IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol. 9, Issue.1, January 2021 E-ISSN: 2347-2693

Real-Time Human Detection in Video Surveillance

Chalavadi Sravanth[1*], Gadde Harshavardhan[2], Kamineni. Kavya[3], Shaik Mohammad Akbar[4], Ch.M.H. Sai Baba[5]

[1,2,3,4,5]Department of Computer Science, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India

[*]*Corresponding Author: sravanthchalavadi@gmail.com, Tel.: +91 9493400542*

DOI: https://doi.org/10.26438/ijcse/v9i1.4450 | Available online at: www.ijcseonline.org

Received: 14/Dec/2020, Accepted: 20/Jan/2021, Published: 31/Jan/2021

Abstract— The basic Fundamental to human-centric computer vision is to make the human motion see and understandable by machines. The hectic task is that the video containing enormous amount of information in the form of pixels, much of meaningless to a computer unless it can decode the data within the pixels. To make it possible, computer what is the mechanism behind which pixel go together and what it represents. The process of detecting and tracking the pixels representing the form of humans is to be notified as Human motion capture. Where there is a lacking of count of the people and we want to overcome. We plan to achieve this goal using intermediate level deep learning project on computer vision concepts, where deep learning is an AI method that imitate the functioning of human brain in processing data for use of object detection, speech recognition, translating languages, and making decisions. OpenCV is the place where it deals will all sorts of camera related things and make the detection easier. This work represents that how a human is detected and counted using SVM. The main idea is to detect the patterns of human motion, to a larger extent which is independent of differences in appearance. To do so, an HOG descriptor is used to detect the patterns of the frame captured, the greatest use of this descriptor is that it detects the patterns with the direction of the movement of the captured picture and hence it makes the job easy to train the pictures using the SVM and get the human detected.

Keywords—Computer Vision; OpenCV; Support Vector Machine; HOG descriptor; Video Surveillance: Human Detection.

I. INTRODUCTION

Artificial Intelligence (AI) has been around for some decades in several theoretical forms and complicated systems; however, only recent advances in the power of computation and big data have enabled AI to achieve brilliant results in an vast and diverse number of domains. For example, AI have tremendously advanced the areas of computer vision [1], medical applications, natural language processing, and several other domains.

Human detection in image sequences is a trending and active research area within computer vision. In various applications, such as surveillance systems or safety systems in cars, the type of detection in there is used to trigger some sort of alarm. The intended application of the method presented here is outdoor video surveillance.

Most of the human detection systems detect human beings or Human faces in a single image. These types of approaches are based on the assumption that humans can be localized in every individual frame, based on a model of human appearance like shape, contrast, colour. However, in an uncontrolled outdoor environment such as the one considered in our application, human appearance will change due to various reasons like environmental factors such as light conditions, clothing, contrast, and identity. The image sequences could even be taken during the night with a light enhancing camera. Furthermore, the subjects can be camouflaged or masked. All these factors cause large variation in the appearance of both human and scene, by obscuring the interesting features for human/non-human classification.

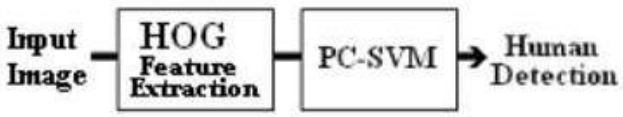


Fig: 1- flow of detection

This section will give a brief overview of our feature extraction as well as PC-SVM adopting chain, which is summarized. This method is based on by evaluating and estimating well-normalized local histograms of image gradient orientations in a dense grid by PC-SVM adopting. HOG feature extraction block which is used for better result may be dividing into several sub blocks as Gamma / Color Normalization, Orientation cells, Block Normalization, Computing the histogram of gradient orientations. Proposal algorithm is implemented by bifurcating the image window into small cells, for each cell by accumulating a local 1-D histogram of gradient with directions and edge orientations over the pixels of the cell considered. The combined histogram entries form the representation. For good and better invariance to illumination, shadowing, etc., it is also useful to contrast-normalize the local responses before they are being used by them. Somewhat larger spatial regions and using the

© 2021, IJCSE All Rights Reserved

results to normalize all of the cells in the block is done by accumulating a measure of local histogram. We will now refer to the descriptor blocks by normalization as Histogram of Oriented Gradient (HOG) descriptors. [2] Tiling the detection window with a dense in fact, overlapping grid of HOG descriptors and by using the converged feature vector in a PC-SVM based window classifier which gives our human detection chain.

