Title IOT application using XXX to recognize Still images and loop videos in real-time video conferences

Authors

Abstract

Keyword

Introduction

In this era of remote work and internet communication, video conferencing has gained appeal as a means of linking businesses, organizations, and individuals[1]. With the rise of virtual backgrounds and pre-recorded recordings, it's harder to tell the difference between live video feeds and images that don't change or loop. In order to use real-time video conferencing, the study need to know the difference between still images and looping videos[2][3]. It first deters individuals from telling lies. Rarely, participants in a video conference may substitute their actual video feed with a still image or looping video to give the impression that they are there and participating in the conversation. If the person is absent from the conference, they may be unable to respond to questions or speak. Second, the ability to recognize still images and replayed videos can aid in the prevention of security breaches. It is possible to communicate private information during a video conference that should only be visible to the parties involved. Someone with access to a still image or looping video could gain unauthorized access to this data[4][5]. Lastly, being able to tell the difference between still photos and videos that loop can help keep the meeting's integrity. People who don't take part in the conversation may miss important information or get the wrong idea of what is going on[6]. This can lead to misunderstandings and ambiguity, which will ultimately hinder the meeting's outcome. In order to address these challenges, numerous videoconferencing providers have included technology that enables users to recognize still photos and looping videos in real time. These capabilities use machine learning to examine the video feed for trends or abnormalities that may indicate the use of a static image or loop video. Real-time image recognition and video looping Video conferencing is necessary to make sure that everyone is there and paying attention, to prevent security breaches, and to keep the session's integrity[7][8]. As long as video conferencing is an integral part of remote work and communication, users must be aware of these challenges and take the appropriate precautions to ensure that their genuine video feed is being utilized during video conferences.

In recent years, the connection between IoT and machine learning has emerged as an important topic of research. IoT provides a tremendous amount of data that may be utilized to improve choices and automate processes due to the vast number of linked devices and sensors. These algorithms enable real-time analysis of this data and the identification of trends that can be used to improve procedures and outcomes. So, researchers are looking into how IoT and machine learning could be used in many different fields, such as manufacturing, transportation, agriculture, and healthcare. In the healthcare industry, for instance, IoT devices can collect patient data that can be utilized in conjunction with machine learning algorithms to diagnose and treat health problems[9][10]. Using Internet of Things sensors can improve traffic flow, while machine learning algorithms can analyze traffic patterns to predict and avoid accidents. Due to the many possible benefits of this convergence, academics are looking for new ways to improve output and results in many fields[11][12].

In computer vision, the Viola-Jones method is a well-known technique for detecting objects. It is commonly employed to identify faces in pictures and films. The method looks for patterns in an image by using Haar-like features, which are simple rectangles that can be quickly calculated. Many classifiers are then used to determine whether or not an object is present in a photograph. This approach allows for the processing of video feeds in real time[13][14]. The Viola-Jones approach has further applications outside face recognition. It has also been utilized for gesture recognition, object tracking, and the identification of pedestrians. This work combines the Viola-Jones technique, along with the Internet of Things and machine learning, to identify static images and looping videos in real-time video conferencing. The following information is necessary for the next steps of the study: The second section of this essay expands on the literature review. In Part 3 of the study, the methodology of the proposed system is looked at, and in Part 4, the study' result analysis are looked at. Section 5 gives a conclusion and some future recommendations.

