Grenze International Journal of Engineering and Technology, June Issue

ATM Surveillance System using Machine Learning

Swapnil Mandavilli[1], Abhinav Pallayil[2], Siddhant Evre[3], Prof. Anand Mane[4] and Prof. Manish Parmar[5]
[1-5]Sardar Patel Institute of Technology, EXTC, Mumbai, India
Email: swapnil.mandavilli@spit.ac.in, {abhinav.pallayil, siddhant.evre, anand_mane, manish_parmar}@spit.ac.in

***Abstract*—Due to use of outdated technology and lack of basic security features ATM machines are becoming prime targets for thieves and hackers. Nowadays there are a lot of crimes occurring in the ATM sector. Thieves have come up with many different ways to trick ATMs, manipulating and forcing people into spitting out large amounts of cash. So we have come to the idea of making this project from our observations of life incidents happening in the world. This project ATM Security using Machine Learning provides some security measures for ATM transactions. This project is used to detect masks, helmets, person overstays, person tracking, face detection, and animal detection. Here, OpenCV and TensorFlow were used as platforms, Python language was used for deep learning techniques, and Keras API was used for object detection. Email sends alerts to nearby banks and police stations.**

***Index Terms*— ATM security, Object detection, CNN, Tensorflow, Machine learning.**

I. INTRODUCTION

From the beginning of a transaction to its conclusion, this anti-theft system guarantees cardholders a secure environment. It keeps in touch with all pertinent national and international security working groups dedicated to the investigation and prosecution of crimes committed against ATMs directly or indirectly through crimes committed at other terminals. Banks increase their ATM usage for two reasons. First to guarantee that clients can access their money at any time, wherever. Secondly, lower their own operating costs associated with sending staff to serve clients. Every day, billions of people utilize ATMs in their daily lives. Therefore, studies and investigations are ongoing to enhance the security of ATM transactions.

As there are more and more crimes involving ATMs, including robberies, break-ins, and hacking of ATM passwords, technology must be developed to combat this, and methodologies must be improved. In order to secure ATM transactions, banks must use greater caution. Present-day technology includes card readers, digital pens, and video cameras. This system provides security for ATMs using video cameras considering various scenarios like mask detection, helmet detection, stray animals detection, etc. We have developed this system using python and deep learning concepts.

A camera that provides images of at least selected portions of an automated teller machine (ATM) is included in a security system useful for monitoring an ATM. Automatically a controller determines whether a difference between a reference and a later acquired image from the camera of the ATM indicates an ATM change. One example detects whether a skimming reader has been placed next to a card receiving slot. Acquiring many reference images corresponding to different lighting conditions and selecting an appropriate one of the reference images based on a lighting condition or time of day associated with a later acquired image.

