Convolutional Neural Network Based Approach to Detect Pedestrians in Real-Time videos

Sandhya N, AnirudhMarathe, JS Dawood Ahmed, Aman Kumar, Harshith R

Abstract- Pedestrians in the vehicle way are in peril of being hit, along these lines making extreme damage walkers and vehicle inhabitants. Hence, constant person on foot identification was done through a set of recorded videos and the system detects the persons/pedestrians in the given input videos. In this survey, a continuous plan was proposed dependent on Aggregated Channel Features (ACF) and CPU. The proposed technique doesn't have to resize the information picture neither the video quality. We also use SVM with HOG and SVM with HAAR to detect the pedestrians. In addition, the Convolutional Neural Networks (CNN) were trained with a set of pedestrian images datasets and later tested on some test-set of pedestrian images. The analyses demonstrated that the proposed technique could be utilized to distinguish people on foot in the video with satisfactory mistake rates and high prediction accuracy. In this manner, it tends to be applied progressively for any real-time streaming of videos and also for prediction of pedestrians in pre-recorded videos.

Index Terms: Aggregated Channel Features (ACF), Convolutional Neural Networks (CNN), HAAR, Pedestrians, SVM(Support Vector Machine)

I. INTRODUCTION

Item location in a video stream is picking up significance now-a-days. Traditional strategies for object discovery istedious since the hunt calculations depend on sliding window worldview, which means, a window of reasonable size is moved allthrough the picture and pixel coordinating is done crosswise over moving window. At present, different person on foot location in profound learning approaches have been created in the writing, for the most part by broadening the primary pipeline of effective item locators, for example, R-CNN, quick R-CNN and quicker R-CNN. This pipeline contains a blend of: proposition extraction and CNN assessment (being indistinguishable from the cross breed conspire, however progressively conventional). In this methodology, we receive a comparative design, which pursues the R-CNN pipeline. Notwithstanding, we don't make a difference any bouncing box relapse, differentiating to. Additionally, and since we are worried about speed, we present an ACF score dismissal limit. beneath this worth, the proposition are killed and are not prepared by the CNN, permitting to accomplish quicker running time figures.For person on foot location, firstly, we receive a profound learning based methodology, by utilizing pre-prepared models. Secondly, we can definitely diminish the exertion related with the comprehensive hunt performed during the sliding window process.Our fundamental center is given to the mix of the PC vision module (person on foot recognition technique) with the HAN module, in request to accomplish an exact and quick enough framework. Computational exertion related with the comprehensive hunt performed during the sliding window process. To achieve this, we course the ACF (non-profound) locator with a CNN. The ACF locator is picked in light of the fact that it displays quick running time figures and high precision when contrasted with other non-profound indicators. This proposed course technique is twofold: first, It gives a particular inquiry approach that fundamentally improves the computational proficiency, since just the yield recommendations of the ACFare considered; at that point, by falling with the CNN, we can help the presentation of the ACF indicator (i.e. improve the characterization exactness of the ACF proposition by diminishing the quantity of bogus positives.)

II. PROPOSED METHOD

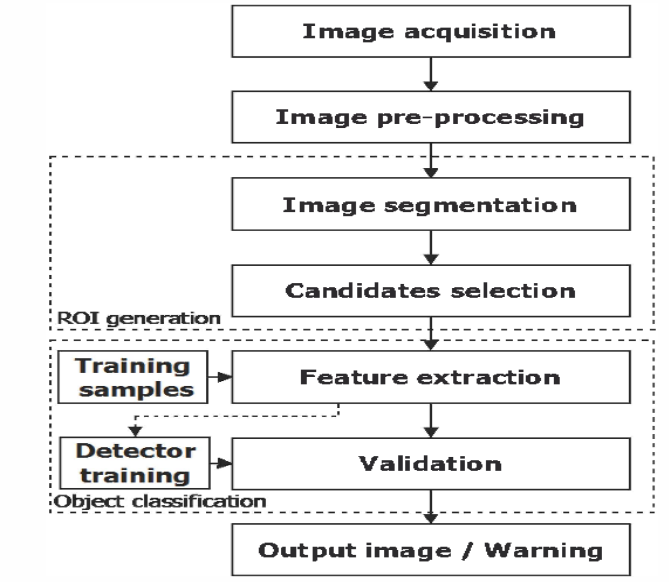

Fig.1: Steps/modules followed in the methodology

Revised Manuscript Received on November 10, 2020.