Literature review

A researcher is implementing a facial recognition system for school attendance. The system consists of two components: verification and recognition. Students receive an NFC tag with a unique ID number upon enrollment, and the tag's built-in camera records their faces. The data is subsequently transmitted to a server at the college for verification. The obsolete system administration software is converted into a portable Python module[15][16]. Face-to-face photographs can be compared in a 1:1 or 1:N matching method to determine the identity of a pupil. Few researchers also examine how event sensors, or dynamic vision sensors, can be used to identify and track facial features as well as examine how individuals blink. The suggested method uses a new algorithm that looks for blinks in event space to find faces and eyes[17][18]. Convolutional neural networks (CNNs) can be taught to operate in event space without intermediate intensity representations using this method. Real-world cameras have been used to show that face and eye tracking and blink detection work well. The supplementary materials include a movie demonstrating how the suggested approach operates. Few researchers also develop vision-based automatic face-recognition systems. The four parts of these systems are face detection, picture pre-processing, feature extraction, and matching. Feature extraction methods might be holistic, feature-based, or a combination of the two. Landmark detection is needed to construct a geometric representation of a face, and it can also be used to find faces[19][20]. The project looked at different ways to find facial landmarks and made suggestions for tools that would make it easier to add other face datasets. A geometrical model of the face was created based on the distance between the eyes to estimate the location of other landmarks for face segmentation. Several researchers have also conceived of a facial detection and recognition system that uses a robust Python algorithm to locate and identify faces rapidly and precisely. A double layer of WISARD in neural networks is used to validate a person's face. This provides 87% accuracy when comparing a face to a database. The system uses the dlib and facial recognition libraries, and the graphical user interface (GUI) enables users to interact with the system and provide input. Overall, the method suggested is a quick and accurate way to find and recognize faces[21]. Few researchers also demonstrate a vision-based technique for identifying voluntary eyelid closures. People with disabilities use this system as a human-computer interface. The eye-blink detection algorithm is made up of four main steps: finding the face, getting the eye region, detecting the eye-blink, and classifying the eye-blink[22]. The algorithm's most important step is face detection, which comes after identifying the eyes in the image. The eyes are then followed using a normalized cross-correlation approach, and the change in correlation coefficient over time is examined to identify voluntary eye blinks lasting more than 250 ms. The technique is used to create a user interface that can automatically detect when a person blinks their eyes willingly. According to what a small number of researchers have discovered, image processing is a type of signal processing where an image serves as the input and an image or set of image characteristics serves as the output. Two types of picture filtering exist. In fundamental image transformations, image data is altered using elementary mathematical processes. Object tracking is the process of finding one or more objects in a group of pictures. It is an important part of computer vision and is used for many things, like surveillance and recognizing objects[23]. Several researchers also show a framework for detecting objects that is made up of new methods and ideas. The framework contains three major components. The first thing that this study adds is the integral image, which is a new way to describe an image that makes it easier to quickly judge its qualities. The second contribution is a method for creating a classifier with AdaBoost by selecting the most important features. The third contribution is a way to combine classifiers that get more and more complicated to speed up the detector by focusing on promising areas of the image. This framework provides an efficient and robust method for locating things. Moreover, a small quantity of data indicates In biometric systems, face recognition identifies individuals based on facial characteristics. There are three primary methods for recognizing faces: based on features, based on the entire individual, and a combination of the two. The nose and eyes are two local features that are used by the face detection system to gather information. This strategy is known as "feature-based." The holistic technique examines the entire face to locate and identify faces. The hybrid method is a combination of the feature-based method and the whole-face method. Local and whole-face input data are used by the face detection system. Fewer researchers also do work that gives computer vision operations a common framework and speeds up how machine perception is used in commercial goods. Blink develops tools that simplify the configuration of commercial IoT devices for end users. During image processing, an image is converted to a digital format, and operations are performed on it to enhance it or extract relevant data. Computer vision syndrome (CVS), which causes people to blink less frequently when viewing bright objects up close, has become more prevalent as the number of computer users has increased. The majority of methods for detecting eye blinks involve three steps: locating the face, the eyes, and the blink. In image processing, the normal flow approach is employed to estimate the movement of the eyelids, whereas Gabor filters are used to locate edges. The distance between an individual's upper and lower eyelids is used to determine when they blink[24][25].

Methodology

Figure 1 shows the proposed methodology diagram.

