and make the information according to the required form for our Yolo - v5 model. There are more than 500 images are collected and labelled. Figure 3 shows the example of the Yolo - v5 data .txt file, the python converts images from XML to .txt successfully.

For human violence detection, real action videos are used. Real-time fight videos that have been recorded in CCTV, clips from movies, and small fights held during games and public gatherings are collected from different sources [12]. There are more than 100 videos collected with different formats like mpg, mp4, avi, etc. All these videos are converted to images to train a detection model. The collected videos are separated into two folders as violence and non-violence. To train the models with videos as data first we have to convert the videos into images. OpenCV, a computer vision package that helps to read videos and also convert them into images is used. First, the collected videos are converted into images from each frame of the video. If a video runs around 1 minute, it converts that into 30 images with a count of a timeframe gap, which is saved into separate folders as human-violence and non-human-violence. Like this process all the collected videos are converted into images, there are more than 10,000 images obtained as shown in Figure 4. With these data frames, the model is trained and tested. The converted images are pre-processed; as our model takes input with a dimension of 229x229x3 here the dimensions of the image are resized using the cv2.resize function.

Figure 4. Subplot of image of human-human violence and non-human violence.

3.2. Working

Working in Google colab will give us free GPU. GPU increases the speed of rendering and compiling time. A huge amount of data which is not only in the numerical form will enhance huge amounts of steps to execute, GPU interface produces graphical base processor and evaluation of this is in unit type. Following procedures are followed: Open colab using google account. mount the collab with google drive to produce floating hardware. For the transfer learning part, we cloned the Yolo - v5 model from an open-source website. We made a duplicate file and stored it in the working directory, checking the requirements for extracting features from this model. Import torch packages, which will give data assessing, function for both numerical and graphical representation, this torch function is built to perform in GPU interface. For checking the version of the packages, requirements.txt will print the availability and required version for all the functions. Now go to your data, open a new folder, create a subfolder called images, and label, inside that both the folders add train and Val subfolders. Write a python code with for and if conditions to divide the train and test set for our data. 80 percent of data should be a trained dataset and the remaining 20 percent will be testing our model. A huge number of datasets will yield a high-performance model.

The extension of both .jpg and .txt should have the same name only, after splitting, check the data are placed correctly in the correct folders. Now go to the config file and create model metrics. Certain keywords functions determine the running and building of the model. we have 4 classes for our model, open the config file and add classes like 4, Batch size as 8000, the calculator part of the batch size is classes * 2000, step size has the max size to 90 percent of batch size that means 7200, min size has 80 percent of batch size that is 6400. Go to the backbone layer here to construct the filter size according to the classes and the filter formula is (classes + layers) *construct three layers. The remaining numerical values can be determined by filter values. We are uploading the test files so predication is directly done here, for training the data, constructing a code, and saving it as train.py in the model file. Train code creates a model with hidden layers and activation functions. Here the learning part takes place. We are creating a work base and asking the model how to learn, what kind of features it should learn from the datasets, and pre-trained model features.

Access this tain.py file using !python function and declare the epochs and batch count to it, epochs defines the iteration number how many times the model should learn the process, learning for so much time the model will learn the given information deeply which will be useful for yielding high accuracies. Install the tensor board function to visualize the data to find performance metrics. This Val data goes to the testing part, attached to the predicating measure so we can check the predicated objects. It automatically saves the weights and testing data in the run folder, where we go check the data to check how well it predicated our objects.

The converted images from two separate folders are imported. All these images are appended as a single list and labels for the classification are also mentioned in the list as [‘fight’, ‘no fight’] [13]. The dataset (framed images) is split into train and test data using the train_test_split. The InceptionV3 model is imported and then it is considered as the base model. With the help of base, a new layer of Global Average pooling is added to make the output as 2 label classification, i.e, human violence, not human-violence. The activation function of the model is given as ‘sigmoid’. The model is compiled with the loss as ‘binary _cross entropy’. Accuracy metrics are also obtained. The model is fitted with the train data and test data with verbose=1, epochs=25, and batch size =64. The model resulted in 74 % of accuracy. The trained model is saved in ‘pickle’ format [12].

Now using the pre-trained model, we can predict the new videos. As Inception V3 can only be used for image classification, to classify a video, it should be converted as image frames. Using cv2 the video to be predicted, can be converted into images [14, 15]. The image classification model is slightly modified with the help of cv2.

4. Results and Discussion

An extensive way of developing a technique related to automatic surveillance video detection is to recognize if there is any human violence that has taken place or not. Hence, the best way to identify this is using a deep learning models. It is more important to develop a model to recognize and to detect human violence. The convolutional neural network is used to pre-train the model. Using long short – term memory that uses fully connected layers. Along with CNN that is used to analyze the local motion in the video.

The model is built using the features of yolov5 architecture and custom neural networks, the train, and test data are fitted and the expected results are obtained from it. The obtained results are to be examined now using the classification metrics. The common metrics used are accuracy, confusion matrix, F-score, precision and recall. The confusion matrix with normalization and without normalization helped to define the accuracy of the model. As mentioned earlier each classification model is built and fitted with test values. The final accuracy for the object obtained is 74%.