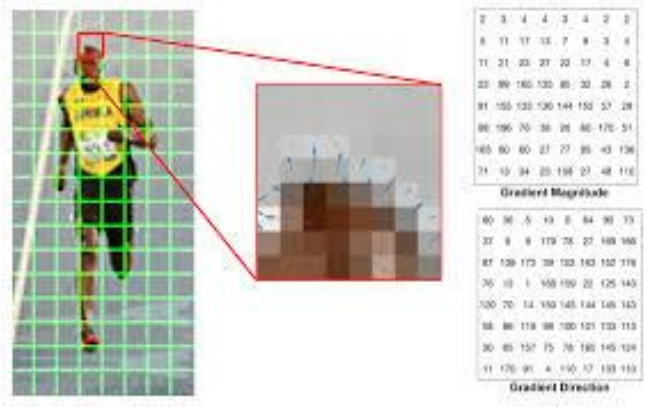


Fig: 2- Histogram Oriented Gradients

Notice how the direction of arrows points to the direction of change in intensity and the magnitude shows how big the difference is.

On the top, we can see the raw numbers with one minor difference representing the gradients in the 8×8 cells with one minor difference that the angles are not between 0 to 360 degrees but in 0 and 180 degrees. A gradient and its negative are represented by the same numbers those are called unsigned gradients. A gradient and its negative are represented by the same numbers. In other words, the one 180 degrees opposite to it and gradient arrow are considered the same. But, why not use the 0 to 360 degrees, empirically it has been shown that the unsigned gradients will work more effective than signed gradients for all types of detection. Some implementations of HOG Descriptor will permit you to specify that if you want to use signed gradients.

The next step is to create a histogram of gradients in these 8×8 cells. The histogram contains 9 bins corresponding to angles 0, 20, 40 …. 160.

After using HOG descriptors, support vector machines (SVM) are usually used in the classification stage. SVM are a set of supervised learning algorithms were introduced for linearly separable [3] and linearly non-separable data. SVM have been used in classification and regression problems in many fields such as text recognition, bioinformatics and object recognition, among others. They have also been used successfully in the detection of persons. Here, we are also interested in knowing if linear SVM trained with colour images which will provide better results in classifying infrared images (IR) without retraining. Should this be true, we could overcome the lack of larger datasets in the infrared spectrum.

1.1 HOG for Feature Extraction

Histogram of oriented gradients (HOG) consists of a series of steps that provide a list of image features representing the objects contained in an image in a clear schematic manner. The image features are then after used to detect the same objects in various other images. In our particular case, we are now interested in obtaining accurate features for human detection.

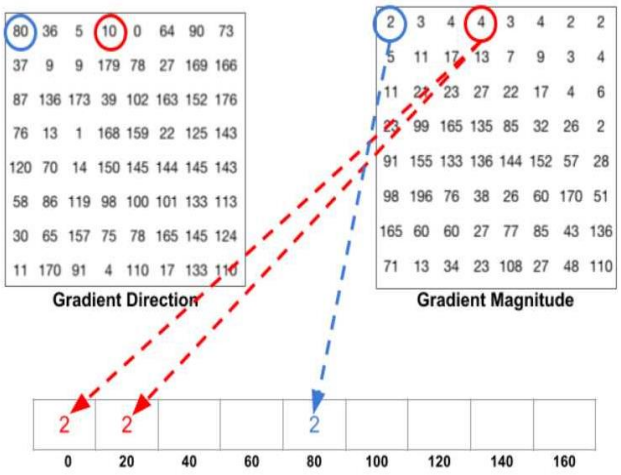


Fig: 3-Histogram of Gradients

The HOG features are mostly used for detecting objects. HOG descriptor decomposes an image into tiny square shaped cells, it will compute a histogram of oriented gradients in each and every cell and then normalizes the result by using a block-wise pattern, and return a descriptor for each cell. Stacking the cells into a square shaped image region can be used as an image window descriptor for detecting an object by taking an example by means of an SVM.