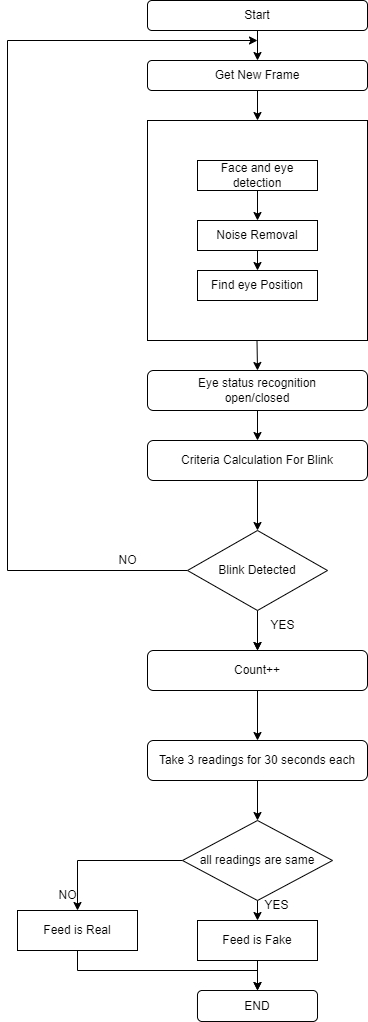


Figure 1: Flowchart of the study

The study will initiate the loading procedure for the live video broadcast to get things moving. Currently, the camera is transmitting live video, which will be loaded. This is accomplished using OpenCV's Video Capture function. This function allows us to record footage from the camera and play it back in real time. The frame rate and resolution of the camera can be altered to meet the requirements of the investigation. Therefore, to convert an image to grayscale, the study must limit the quantity of color information and display the image in various shades of grey. After the image has been saved, this action is taken. The amount of color saturation in each pixel used to create the grayscale image determines the brightness of each pixel. The well-known Haar cascade classifier can be used to identify objects in an image or video stream. This approach can also be used to recognize text. After completing this phase, it is necessary to employ the Haar Neural Network. The algorithm examines the image's "Haar features" and compares them to a set of trained models to determine whether or not an object of interest is there. This allows it to determine whether or not the object of interest is present in the image. In Table 1 shows the pseudocode of Haar Cascade Classifier

Table : Pseudocode for Haar Cascade Classifier

1. Pick (maximum acceptable false positive rate per layer) and (minimum acceptable detection rate per layer)
2. Lets is target overall false positive rate
3. Lets is a set of positive examples
4. Lets is a set of negative examples
5. Lets , Do , and ( : overall false positive rate at layer 0 , Do: acceptable detection rate at layer 0 , and is the current layer )
6. While : overall false positive rate at layer :
   1. i++ (layer increasing by 1)
   2. ( : negative example :
   3. While :
      1. (check a next negative example)
      2. Use and to train with AdaBoost to make a xml (classifier)
      3. Check the result of new classifier for and D0
      4. Decrease threshold for new classifier to adjust detection rate
7. empty
8. If use the current classifier and false detection to set

, and Table 2 shows the pseudocode of Viola Jones Algorithm.

­

Table 2 : Pseudocode for viola jones face detection algorithm

­

1: Input : Original test image

2: Output :image with face indictors as rectangles

3: For I 🡨 1 to num of scales in pyramid of images do

4: Downsample image to create image1

5: Compute integral image , image2

6: for j 🡨1 to num of shift steps of sub-window do

7: for k 🡨1 to num of stages in cascade classifier do

8: for l 🡨1 to num of filters of stage k do

9: Filter detection sub-window

10: Accumulate filter outputs

11: end for

12: if accumulation fails per-stage threshold then

13: Reject sub-window as face

14: Break this k for loop

15: end if

16: end for

17: if sub-window passed all per-stage checks then

18: Accept this sub-window as a face

19: end if

20: end for

21: end for

The Haar cascade classifier is implemented in multiple phases. These steps include choosing a pre-trained model that works with the object to be found, preparing the image to be found, and using a classifier on the image to find interesting parts. Choosing a model that has been previously trained and is compatible with the object to be located is one of the most crucial steps. The speed with which the Haar cascade classifier operates is one of its outstanding features. Even devices with low processing power may identify things in real time as a result. It can also be trained to recognize a broad variety of objects, making it a versatile tool that can be implemented in a variety of computer vision applications. This is because it can be trained to recognize a vast array of objects. The Haar Cascade Classifier is often used to identify faces, people, and objects, in addition to its many other uses. Anyone who works in the field of computer vision would benefit a lot from using this technology because it is so accurate and useful. After completing the stages for the classifier, the study must submit the images using the Viola Jones algorithm, a well-known and efficient method for locating objects in computer vision. Even though identifying faces in images or videos is by far its most prevalent application, it may also be used to identify other objects. While the Viola Jones method processes an image, the first step is to prepare the image for detection by converting it to grayscale and standardizing its dimensions. This is completed before the images are loaded into the algorithm. This step is performed before the images are loaded into the algorithm. Then, Haar-like features, which are rectangular parts of an image with a certain pattern of intensity values, are used to pull out characteristics from each image. This is how a category is assigned to the image. These characteristics are utilized in the image classification process. In the end, the suggested model will be able to recognize faces and tell them apart based on their features.