Sandhya N*,Department of Computer Science and Engineering, Dayananda Sagar Academy of Technology and Management, Bangalore (Karnataka), India. Email: dr.sandhya1909@gmail.com

Anirudh Marathe,Department of Computer Science and Engineering, Dayananda Sagar Academy of Technology and Management, Bangalore (Karnataka), India. Email: anirudhmarathe2006@gmail.com

JS Dawood Ahmed,Department of Computer Science and Engineering, Dayananda Sagar Academy of Technology and Management, Bangalore (Karnataka), India. Email: jsanees.005@gmail.com

Aman Kumar, Department of Computer Science and Engineering, Dayananda Sagar Academy of Technology and Management, Bangalore (Karnataka), India. Email: amanwildshark@gmail.com

Harshith R., Department of Computer Science and Engineering, Dayananda Sagar Academy of Technology and Management, Bangalore (Karnataka), India. Email: harshith.r.reddy@gmail.com

Fig. 1 represents the working modules of the methodology of pedestrian detection:

A. Image Acquisition:

Image acquisition is the creation of a digitally encoded representation of the visual characteristics of an object, such as a physical scene or the interior structure of an object.

B. Image Pre-processing:

Pre-processing is the improvement of the imagedata that suppresses unwilling distortions or enhances some image features important for further processing, although geometric transformations of images (e.g. rotation, scaling, and translation) are also classified among pre-processing methods.

C. Image Segmentation:

Image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as image objects). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.

D. Feature Extraction:

It starts from an initial set of measured data and builds derived values (features) intended to be informative and non- redundant, facilitating the subsequent learning and generalization steps and in some cases leading to better human interpretations.

E. Candidate Selection:

It is a process that assigns a label to an object based on its descriptors. In our case after detecting the pedestrians, the system assigns a label to the pedestrians based on the description of the pedestrians.

F. Validation:

The last stage that finally validates the object is a classifier. The most common classifiersare: support vector machine (SVM) as example of the supervised learning method, neural networks, self-organizing maps (SOM),andmatrices of neurons.

G. Output image/Warning:

The final output of the ACF(Aggregate Channel Feature) technique is the output of the pedestrians and also gives a count of the number of pedestrians that are actually present on the road/crossing at a given time depending on the safety measures to be taken for the pedestrians to avoid any type of accidents. The first methodology of our project is the usage of the SVM (Support Vector Machine) classifier with HOG (Histogram of Oriented Gradients) and HAAR cascades features for pedestrian detection. There are various features of HAAR cascades available, among which we have made use of the HAAR "full-body" features for the detection of pedestrians since we treat pedestrians as humans and they can be distinguished from others by various body parts such as hands, legs and torso. In the methodology of SVM with HOG, we took an input video and to detect the pedestrian regions of interest (ROI), we make use of a sliding window of size of 4x4 dimensions of length and width with a padding of 8x8 pixels in each x and y directions. The normal size of the sliding window is of 4x4 dimensions where we stride across the entire given input video with the 4x4 matrix, i.e. by taking 4 pixels in each x and y directions throughout the image for looking the features of the pedestrians. The typical size of the padding is 8x8 pixels. After detecting the features of pedestrians, we draw a bounding box around the pedestrians which is our regions of interest and display it in the output window by choosing a video of our choice from the file-dialog and running the video player as shown in fig. 3.