This Val data goes to the testing part, attached to the predicating measure so we can check the predicated objects. It automatically saves the weights and testing data in the run folder, where we go check the data to check how well it predicated our objects. Figure 5 shows the boundary box and the label classes for prediction test images and prints a person's face with their class name.

Figure 5. Boundary box and the label classes for prediction test images.

Figure 6. Predicted output from the captured video frame.

The model is trained with the Inception - v3 model and also modified video classification is used. With help of a modified classification script and base model, the test video is successfully classified with an accuracy of 74%.

Here the live detection implementation is also applied, that is either from the source camera or web camera the streaming video will be converted into data frames and predicted as shown in Figure 6. The future work of this study has an idea of implementing the database as backend. That is the video to be predicted is uploaded which gives results as human violence or non-human violence, also the object detects are stored into database. Also, an additional feature to be added into the website that includes report generator. If the user wants to generate a report its results the detected objects or faces that have been already present in the database. If it is already in the database, it shows, time and date when the detected images are stored. It will provide a complete report of objects detected, no. of faces detected and whether the detected faces have already existed in the database involved during human-violence or non-human-violence scenario. The limitations of this study are it lags in processing the detection of objects, i.e., it takes time to recognise objects in the video.

5. Conclusion

In this study, a model that helps in human-violence detection and also detection of faces and objects is developed. These two different models were converted as a pickle file which is imported into the local website that has been created using the CSS/HTML as front end. The FAST API web framework helps in combining these deep learning models and implementing them in the website. On this website, there are two buttons provided, one for the human-violence detection model another one for the object & face detection model. Using these, the uploaded video results in either it is human-violence or non-human-violence and another button option is provided which helps to detect the objects, faces in the video.

Acknowledgement

Authors thank the management of Vellore Institute of Technology for providing the computing facilities to complete this research work. The authors thank all peer reviewers for their helpful suggestions on how to enhance the paper's quality.

References

[1] Kaya V, Tuncer S and Baran A 2021, Detection And Classification Of Different Weapon Types Using Deep Learning. *Applied Sciences*, **11** (16), 7535.
[2] Singh P and Pankajakshan V 2018 A Deep Learning Based Technique For Anomaly Detection In Surveillance Videos. Proc. of the 24th National Conf. on Communications, pp. 1-6.
[3] Dandage V, Gautam H, Ghavale A, Mahore R and Sonewar P A 2019 Review Of Violence Detection System Using Deep Learning. Int. Research Journal of Engineering and Technology 6 (12), pp. 1899-1902.
[4] Wang K, Liu M 2022 YOLOv3-MT: A YOLOv3 Using Multi-Target Tracking For Vehicle Visual Detection. Appl. Intell. 52, pp. 2070–2091.
[5] Antoniou A and Angelov P 2016 A General Purpose Intelligent Surveillance System For Mobile Devices Using Deep Learning. Proc. of the Int. Joint Conf. on Neural Networks, pp. 2879-2886.
[6] Savran A 2007 Multifeedback-Layer Neural Network. IEEE Transactions on Neural Networks, 18 (2), pp. 373-384.
[7] Bilinski P and Bremond F 2016 Human Violence Recognition And Detection In Surveillance Videos. Proc. of the 13th IEEE Int. Conf. on Advanced Video and Signal Based Surveillance, pp. 30-36.
[8] Fu E Y, Leong H V, Nga G and Chan S 2016 Automatic Fight Detection In Surveillance Videos. Proc. of the 4th Int. Conf. on Advances in Mobile Computing and Multimedia, pp. 225-234.
[9] Manoharan S 2019, Image Detection Classification And Recognition For Leak Detection In Automobiles. Journal of Innovative Image Processing, 01 (02), pp. 61–70.
[10] Kim J H, Song J H and Lim D H 2020. CT Image Denoising Using Inception Model. *Journal of the Korean Data And Information Science Society,* **31** (3), pp. 487–501.
[11] Bhargav P, Sree Lakshmi Keerthi B S L, Charitha K, Sarath B and Pratap A R 2020 Face Clustering On Image Repository Using Convolutional Neural Network. Int. Journal of Psychosocial Rehabilitation, 24 (5), pp. 5104–11.
[12] Fauzi F, Szulczyk K and Basyith A 2018 Moving In The Right Direction To Fight Financial Crime: Prevention And Detection. Journal of Financial Crime, 25 (2), pp. 362–368.
[13] Du S, Zhang B, Zhang P, Xiang P and Du H 2021, FA-YOLO: An Improved YOLO Model For Infrared Occlusion Object Detection Under Confusing Background. Wireless Communications and Mobile Computing, 1896029.
[14] Sharma J, Giri C, Granmo O C and Goodwin M 2019 Multi-Layer Intrusion Detection System With Extratrees Feature Selection, Extreme Learning Machine Ensemble, And Softmax Aggregation. EURASIP Journal on Information Security, 15.

[15] Xu B 2021 Improved Convolutional Neural Network in Remote Sensing Image Classification. Neural Computing and Applications, 33, pp. 8169–80.