These are just a few of the many ways that a company's overall success can be judged. The Equation 1 measurement, wherein the variable MSE is represented by the Mean Squared Error -

Equation 1

where

* is the ith observed value.
* is the corresponding predicted value.
* N = the number of observations.

By using the formula MSE value obtained is – 0.0205

Precision – it the measure of the positive predictions made by the model were correct. Which is represented by Equation 2

Equation 2

The value of precision score obtained is 1.0

Recall – it measures that how many of the positive class samples present in the dataset were correctly identified by the model. Which is represented by Equation 3

Equation 3

The value of recall is obtained is 0.994

F1 score – the equation is represented by Equation 4

Equation

Score

The f1 score obtained is 0.997

The Adaboost algorithms uses the formula –

The first predictor is trained on the weighted error rate r1 is computed on the training set using Equation 5

Equation 5

The more accurate the predictor is the higher its weight will be using Equation 6.

Equation 6

Next instance wright is provided below using the formula using Equation 7–

Equation 7

Then all the predictors are normalized using the formula using Equation 8

Equation 8

Results

In the section of the report that is about data analysis, the main goal is to look at possible outcomes using graphs and numbers. The basic premise of this study is that there is a distinction between the upper and lower eyelids when the iris of an eye is observed. The marks p1, p2, p3, and p4 mark the upper eyelid, while the points p1, p6, p5, and p4 mark the lower eyelid. When the eyelids are closed, there is no gap between the eyes, and p2, p6, and p3, p5 overlap. This enabled the observation of a blink. Figure 2 depicts how a blink is captured.

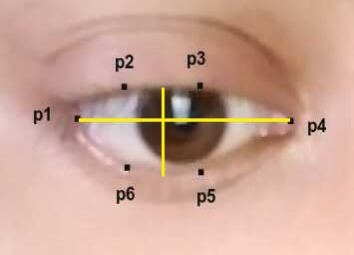
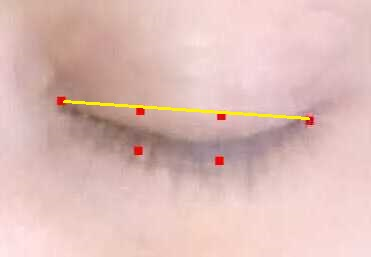
 

Figure 2 : Blink Capture

The suggested model calculates the F1 score, MSE score, and precision score based on the blinking value. Recall the hour of execution and other details. Precision is the model's capacity to correctly recognize positive samples. Because the accuracy score in this instance is 1, all positive samples are present. This is a favorable outcome since it demonstrates that the model correctly interprets all negative input as positive. Notably, a model may not always have an accuracy score of 1, as it may ignore all positive examples. In contrast, the recall score quantifies the model's capacity to locate all positive samples. In this instance, the recall is 0.994%, which indicates that the model correctly detects positive samples 99.4% of the time. This is a beneficial effect because it demonstrates that the model lacks an abundance of success stories. Note that a recall score of 0.994% indicates that the model lacks some positive examples. Depending on the situation and what will happen if positive samples are lost, this may or may not be okay. The F1 score is the harmonic mean of the accuracy and recall of the model. It provides a single measure of the model's accuracy that includes both precision and recall. In this case, the F1 score is 0.997, which means that the model reliably and accurately finds the input. This result shows that the analysis is correct, which is in line with the fact that it got high marks for both accuracy and recall. MSE stands for mean square error, which measures the difference between expectations and reality. In this case, the MSE is 0.0205, which means that the model accurately predicted the real values. This demonstrates how accurate the model is at predicting regressions. Based on the results, it appears that the model works well and matches the data well. However, keep in mind that the performance of the model is only as good as the data used to train and evaluate it. If the data are skewed or noisy, the results may not accurately reflect the performance of the model. It is important to test how well the model works with a separate set of data to make sure it will work well with new data. It is also essential to analyze the application's potential false-positive and false-negative consequences. Sometimes false positive results are worse than false negative results; however, this is not always the case. Thus, it is essential to analyze how the tradeoffs between recall and precision affect the particular application. In addition, a model evaluation for training on the Haarcascade classifier was conducted. The value as a function of the number of faces is displayed in Table 3.