The HAAR cascade is the second methodology that's implemented in our paper. We make use of HAAR "full-body" features of the pedestrians to detect the full body of the pedestrians. We take an input video and detect the pedestrians regions of interest. The detector makes use of the HAAR "full-body" features and detects the humans in the input video by drawing a bounding box around the humans ROI in the output window by selecting any required input video of our choice from the file-dialog and running it on the video player as shown in fig. 4.

The third method proposed is that of the K-Nearest Neighbors.

We acquired the pre-requisites of the K- nearest neighbor algorithm by collecting our own dataset which comprised of over 70 images of pedestrians. We collected these datasets by choosing an input video and paused the video to look out for the areas where we can find the pedestrians. We then draw the bounding boxes around the pedestrians and we store it into a different directory named "training_data" automatically after drawing the bounding boxes. We also use a "test_data" directory to test for the images to see the prediction of the algorithm. The K-NN algorithm implemented works on the contour detections of pedestrians.Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity. So for training the algorithm, we take our own dataset and take just 10 images for training. During the training, if the algorithm detects a full-body of the human, we press a random key. For predicting the

Pedestrians testing set, we provide only a random image from the "test_data" directory. The algorithm detects the full-body of the pedestrians.

Our final methodology implements the Convolutional Neural Network approach for detecting the pedestrians. We used 2 different directories: one for training the network and other for testing the network. For training the network, we made use of the "Cats and Dogs" dataset directory which consisted of around 1000 images of dogs and cats each. We train these images based on training parameters such as batch-size, number of epochs and steps per epoch for training. We built the neural network from scratch and trained the neural network by using the cats images and dogs images and then tested them on the dataset images which consisted of around 200 images each of cats, dogs, horses and humans classes. For our application, we trained the network using a batch-size of around 100 images, the total number of iterations i.e., the number of epochs as 10 and used 100 steps per each epoch as a step-size. By training the network using this configuration, we achieved an accuracy of around 89% where the network detected the classes of cats and dogs correctly without any ambiguity. We also tested the network for the accuracy of prediction of humans and horses classes datasets. The network classified correctly the humans classes with a highest accuracy of 99%, which was useful for the current application for the detection and validation of the pedestrians which we will be running on the videos dataset. We will be presenting our results and inferences obtained in a tabular form in the next section of results and analysis.

The following are the formulas/equations required for evaluation and functioning of the Convolutional neural network:

$w_{ij}= w_{ij}+ \Delta w_{ij}$ (1)

$\Delta w_{ij}= - \eta. (\delta E/\delta w_{ij})$ (2)

$\Delta w_{ij}= - \eta.o_i\delta_j$ (3)

Error = $1/2 (t_d-o_d)^2$ (4)

Equation (1), (2) and (3) represent the training rules of the Convolutional Neural Network. The training of the Convolutional Neural Network basically has updating of the weights of the neural network as it passes through the various layers of the network. As it passes through each layer, the weights are updated using (1). The change in weights when it passes through a layer is given by (2), where the symbol η is a constant which is also called as the learning rate of the neural network.o_i represents the output which is obtained after passing through each of the layer in (3). The error term which is calculated after passing through each layer is given by (4).$t_d$representsthe target and $o_d$represents the output for the training example d and Error represents the error for the particular example. In short, we train the various examples of different images of humans and test it using our network. As the training of the example goes on, the weights gets updated at every layer and at the end of the output layer, we obtain the final trained weights for the training examples and store it in a weights file. Later these updated weights file is used to test the disjoint set of testing images and hence we get the accuracy of the network.