One of the major reasons to use a feature descriptor to describe a patch of an image that is a part of an image is that it provides a compact representation. An 8X8 patch of an image contains 8X8X3 which is 192-pixel values. The gradient of this patch contains 2 values in magnitude and direction per pixel which sums up to 8X8X2 which is 128 numbers. By the end of this statement, we will see how these 128 numbers which are represented using a 9-bin histogram that can be stored as an array of 9 numbers. Not only is the representation more compact but also calculating a histogram over a patch of an image that makes this representation more robust to noise. Gradients which are individual may have noise, but a histogram over 8X8 patch of image makes the representation much less sensitive to noise.

There is one more detail to be aware of. If the angle is greater than 160 degrees, it is between 160 and 180, and we know the angle wraps around making 0 and 180 equivalents. So, the pixel with angle 165 degrees contributes directly proportional to the 0-degree bin and the 160-degree bin.

© 2021, IJCSE All Rights Reserved

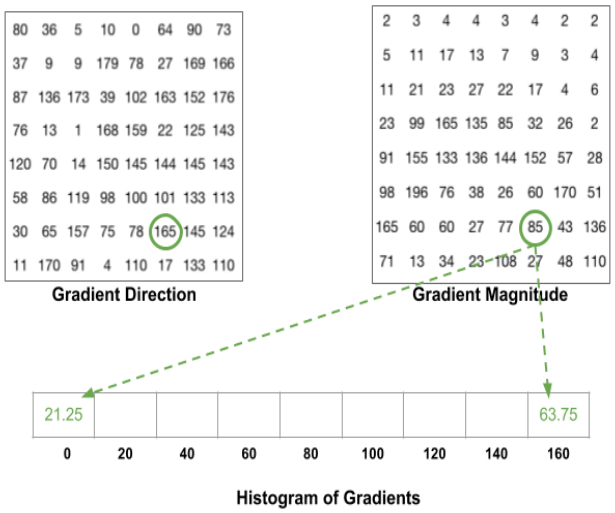


Fig: 4- 8×8 image patch

The object detection is the key technique in the field of intelligent transportation, a rich body of scholarly work exists on the subject. In this field, the targets frequently include cars or traffic signs; such targets are generally accompanied by rich prior information, which can be utilized to enhance the accuracy of target tracking (for example., car shape and group behaviour are used to distinguish and predict the cars), and spatial and scale prior is used to improve the detection performance of traffic signs.

II. RELATED WORK

For fixed cameras, a combined motion segmentation and optical flow algorithm for moving object tracking were found in. Nonetheless, optical flow was only calculated on pixel level using motion segmentation to ensure that no comparisons were made between background-foreground regions. In, the optical flow is calculated in silhouette regions using 2-way ANOVA[4] and object segmentation is used to minimize the effect of brightness change. In proposed detecting abnormal motions in crowd monitoring scenes using Horn and Schanck method in video streams. A detection and tracking method for outdoor scenes by applying line computed by gradient-based optical flow and edge detector was proposed in. The edge-based feature has robust performance and is insensitive to illumination changes. For free-moving cameras, motion clustering and classification were used to detect the moving objects in. [5] Fusion Horn and Schanck method in were proposed for aerial-coloured images using least squares to estimate the flow field of each colour plane and then fuse all distinct fields into one field. Lucas and Kanade optical flow method were used in and were combined with stereo camera for UAVs to navigate urban environments in a reactive way by performing a control-oriented fusion.

In the reviewed literature, Lucas and Kanade method was more suitable and provided effective results when the flow was presumed to be constant in local neighbourhood of the pixel under consideration in, where the method's equation is solved for all pixels in the local neighbourhood by least squares criterion. While Horn and Schunck method was more effective for scenarios with the smooth flow over the entire frame, i.e., motion of objects is not restricted to a certain neighbourhood. Furthermore, many researchers have attempted to apply hybrid approaches for motion detection, which uses a combination of two or more different methods including optical flow to solve the motion detection problems.