Table 3: Time consumed for the number of faces

|  |  |  |  |
| --- | --- | --- | --- |
| **Training on Haarcascade classifier** | | | |
| **No of Faces** | **Time of execution** | **No of Faces detected** | **Accuracy(%)** |
| **5** | **0.3** | **5** | **100** |
| **10** | **0.8** | **7** | **70** |
| **15** | **0.7** | **14** | **93** |
| **20** | **0.4** | **15** | **75** |
| **56** | **56** | **56** | **56** |

The accuracy graph in Figure 3 displays the link between time and model iteration;

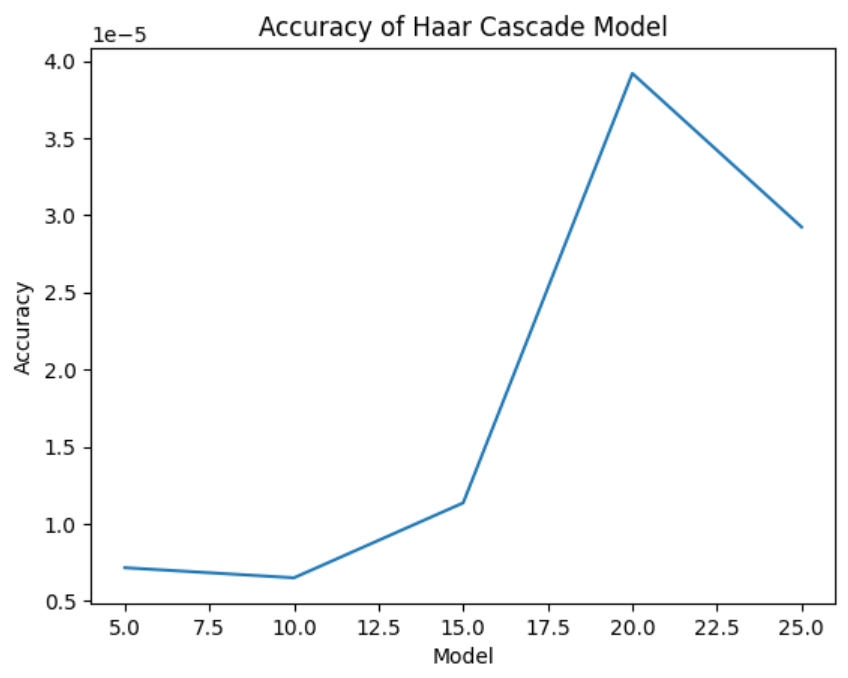


Figure 3 : Accuracy Graph

Figure 4 depicts the time required to compute the findings.