Grenze ID: 01.GIJET.9.2.48_1

II. LITERATURE SURVEY

ATM Security using Machine Learning: Facial detection & emotion recognition features have been done using various algorithms like K-Nearest Neighbours Algorithm, and Back Propagation Neural Networks. The machine will only work if the expressions and emotions are normal and there is no sign of forced usage or any other illicit. Accuracy that can be improved. Multiple object detection is missing.
The time span of each object is not defined [1].
ATM crime detection using image processing integrated video surveillance: A systematic review. Video surveillance system integrated with image pre-processing, moving object detection, face detection, facial component detection, object shape-appearance detection, and decision-making. Time complexity is very high. Unnecessary object detection is included. [2]
An Intelligent Vision System for monitoring Security and Surveillance of ATMs: Pre-processing techniques are used to eliminate the noise components and decrease false detections. A Bounding Box is formed around the detected object used for face detection. The model does not support real-time analysis. Multiple object detection is missing. [3]
Face Recognition by Coarse-to-Fine Landmark Regression with Application to ATM Surveillance: For face recognition, trained CNN to obtain a compact representation, where the rectified linear unit (ReLU) is replaced by max-feature-map (MFM). The model does not support real-time analysis. Multiple object detection is missing. [4]
A New Video Surveillance System Employing Occluded Face Detection: Detection of the partially occluded face, which can be used in the characterization of suspicious ATM users, is then performed based on Support Vector Machine (SVM) method. PCA is to find a low-dimensional subspace time complexity that y is very high. The model does not support real-time analysis. [5]
Automatic video surveillance for theft detection in ATM machines: An enhanced approach. Used unsupervised learning techniques like pose clustering and consistency in order to train the system. The captured video is fragmented into smaller frames and then the vector graphics and image processing techniques are implemented. Accuracy can be improved Multiple object detection is missing. The time span of each object is not defined. [6]
Using Attack-Défense Trees to Analyse Threats and Countermeasures in an ATM: A Case Study. Captured the most dangerous multi-stage attack scenarios applicable to ATM structures, and establish a practical experience report, where there's a reflection on the process of modelling ATM threats via an attack-defence tree. The algorithm isn't clear in the implementation which will reduce the accuracy. Attacks that are not there in the pre-defined attacks won't send an alert. [7]
Real-time security framework for detecting abnormal events at ATM: In the testing phase, the query class is matched by each individual point using a random forest model and generates an individual voting score for each class. The random forest approach enables an efficient matching for the larger. Doesn't contain functionality such as object selection, structural variation, and audio-based recognition [8].
Abnormal Event Detection Method for ATM Video and Its Application: Abnormal event detection approach based on intelligent video content analysis and applies to ATM video forensics analysis. Based on the advantage combined the weighted geometric characterization with the PCA method, and discusses a new algorithm, namely an improved PCA algorithm dataset. Multiple object detection is missing. The time span of each object is not defined. [9]
Unusual Event Detection in Low-Resolution Video for enhancing ATM security: After pre-processing background modelling is used to create an ideal background (static or dynamic) that may include image subtraction operations. Model-based, Region-based, Contour based, and Feature-based algorithms are used for object detection and track Ing. Focuses only on low-resolution video detection and don't include test cases that are related to our country. [10]

III. METHODOLOGY

In this section, we will discuss our proposed model for ATM Surveillance System Using Machine Learning.The object detection algorithm is coded in Python and the methodology for the same is mentioned below.
The first step in any detection is to collect a dataset of aparticular object that has to be detected. This dataset consists of positive and negative images. The positive images are the images that contain the object to be detected in various angles, colors, shapes, and sizes.The negative images are the images that do not contain the object to be detected at all. Both the positive and negative images dataset are huge in number. More the number of images the better the accuracy of the model. The second step is to train the model using the dataset.

This model uses classification techniques to classify positive and negative images. For the training of the model, all the images should be of the same size so all the images are of the size 224x224.
The next step in training the model is to create two arrays (data and label). Then this data is split by the train-test split method and this data is used for classification techniques. The training part of the models is done and the trained models are ready to detect their respective objects. The next step is to do live tracking using the camera installed in the ATM (Automated Teller Machine). For live tracking, we use OpenCV in Python. The video that is being detectedis converted into image frames and these image frames are used for object detection by trained models. Now each image frame is converted into the size 224x224 which is the same size as that of the trained model dataset. This concludes the framing of the image and image normalization. If any moving object enters the frame of the camera the moving object and their respective movements are tracked. This tracking is donewith the help of a centroid tracker which marks the centroid of the moving object and tracks its movements according to the path of the respective centroid.
A bounding box is created around the moving object according to the coordinates given by the centroid tracker. This bounding box helps us in various detection such as multiple people detection, and person overstay detection. For this bounding box to be detected we convert the image into a NumPy 2-D array and this array gives us the coordinate of each pixel. With the help of these coordinates, the person and animal movements are tracked. A bounding box changes shape according to the movements of animals or people. For example, if a person spreads his hands the width of the bounding box will increase.
This particular project is used to detect the following:

- Mask Detection
- Helmet Detection
- Person Overstay
- Person Tracking
- Animal Detection