Nine optical flow computation techniques were studied and evaluated and eight of the classical optical flow algorithms and were compared and their performance on complex scenes was objectively evaluated in. A hybrid algorithm of the optical flow field and the temporal difference method was proposed by to detect the motion object area. The method uses the temporal difference to calculate the difference between two or three consecutive frames, low pass filter and edge detection of moving objects were used to filter the differential image. However, the optical flow method implemented the Horn and Schanck algorithm which is used to compute the velocity from a spatiotemporal derivative of image intensity.

Though the combination of temporal difference and optical flow in was indicated to be suitable and effective for objects moving within stationary camera video, in moving cameras the combination does not produce effective results.

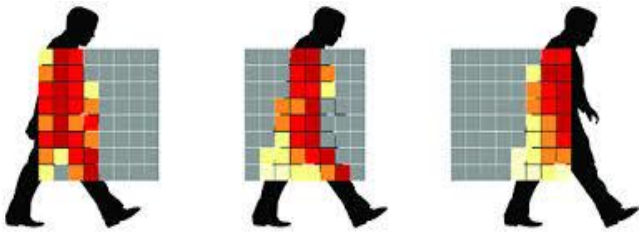


Fig: 5-Motion Detection

2.2 Human Detection

This module is responsible for the detection of the algorithm. Ideally, the algorithm for detection is to be executed on each input frame. However, this will helps in the system from achieving its real time requirements. Instead, the algorithms for detection in our implementation is invoked for each two seconds. The location of the human targets in the available remaining time is determined by tracking using the tracking algorithm on the detected humans. [6] Therefore, to speed up the process, the algorithm for detection in the entire frame does not look for humans. Instead, it seems to look for humans in the regions determined to be fore-ground regions. To find the foreground regions, a algorithm of stabilization is used to align the present frame with a preceding frame and with a succeeding frame in a continuous process. After alignment, the present frame is removed from the two other frames. The result of each removal is threshold to form a binary image that represents the locations of fore-ground objects in the two removed frames. To get the locations of the foreground objects in the present frame,

© 2021, IJCSE All Rights Reserved

the results of the two removals are combined by an AND gate operation. The removal is performed in the hue channel of the HSV color space.

When the duration of a track goes high some specific length, mostly, two seconds, the analysis of motion module is called out. The analysis of motion module analyses the periodicity triggered in the track. Taking on the consideration of result of this analysis, it decides that the tracked object that has identified is a human or not. In this way, the results of the detection are checked twice by the analysis of the motion. In our experiments, that results in a deduction in the false positives given by the detector.

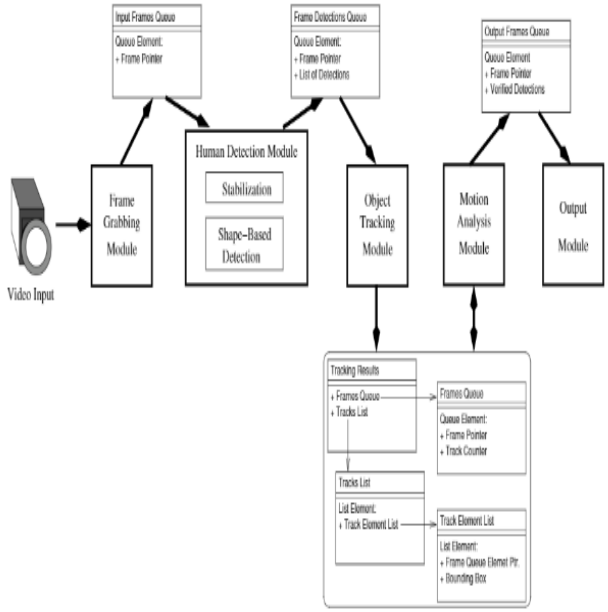


Fig: 6- Basic system architecture

2.3 Human Detection Algorithm

This algorithm identifies for humans in the image by resembling its edge features to a database of templates of humans. [7] Examples of these templates are shown in Fig. 7. The matching is done by computing the average Chamfer distance between the template and the edge map of the target image area. The image in the area under acceptance must be of the same size as the template. Let the template be a binary image that is everywhere except for the silhouette pixels where the value is1, and let the Chamfer distance transform of the edge map of the target image area will be denoted by "C". The distance between a template and the target image area can be computed by

$$D(I, T) = 1|T| \sum iCiTi$$

Where |T| (modulo)is the number of silhouette pixels in T, Ti is the pixel number in T, and Ci is the Chamfer distance value at pixel number i n I. The lesser the value of the distance between the target image [8]area and template, the greater the match between them.