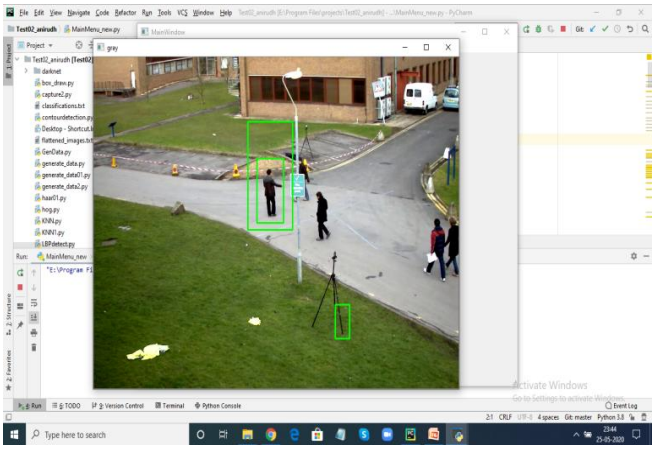

Fig 4: Pedestrian detection using SVM+HAAR

Fig 2: Architecture of a CNN model

Fig. 2shows the architecture of the Convolutional Neural Network. It has the following layers:

1. Input data
2. Convolutional Layer
3. Fully connected Layer
4. Output
5. Error

A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of a series of convolutional layers that convolve with a multiplication or other dot product. The input layer takes images of dimensions 64x64 with 3 as the depth of the image. The activation function is a RELU (Rectified Linear Unit) layer, and is subsequently followed by additional convolutions such as pooling layers and fully connected layers referred to as hidden layers because their inputs and outputs are masked by the activation function and final convolution.Then after passing through a convolutional layer, the image becomes abstracted to a feature map, with shape (number of images) x (feature map height) x (feature map width) x (feature map channels).Convolutional layers convolve the input and pass its result to the next layer.Pooling layers reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, typically 2 x 2. There are totally 500 nodes used in our CNN model.Fully connected layers connect every neuron in one layer to every neuron in another layer.The flattened matrix goes through a fully connected layer to classify the images.

III. RESULTS AND DISCUSSION

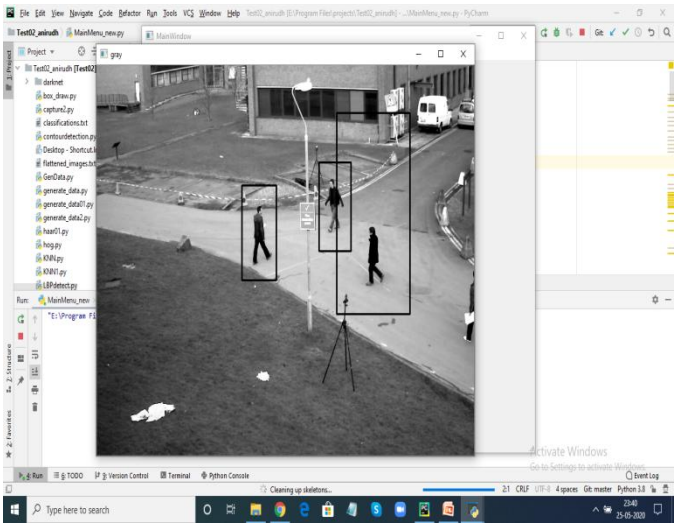

Fig 3: Pedestrian detection using SVM+HOG

The above table gives a comparison of accuracy versus varying values of training parameters such as batch-size, epochs and number of steps per epoch. We started off initially with 100 batch-size, 10 epochs and 30 steps per epoch. We trained the directory consisting of about total of 2000 images of dogs and cats taken collectively. We were getting an accuracy of 40% after testing it on the testing set of images which contained 200 images of each classes of dogs, cats, horses, and humans. We were getting a such a low score because we didn't have much permutations and combinations of images when we were dealing with only 200 images for training. We then selected around 2000 images total and when we started to train them and when we tested them on a set of 200 images, we started to get a better score and the score kept increasing rapidly. We then took a batch-size of 50 images, 10 epochs and 100 steps per epoch, our score for testing or prediction raised upto 69%.