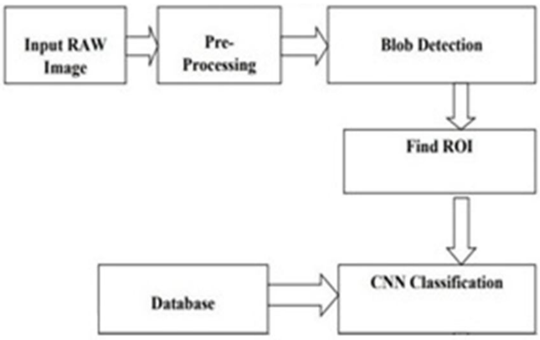

Figure 1. Flowchart

A. Mask Detection

This detection is in particular with face detection so another bounding box is drawn around the face, this bounding box extracts the features which help in detecting if the person is wearing the mask. The trained model mentioned above is used along with this face tracker and the classification model helps us to classify whether the person is wearing a mask or not. In ambiguous situations where the software is not sure of the person wearing a mask, the confidence parameter comes into the picture. The confidence parameter value is between 0 and 1. If the confidence parameter has a value 0.7 it means that the model will show that person is wearing a mask if it is more than seventy percent sure.

B. Mask Detection

Helmet Detection is in particular with face detection so another bounding box is drawn around the face, this bounding box extracts the features which help in detecting if the person is wearing the helmet. The trained model mentioned above is used along with this face tracker and the classification model helps us to classify whether the person is wearing a helmet or not. In ambiguous situations where the software is not sure of the person wearing a helmet, the confidence parameter comes into the picture. The confidence parameter value is between 0 and 1. If the confidence parameter has a value of 0.7 it means that the model will show that person is wearing a helmet if it is more than seventy percent sure.

C. Person Overstay

To calculate the time the person is spending in the ATM we have used datetime library of Python datetime.datetime.now() is used to calculate the current time. A variable old time is assigned to store the time the object had entered the ATM. The time spent in the ATM is the difference between the current time and the old time. An alert will be generated if the person stays for more than ten minutes.

D. Person Tracking

The main goal of centroid tracker algorithms is to monitor coordinates by specifying some threshold values to be considered as the same spots. There is no use of machine learning in it. We determine the Euclidean distance of the centroids of all detected objects in following frames using a multi-step approach.

E. Animal Detection

It consists of Raw input image section, RGB to Gray scale conversion, Noise Removal, Image Enhancement, Find ROI and animal Extracted and CNN classification section. Raw input image is given as input to RGB to Grayscale conversion section and this section convert color image into grayscale image which is understand by machine tool. Gray scale image noises are removed at noise remover section. Blob techniques used to detect animal from photos or video. CNN classification section finds ROI of the image and extract the animal part. Extracted image is compared with database to classify animal and display the result.

IV. Implementation Details

Image processing: Image processing is the process of recognizing real-world objects such as bottles, helmets, and cars. Image processing is also used to detect faces, eyes, mouths, etc. Image processing is based on image, video, and live stream data. This image-processing application is widely used in security, surveying, and other fields.

TensorFlow: TensorFlow is Google's open-source machine learning framework for data flow programmingfor a variety of tasks. The nodes of the graph represent mathematical operations, and the edges of the graph represent multidimensional arrays of data (tensors) passed between them. A tensor is a simple multidimensional array that extends a two-dimensional table to an extended dimension of data. TensorFlow has many features that make it suitable for deep learning. This project uses Tensor Flow to implement object detection.

Object detection: Every single Object Detection Algorithm has a different way of operating, but they all work on the same outlook. They extract features from the input images at indicators and use these features to regulate the class of the image. Be it through Open-CV or Deep Learning. Object detection has been used to detect faces, people, animals, helmets, and masks.

Keras: Keras for the Tensor Flow platform. Keras is a high-level training and building of deep learningmodels. Keras was designed for people, not machines. We put the user experience at the center. Keras follows best practices to reduce cognitive load, minimize the number of user actions required for common use cases, and provide clear, actionable feedback on user errors.