Fig: 7-Example Human Silhouette Templates

III. METHODOLOGY

Human-Computer interaction has received lately renewed interest due to advances in technology. Applications, which could have been considered fiction a decade ago, are now possible due to smaller, cheaper and powerful for computing devices. For example, smart rooms could have smart cameras in charge of tracking the user so his/her gestures and movements are evaluated as input commands. It is obvious that visual tracking should be a fundamental capability of the system, and since the interaction between the room and the user is required, this should be accomplished in real-time.

A common proceed towards to tracking an object in a video stream is to use an object detector, a classifier and a motion estimator or tracker in sequential order. [9] The object detector scans a frame from the video stream and selects the candidates to be analyzed by the classifier. The classifier evaluates every candidate assigning it a measure that indicates the likeliness of the candidate to be the object searched.[10] The candidate with the best score is then locked and the tracker is used to follow it through the field of view.[11] This standard tracking architecture, represented below.

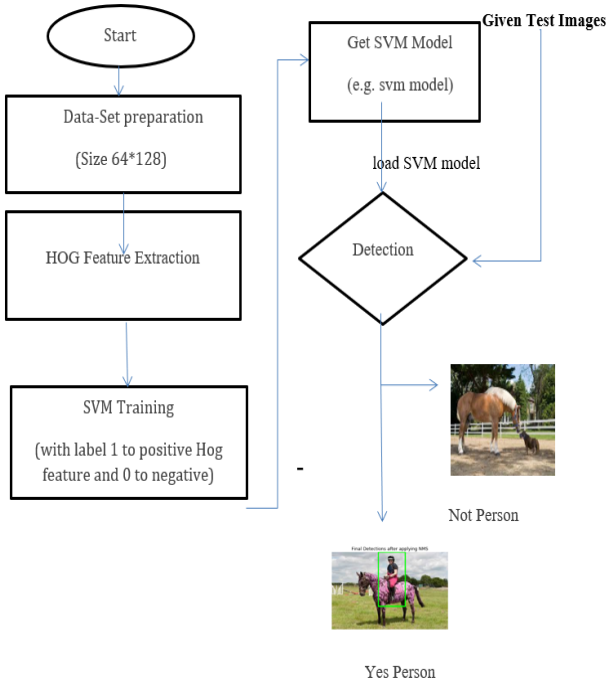


Fig: 8- Standard Tracking Architecture

© 2021, IJCSE All Rights Reserved

In recent years, Support Vector Machines have been a breakthrough in the machine learning community. [12]They have been proved effective in a variety of tasks related to classification such as Character Recognition, Image Rotation Detection, Gesture Recognition, Face Recognition, Bioinformatics, and Spam Classification.

SVM performance in accuracy and generalization makes them an perfect candidate for this standard tracking architecture. [13] However, in order to obtain their best performance, different issues should be addressed: feature selection, tuning, training and classification complexity. The first two influence directly the classifier accuracy and generalization properties. The last two have an impact on the implementation of SVMs.

The features given as input to the classifier are of very importance to its accuracy. If the wrong set of features are used, not even the best classifier would be able to perform correctly. Selecting the correct input is problem dependent and the overall performance of the system depends on this step.

As a supervised learning machine, SVM's need to be trained. The training process amounts to solving a Quadratic Programming (QP) problem. Due to the size of the problem, [14] standard QP solvers cannot be used here. This training complexity has limited the wide spread usage of svm. Even though new training algorithms have been developed recently, their time performance is problem dependent and the computational time needed is still large. The training process gives as a result a set of vectors, called support vectors, which characterize the decision function of the classifier. The computational time involved in the evaluation of the classifier is directly proportional to the number of support vectors. This number is problem dependent but empirical results have shown that it is approximately between 10% and 20% of the number of training vectors in the training set. Since the common training set contains large amounts of data, the number of support vectors is also large. 4 Therefore, SVM's time performance in test phase is less than other classifiers such as Neural Networks.