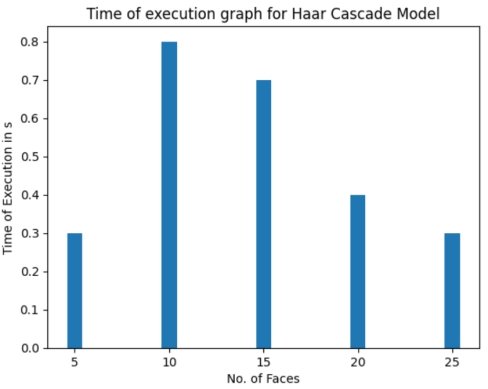


Figure 4 : Time Consumed for number of faces

Conclusion

Participants in a real-time video conference need to be able to tell the difference between still images and looping videos so they don't feel left out. As more people work from home and attend meetings online, it may be easy for them to avoid talking to each other by turning off their cameras or watching what they've already recorded. If the meeting organizer observes this happening in real time, they can intervene and encourage attendees to take part. This ensures that everyone's ideas are considered and that the meeting's objectives are attained. In addition, it aids in retaining participants' attention and concentration, both of which are required for cooperation and the formation of sound judgements. Real-time video conferencing makes it important to find still images and loop videos so that virtual meetings are productive and run smoothly. In this investigation, it was determined that the proposed classification model could appropriately categorize the data, and the summary of findings indicated the usefulness of classification models. A score of 1 for a model's accuracy means that it properly predicted each sample in the dataset. A recall percentage of 0.994% means that the model was able to correctly identify 99.4% of the real positive samples in the dataset. This shows that the model is sensitive to positive samples, which is a trait that many classification tasks look for. With an F1 score of 0.997, the model is very accurate and has a very good memory. In addition, it has done an excellent job of minimizing both false positives and false negatives. In addition, the F1 value reveals a high degree of precision and memory equilibrium. The mean squared error is often tiny, with a value of 0.0205, suggesting that the model's predictions are correct. Overall, it looks like the model works well in terms of its accuracy, sensitivity, precision, and number of mistakes. These results show that the model has a lot of potential to be used in real-world efforts to classify things. Yet, it is essential to remember that measurements alone do not provide a whole picture of the model's performance. In addition to metrics, the study should look at the quality of the data, the complexity of the model, and how easy it is to understand.

Reference

[1] M. Premkumar, S. R. Ashokkumar, G. Mohanbabu, V. Jeevanantham, and S. Jayakumar, “Security behavior analysis in web of things smart environments using deep belief networks,” *Int. J. Intell. Networks*, vol. 3, no. October, pp. 181–187, 2022, doi: 10.1016/j.ijin.2022.10.003.

[2] D. P. Tian, “A review on image feature extraction and representation techniques,” *Int. J. Multimed. Ubiquitous Eng.*, vol. 8, no. 4, pp. 385–395, 2013, doi: 10.1109/HIS.2012.6421310.

[3] H. Zhang, J. E. Fritts, and S. A. Goldman, “Image segmentation evaluation: A survey of unsupervised methods,” *Comput. Vis. Image Underst.*, vol. 110, no. 2, pp. 260–280, 2008, doi: 10.1016/j.cviu.2007.08.003.

[4] S. Gupta, “Chan-vese segmentation of SEM ferrite-pearlite microstructure and prediction of grain boundary,” *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 10, pp. 1495–1498, Aug. 2019, doi: 10.35940/ijitee.A1024.0881019.

[5] H. Goyal, K. Sidana, C. Singh, A. Jain, and S. Jindal, “A real time face mask detection system using convolutional neural network,” *Multimed. Tools Appl.*, no. November 2021, 2022, doi: 10.1007/s11042-022-12166-x.

[6] J. Faritha Banu, R. Revathi, M. Suganya, and N. R. Gladiss Merlin, “IoT based Cloud integrated smart classroom for smart and a sustainable campus,” *Procedia Comput. Sci.*, vol. 172, no. 2019, pp. 77–81, 2020, doi: 10.1016/j.procs.2020.05.012.

[7] “Erratum regarding missing Declaration of Competing Interest statements in previously published articles (Journal of King Saud University - Computer and Information Sciences, (S1319157818300545), (10.1016/j.jksuci.2018.04.001)),” *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 10. King Saud bin Abdulaziz University, pp. 1206–1207, Dec. 01, 2020. doi: 10.1016/j.jksuci.2020.10.026.

[8] R. Herrero, “A comparison of mechanisms for RTC in the context of IoT,” *Internet of Things*, vol. 8, p. 100110, Dec. 2019, doi: 10.1016/j.iot.2019.100110.

[9] S. Gupta, J. Sarkar, M. Kundu, N. R. Bandyopadhyay, and S. Ganguly, “Automatic recognition of SEM microstructure and phases of steel using LBP and random decision forest operator,” *Measurement*, vol. 151, p. 107224, Feb. 2020, doi: 10.1016/j.measurement.2019.107224.

[10] S. Gupta, J. Sarkar, A. Banerjee, N. R. Bandyopadhyay, and S. Ganguly, “Grain Boundary Detection and Phase Segmentation of SEM Ferrite–Pearlite Microstructure Using SLIC and Skeletonization,” *J. Inst. Eng. Ser. D*, vol. 100, no. 2, pp. 203–210, Oct. 2019, doi: 10.1007/s40033-019-00194-1.

[11] M. O. Farooq, “Multi-hop communication protocol for LoRa with software-defined networking extension,” *Internet of Things*, vol. 14, p. 100379, Jun. 2021, doi: 10.1016/j.iot.2021.100379.