Data processing: Training to detect objects requires images. The images must be the same size, and images of the same size are given to the training algorithm as input images. The training algorithm holds images. As with all deep learning techniques, we split the image into many parts and process each part of the image. After processing each image part, the processed image part is converted to a variable and saved as a .model file. The stored variables were given a different value for each variable to distinguish them from each other. Variable values are saved in the form of .pickle format files. Variable values contain the features of the trained image.

OpenCV: OpenCV (Open Source Computer Vision Library) is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms.

cv2.dnn.blobFromImage: A blob is just a (potential collection) of images that have the same spatialdimensions (i.e. width and height), and the same depth (number of channels), all preprocessed in the same way.he cv2.dnn.blobFromImage andcv2.dnn.blobFromImages functions are nearly identical.

blob = cv2.dnn.blobFromImage(image, scalefactor=1.0,size, mean, swapRB=True)

Image: the input image to preprocess before passing it to the deep neural network for classification

Scale factor: After doing the mean subtraction you can scale the image by a factor if you want. This value defaults to "1.0" (meaning no scaling), but you can specify a different value. It's also important to note that the scale factor should be 1/sigma since we are actually multiplying the input channel (after mean subtraction) by the scale factor.

Size: Here you specify the spatial size expected by theconvolutional neural network. For most modern neural networks, this is either 224x224, 227x227, or 299x299.
Mean: These are mean subtracted values. These may be3-tuple RGB averages or single values, in which case the specified value is subtracted from each channel ofthe image. Always specify 3-tuples in the order "(R, G, B)" when doing average subtraction, especially whenusing the default swapRB=True behavior.
swapRB: OpenCV assumes the image is in BGR channel order. However, "average" assumes you are using RGB ordering. To resolve this discrepancy, setthis value to True to swap the R and B channels of the image. By default, OpenCV does this channel switching.
The cv2.dnn.blobFromImage function returns a blob which is our input image after mean subtraction, normalizing, and channel swapping.

V. RESULTS

The image below is the screenshot showing the output fora person wearing a mask which the model has predicted and an alert is generated. A bounding box is drawn around the face, this bounding box extracts the features which help in detecting if the person is wearing the mask.The accuracy of mask detection is: 95 percent.

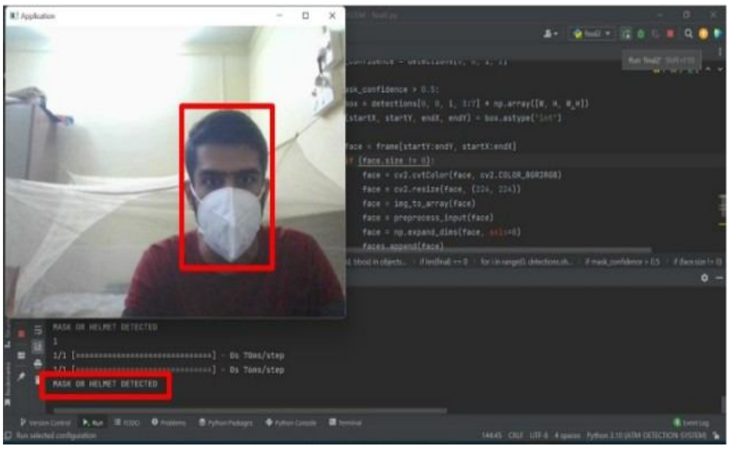

Figure 2. Mask Detection

The image below is the screenshot showing the output for a person wearing a helmet which the model has predicted and an alert is generated. A bounding box is drawn around the face, this bounding box extracts the features which help in detecting if the person is wearing a helmet.The accuracy of helmet detection is 92 percent.