This work explores the use of SVM's as classifiers in synchronal tracking systems. From the issues presented above, it addresses the ones which can be applied to any visual tracking problem, namely decreasing the training time and computational time involved in classification. As a result, this work presents a framework for synchronal visual tracking applications using SVM's.

Support Vector Machines - SVM are learning systems that classifying a given input in two classes, trained with a learning algorithm from optimization theory that implements a learning bias. [15] One of its advantages over other classifiers is that is has a strong theoretical foundation on statistical learning theory. In this section, a brief description of SVM as a classifier will be given

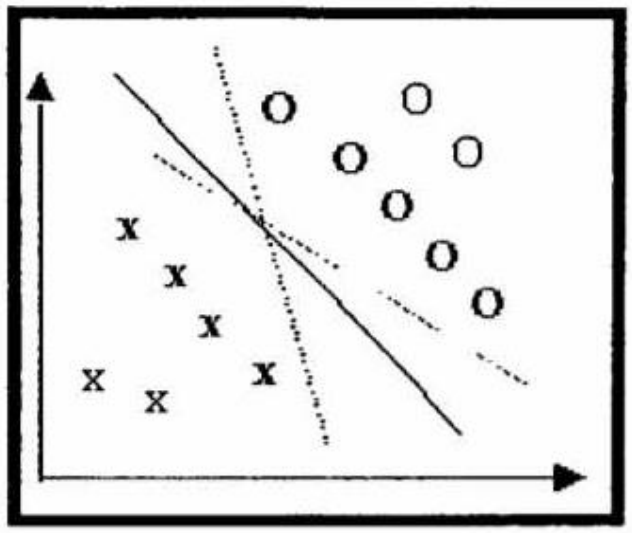
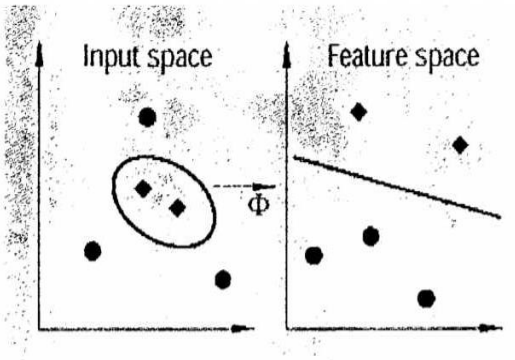


Fig: 9- SVM decision boundary

The real power of Support Vector Machines is that they can also be applied to nonlinear classification problems. [16] To deal with non-linearly separable classes, a non-linear mapping $: Xi + T is used to map the input space (X) into a higher dimensional space, called the feature space {T), where data becomes linearly separable.[17] Under this mapping the decision rule (2.4) transforms into:



$\Phi = \mathcal{X} \rightarrow \mathcal{F}$

Φ is the mapping function between input space and feature space

Fig: 10- Example of nonlinear Mapping

IV. RESULTS AND DISCUSSION

We have learned how to create a people counter using HOG and OpenCV to generate an efficient people counter. We developed the project where you can supply the input as: video, image, or even live camera. This is an intermediate level project, which will surely help you in mastering python and deep learning libraries.

After running the human detection python project with multiple images and video, we will get:

© 2021, IJCSE All Rights Reserved

Fig: 11

This work explores the use of Support Vector Machines (SVMs) as classifiers in real-time tracking systems. From the issues presented above, it addresses the ones which can be ap plied to any visual tracking problem, namely reducing the training time and computational time involved in classification. As a result, this work presents a framework for real-time visual tracking applications using SVMs.

V. CONCLUSION AND FUTURE SCOPE

A real-time Human tracking system based on SVM was implemented and tested. Significant amount of work has been done with a view to detect human beings in a surveillance video. However, the low-resolution images from the surveillance cameras always make this work challenging. It also demonstrated that a significant reduction on the number of support vectors can be achieved without compromising classification accuracy significantly.

Here the project can be used to detect the humans and count them as well, in various ways like providing path to a video or providing path to the image or by using the web camera and all the different ways work effectively.