Next we tried to optimize the learning rate of the network by selecting a batch-size of 100, 10 epochs and 100 steps per epoch, we found out that the network not only predicted the cats and dogs images accurately, but it was predicting the humans classes as well with an accuracy of 99% and the error rate was too quite minimal. Hence we observe that as we give the amount of batch-size more, the system gets more choices of training and testing a large permutations and combinations of images and
hence it gives a more better accuracy with a minimal error rate. We also observe that as we increase the number of steps per epoch gradually, the network takes more time to get trained and hence it receives a high amount of training which is needed for the better prediction of any given test datasets. Overall we observe that increasing the steps per epoch and also the batch-size, the network predicts the testing data more accurately. So in our case, a batch-size of 100 and 100 steps per epoch gives us an accuracy of around 89% which is betterfrom our previous values of accuracy and also the training parameters as evident from Table 1.Fig.5 represents the plot describing the change in the accuracy levels of prediction with varying values of the number of steps per epoch. X-axis denotes the steps per epoch and the y-axis denotes the predictive accuracy. It is evident from the graph that the value of predictive accuracy goes on increasing with the increase in the number of steps per epoch. For the first bar in the above graph, the number of training images (batch-size) is 100 and for 30 steps per epoch, we obtain a predictive accuracy of 40%. Similarly for a batch-size of 50 and 50 steps per epoch, we obtain an accuracy of 69% and for 100 batch-size and 100 steps per epoch, we get an accuracy of 89%.

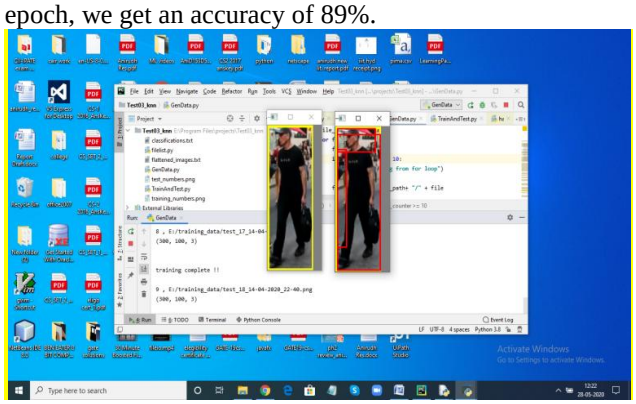

Fig. 6: Training of pedestrian dataset

Table 1: Accuracy v/s Training Parameters

Batch-size	Epochs	Steps per epoch	Accuracy
100	10	100	89%
50	10	50	69%
100	10	30	40%

Fig.3 demonstrates the working of the pedestrian detection using SVM+HOG detector. The figure is grayed out and then the bounding boxes are drawn over the pedestrians detected. In the fig. 4, we demonstrate the working of the SVM+HAAR cascades pedestrian detection. We made use of the HAAR "full-body" features of the pedestrians to detect them. The detector makes use of these features and then detects the pedestrians by drawing bounding boxes around the pedestrians. For this detection, we make use of the colored images.