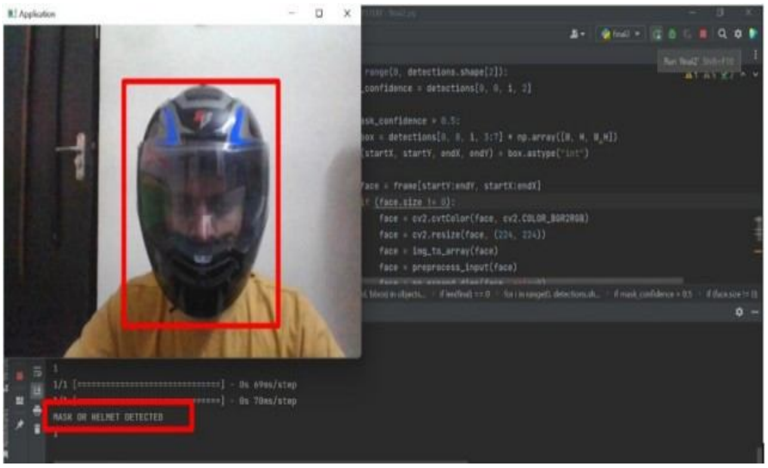

Figure 3. Helmet Detection

The image below is the screenshot showing that the person has stayed in the ATM for more than 10 minutes and alert has been raised for the same. The accuracy of person overstay detection is: 99 percent.

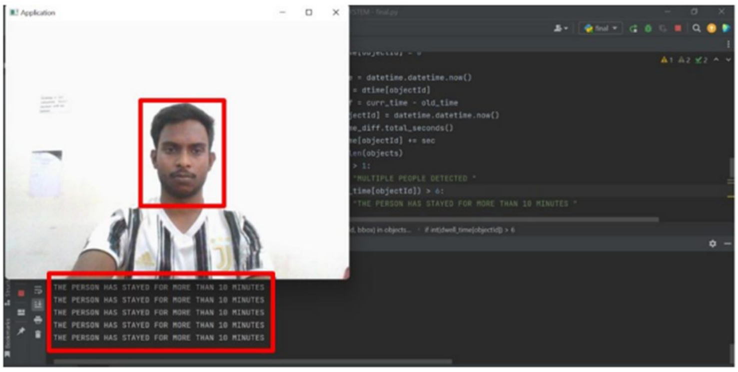

Figure 4. Person Overstay Detection

The image below is the screenshot showing the detection of dog and alert has been raised for the same.The accuracy of animal detection is : 97 percent.

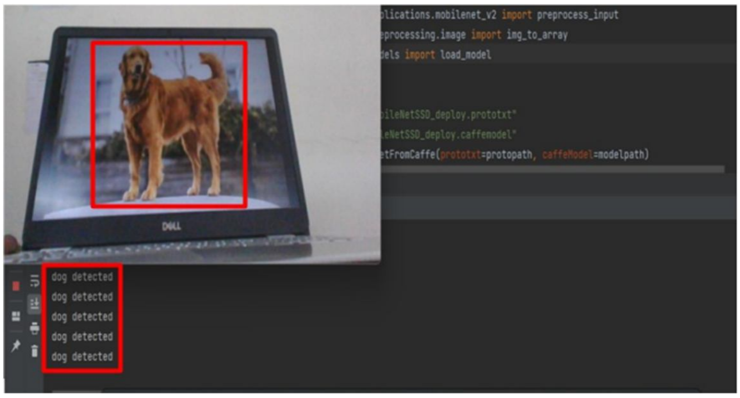

Figure 4. Animal Detection

VI. CONCLUSIONS AND FUTURE SCOPE

In this proposed work, the ATM surveillance system is partially developed and can be implemented in a real-timeenvironment. The system will provide better performance in the experimental setup as the lighting is better and the cameras used in ATMs are far better than cameras in laptops. We developed the software for the ATM surveillance system and all the detection test cases as part of this project. The future scope is the part which we will be focusing on in the coming months.