This project can further be worked on and updated to identify suspicious human behaviour or to identify gender of the people in the surveillance. There needs to be a complex computational model for this that can be developed if the factors like noise in surveillance, background light problem and training the model to exactly identify the correct output required can be taken care of.

To conclude, the project “Real-Time Human Detection in Video Surveillance” has been executed quite successfully giving promising results and showing a great potential and scope for further enhancements and modifications.

REFERENCES

[1]. W Fernando et al., in Information and Automation for Sustainability (ICIAfS), 2014 7th International Conference on. Object identification, enhancement and tracking under dynamic background conditions, IEEE, **2014**

[2]. Xu, R., Guan, Y., & Huang, Y., Multiple human detection and tracking based on head detection for real-time video surveillance. *Multimedia Tools and Applications*, ***74*(3), 729-742, 2015.**

[3]. Dalal, N., Triggs, B., & Schmid, C., Human detection using oriented histograms of flow and appearance. In *European conference on computer vision*. Springer, Berlin, Heidelberg. **pp. 428-441, May, 2006.**

[4]. https://www.researchgate.net/publication/4215591_Real-Time_Human_Detection_Tracking_and_Verification_in_Uncontrolled_Camera_Motion_Environments

[5]. Sulman, N., Sanocki, T., Goldgof, D., & Kasturi, R., How effective is human video surveillance performance? In *2008 19th International Conference on Pattern Recognition*, IEEE, **pp. 1-3, December, 2008.**

[6]. Murat EKINCI, Ey¨up GEDIKL "Silhouette Based Human Motion Detection and Analysis for Real-Time Automated Video

© 2021, IJCSE All Rights Reserved

Surveillance" Dept. of Computer Engineering, Karadeniz Technical University, Trabzon, TURKEY, **2005.**

[7]. Pang, Y., Yuan, Y., Li, X., & Pan, J., Efficient HOG human detection. *Signal Processing*, ***91*(4), 773-781, 2011.**

[8]. N. Cristianini and J. Shawe-Taylor. Support vector net [http://www.supportvector.net/]. Cambridge University, **2015.**

[9]. Osuna, E., Freund, R., & Girosit, F., Training support vector machines: an application to face detection. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition*. IEEE. **pp. 130-136, June, 1997.**

[10]. Smail Haritaoglu, David Harwood and Larry S. Davis "W4: Who? When? Where? What? A Real Time System for Detecting and Tracking People" Computer Vision Laboratory, University of Maryland College Park, 1998. *(PDF) Real-Time Human Motion Detection and Tracking*. Available from: https://www.researchgate.net/publication/251852856_Real-Time_Human_Motion_Detection_and_Tracking

[11]. Kim, Y., & Ling, H., Human activity classification based on micro-Doppler signatures using a support vector machine. *IEEE Transactions on Geoscience and Remote Sensing*, ***47*(5), 1328-1337, 2009.**

[12]. Foroughi, H., Rezvanian, A., & Paziraee, A., Robust fall detection using human shape and multi-class support vector machine. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE. **pp. 413-420, December, 2008.**

[13]. Dadi, H. S., & Pillutla, G. M., Improved face recognition rate using HOG features and SVM classifier. *IOSR Journal of Electronics and Communication Engineering*, ***11*(4), 34-44, 2016.**

[14]. Chen, P. Y., Huang, C. C., Lien, C. Y., & Tsai, Y. H., An efficient hardware implementation of HOG feature extraction for human detection. *IEEE Transactions on Intelligent Transportation Systems*, ***15*(2), 656-662, 2013.**

[15]. Liang, Y., Reyes, M. L., & Lee, J. D., Real-time detection of driver cognitive distraction using support vector machines. *IEEE transactions on intelligent transportation systems*, ***8*(2), 340-350, 2007.**

[16]. Vijayalakshmi, S., & Kumar, N. S., A Review on Fetal Brain Structure Extraction Techniques from Human MRI Images. **2008.**

[17]. Vayadande, K. B., & Yadav, S., A Review paper on Detection of Moving Object in Dynamic Background, **2018.**

© 2021, IJCSE All Rights Reserved