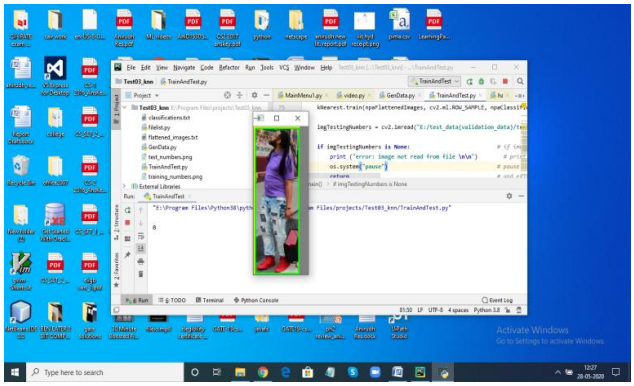

Fig 7: Testing of pedestrian test dataset

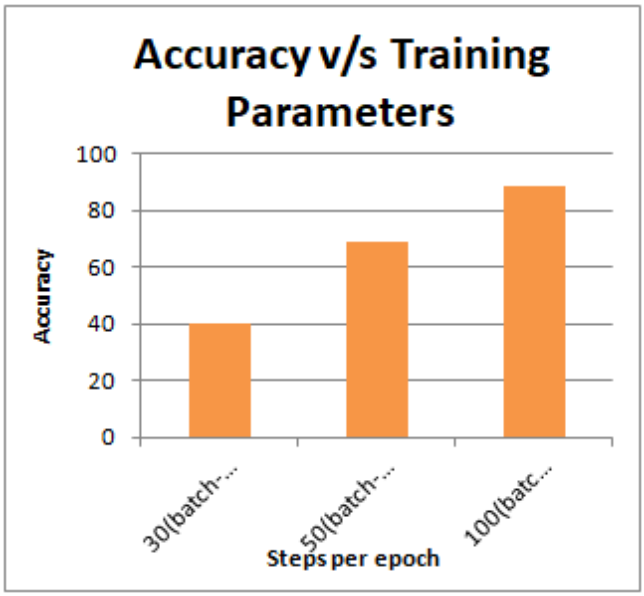

Fig 5: Plot showing the behavior of accuracy with the varying steps per epoch and a constant epoch size

The above figure Fig. 6 shows the training of the pedestrian dataset. For the training of the pedestrian datasets, we trained around 10 data images. During the training phase, from fig. 6, we observe that the red color boxes which are drawn by the system are the parts of the body that have been detected. The algorithm learnsby the method of contour detection where the system detects the contours which are continuous points having the same intensity values. By detecting those contours present in the human dataset, the algorithm detects the parts of the human body and draws the red boxes. Similarly in this way, we completed the training of 10 data images from a batch of70 images. After training 10 batches of images, we go for the testing of any random test images of pedestrians. Next phase consists of the testing of the test images from our test directory consisting of 10 images for testing. Out of 10 images we test for one pedestrian image to predict the full body of the pedestrian.

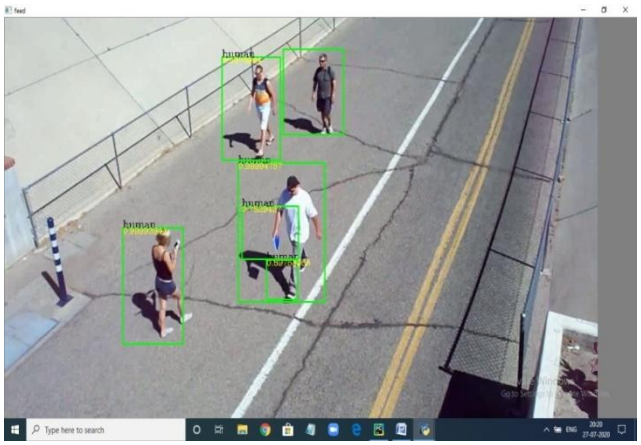

Fig 8: Working of CNN model on a given pre-recorded video file

As seen from Fig. 7, when we run the test process, we find that a desired image on which we wanted to test the algorithm on appears on the screen with a green colored box drawn on it. The green colored box denotes that the full body of the pedestrian has been detected. During the training process, we gave a key 0 if the system detects a part of the body of the human and a key of 1 if it detects the full-body of the human. In the fig. 7, during the testing of a desired image, the green colored box appears on the image of the pedestrian along with the key 1 as the output which denoted the key previously we were using to detect the full body of the pedestrian. The above figure fig. 8 shows the working of the pedestrian detection system in a given pre-recorded video file. The Convolutional Neural Network has been built by using the weights (.h5) file which is generated by the training of 2000 images of various categories such as cats, dogs and humans and then the network has been tested on a testing set of pedestrians which consisted of about 80 different images of humans. As it is evident from the above graph and tabular representation, we achieved an accuracy of the system of about 89%. This data was later stored in a separate weight file. This resultant weights file was then used in our above implementation of the CNN. As we see from the above output, the network is first capturing the pedestrians present in the video frame by drawing the bounding boxes on them by detecting the contours of the pedestrians, which is a set of points that forms a continuous boundary in an image. After detection of the pedestrians, it is labeling the captured pedestrians with the label as "Human" along with the accuracy of detection which is displayed below the label. From this figure, we observe that the network is also detecting the shadows of the humans as the 'pedestrians', because the network treats the shadows of humans also as humans.