In the next phase of the project, we will be adding new features that can be detected such as multiple person detection, so that if at any time in the ATM if more than one person enters or is present in the ATM it will detect and generate an alert and it will be added advantageover our existing model. We will decrease the run time by increasing the speed of the program by using more efficient algorithms. We will be putting all the detection in one program file, that is using only one camera, and will ensure that everything runs smoothly and nothing gets disturbed due to the collaboration. We will be improving the accuracy of our model. We will be implementing an alert system that will send an alert to the security agency in case of any abnormalities. This alert system will be done through messages and emails directly from the program with zero human intervention.

ACKNOWLEDGMENT

We would like to thank Prof. Dr. Anand Mane and Prof. Manish Parmar for their guidance, assistance and tips while preparing this technical paper.

REFERENCES

[1] International Research Journal of Engineering and Technology (IRJET)" ATM Security using Machine Learning" Volume: 06 Issue: 05 May 2019.

[2] https://crossmark.crossref.org/dialog/?doi=10.35940/ijrte.D9567.118419domain=www.ijrte.org.

[3] https://fdocuments.in/document/atm-security-5584542 c6c5cd.html?page=17
[4] P Liu, K. Jia, Z. Wang and Z. Lv, "A New and Effective Image Retrieval Method Based on Combined Features", Proc. IEEE Int. Conf. on Image and Graphics,vol. I, pp. 786-790, August 2007
[5] Scott, M.S.: Robbery at automated teller machines. US Department of Justice, Office of Community Oriented Policing Services (2001).
[6] Sharma, N.: Analysis of different vulnerabilities in auto teller machine transactions. J. Global Res. Comp. Sci. 3(3), 38–40 (2012)
[7] Ahad, M.A.R., Tan, J.K., Kim, H., Ishikawa, S.: Motion history image: its variants and applications. Mach. Vision Appl. 23(2), 255–281 (2012)
[8] Davis, J.W., Bobick, A.F.: The representation and recognition of human movement using temporal templates, pp. 928–934. In Computer Vision and PatternRecognition, IEEE Proceedings (1997)
[9] Bobick, A. F., Davis, J. W.: The recognition of human movement using temporal templates: IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 23, no. 3,pp. 257–267 (2001)
[10] Poppe, R.: A survey on vision based human action recognition. Image Vis. Comput. 28(6), 976–990 (2010)
[11] Debard, G., Karsmakers, P., Deschod, M., Vlaeyen, E., Bergh, J.V.D., Dejaeger, E.: Camera based fall detection using multiple features validated with real life video. Intell. Environ. Workshops 10, 441–450 (2011)
[12] Mahbub, U., Imtiaz, H., Ahad, M. A. R.: Action recognition based on statistical analysis from clustered flow vectors. Signal, Image and Video Processing, 1–11(2013)
[13] Gong, W., Gonza'lez, J., Roca, F. X.: Human action recognition based on estimated weak poses. *EURASIP Journal on Advances in Signal Processing*, 1–14 (2012)
[14] Paul, M., Haque, S. M., Chakraborty, S.: Human detection in surveillance videos and its applications-areview. *EURASIP J. Adv. Signal Proc*., 1–16 (2013)
[15] Kim, W., Lee, J., Kim, M., Oh, D., Kim, C.: Human action recognition using ordinal measure of accumulated motion. EURASIP J. Adv. Signal Proc., (2010)
[16] Vaishali, Shilpi Singh: Real-Time Object Detection System using Caffe Model *International Research Journal of Engineering and Technology (IRJET)* Volume: 06 Issue: 05 May 2019.
[17] Kruthi H , Nisarga A C, Supriya D K, Sanjay C R, Mr. Mohan Kumar K S : Animal Detection Using Deep Learning Algorithm , Vol. 10, Issue 6, June 2021.
[18] https://davidmatablog.wordpress.com/2017/12/05/real-time-object-recognition-with-opencv-python-deep-learning-caffe-model/
[19] https://medium.com/acm-juit/ssd-object-detection-in- real-time-deeplearning-and-caffe-f41e40eea968