[12] I. H. Sarker, “A machine learning based robust prediction model for real-life mobile phone data,” *Internet of Things*, vol. 5, pp. 180–193, Mar. 2019, doi: 10.1016/j.iot.2019.01.007.

[13] M. V. Alyushin, V. M. Alyushin, and L. V. Kolobashkina, “Optimization of the data representation integrated form in the viola-jones algorithm for a person’s face search,” *Procedia Comput. Sci.*, vol. 123, pp. 18–23, 2018, doi: 10.1016/j.procs.2018.01.004.

[14] B. Sathiyaprasad, “Ontology-based video retrieval using modified classification technique by learning in smart surveillance applications,” *Int. J. Cogn. Comput. Eng.*, vol. 4, no. February, pp. 55–64, 2023, doi: 10.1016/j.ijcce.2023.02.003.

[15] S. Chattaraj, T. Chakraborty, C. Koner, and S. Gupta, “Machine Learning in ASD,” in *Agile Software Development*, Wiley, 2023, pp. 345–362. doi: 10.1002/9781119896838.ch18.

[16] M. Khan, S. Chakraborty, R. Astya, and S. Khepra, “Face Detection and Recognition Using OpenCV,” *Proc. - 2019 Int. Conf. Comput. Commun. Intell. Syst. ICCCIS 2019*, vol. 2019-Janua, pp. 116–119, 2019, doi: 10.1109/ICCCIS48478.2019.8974493.

[17] M. L. R. Chandra, B. V. Kumar, and B. Sureshbabu, “IoT enabled home with smart security,” *2017 Int. Conf. Energy, Commun. Data Anal. Soft Comput. ICECDS 2017*, pp. 1193–1197, 2018, doi: 10.1109/ICECDS.2017.8389630.

[18] P. Mukherjee *et al.*, “Monitoring, Recognition and Attendance Automation in Online Class: Combination of Image Processing, Cryptography in IoT Security,” in *Proceedings of International Conference on Network Security and Blockchain Technology.*, Singapore: Springer, 2022, pp. 18–27. doi: 10.1007/978-981-19-3182-6\_2.

[19] M. Naveenkumar and V. Ayyasamy, “OpenCV for Computer Vision Applications,” *Proc. Natl. Conf. Big Data Cloud Comput.*, no. March 2015, pp. 52–56, 2016, [Online]. Available: https://www.researchgate.net/publication/301590571\_OpenCV\_for\_Computer\_Vision\_Applications

[20] B. Mondal, D. Chakraborty, N. K. Bhattacherjee, P. Mukherjee, S. Neogi, and S. Gupta, “Review for Meta-Heuristic Optimization Propels Machine Learning Computations Execution on Spam Comment Area Under Digital Security Aegis Region,” in *Integrating Meta-Heuristics and Machine Learning for Real-World Optimization Problems*, P. Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Ed. Springer Nature, 2022, pp. 343–361. doi: 10.1007/978-3-030-99079-4\_13.

[21] A. Królak and P. Strumiłło, “Eye-blink detection system for human-computer interaction,” *Univers. Access Inf. Soc.*, vol. 11, no. 4, pp. 409–419, 2012, doi: 10.1007/s10209-011-0256-6.

[22] N. J. Wang, S. C. Chang, and P. J. Chou, “A real-time multi-face detection system implemented on FPGA,” *ISPACS 2012 - IEEE Int. Symp. Intell. Signal Process. Commun. Syst.*, pp. 333–337, 2012, doi: 10.1109/ISPACS.2012.6473506.

[23] R. Padilla, C. C. Filho, and M. Costa, “Evaluation of haar cascade classifiers designed for face detection,” *J. WASET*, vol. 6, no. 4, pp. 323–326, 2012, [Online]. Available: http://www.waset.org/publications/2910

[24] C. Ryan *et al.*, “Real-time face & eye tracking and blink detection using event cameras,” *Neural Networks*, vol. 141, pp. 87–97, 2021, doi: 10.1016/j.neunet.2021.03.019.

[25] T. Sumer and J. Bhandari, “Reviews on Eye blink detection,” *Int. J. Sci. Eng. Res.*, vol. 8, no. 5, pp. 48–51, 2017, [Online]. Available: http://www.ijser.org