IV. FORMULATION:

The formula for the calculation of accuracy of the network is
given by:
Let the number of pedestrian classes detected = N_P and
Let the number of training classes = N_T

$$\% \text{ accuracy} = \frac{\text{Number of pedestrian classes detected}}{\text{Number of training classes}} X\ 100$$

(OR)

$$\% \text{ accuracy} = \frac{N_P}{N_T} X\ 100$$

CASE I: Number of pedestrian classes detected = 181
Number of training classes = 202

$$\% \text{ accuracy} = \frac{181}{202} X\ 100$$

≈ 89%

This is same as the accuracy as the first statistic obtained from the observation table.

CASE II: Number of pedestrian classes detected = 139
Number of training classes = 202

$$\% \text{ accuracy} = \frac{139}{202} X\ 100$$

≈ 69%

This is same as the accuracy as the second statistic obtained from the observation table.

CASE III: Number of pedestrian classes detected = 80
Number of training classes = 202

$$\% \text{ accuracy} = \frac{80}{202} X\ 100$$

≈ 40%

This is same as the accuracy as the third statistic obtained from the observation table. We select the best accuracy obtained out of these three statistics calculations which is 89%.

V. CONCLUSION

From the investigation of the considerable number of papers we have reached the resolution that in these papers we proposed and assessed the utilization of ACF (Aggregate Channel Features) with convolutional neural system (CNN) as the classifier continuously passerby identification framework. In this way, the most significant specialized highlights that are investigated from all these overviewed papers is that the great highlights consequently are upgraded to distinguish the assignments. What's more, by utilizing such methods as CNN, we can accelerate the identification procedure and cost of usage is less and furthermore has higher exactness in recognizing which it decreases the quantity of False-Positives results acquired during the preparation stage. Also, the ACF is a quick and productive technique utilized with the end goal of ongoing person on foot identification and this calculation permits to determine the ROI (Regions of Interest) where the people on foot can be discovered precisely. The exactness of the SVM classifier outperforms the precision of the fundamental CNN when it could utilize similar highlights, because of the utilization of huge edge preparing inalienable to the SVM. At the point when the CNN was regularized during preparing so as to get an enormous edge choice limit, its bogus positive (FP) rate dropped to half of the fundamental FP-rate, practically getting identical to the CNN highlights and SVM mix. Other than its capacity to gain proficiency with the ideal includes, another huge favorable position of the CNNis its low computational intricacy.

Despite the fact that ourusage is a long way from ideal, an opportunity to process one passerby up-and-comer with the CNN which is under 3% of the time required by the SVM. Since the convolutional neural system gives both higher exactness and needs altogether less calculation, we accept that it is a more qualified classifier for person on foot location.

REFERENCES

1. Alexey, Yolo-v3 Windows and Linux version. [Internet]. [cited 2019 April 08], Available from:https://github.com/AlexeyAB/darknet(website).
2. Pierre Duthon."Descripteursd'images pour lessystèmes devisionroutiersensituationsatmosphériquesdégradéesetcaractérisation des hydrométéores".PhD thesis, université Clermont-FerrandII.2017.
3. Tumas, P.; Jonkus, A.; Serackis, A. 'Acceleration of HOG based Pedestrian Detection in FIR Camera Video Stream'. Open conference of Electrical, Electronic and Information Sciences. 2018.
4. Bilal M, Khan A, Karim, Khan MU, Kyung C. Alow complexitypedestriandetectionframeworkforsmart video surveillance systems. IEEE Transactions on Circuits and Systems for Video technology, 2017
5. YingdongMa,XiankaiChen,LiuJin,andGeorgeChen. A Monocular Human Detection System Based on EOHand OrientedLBPFeatures,ISVC2011.
6. Kar, A., Rai, N., Sikka, K., &Sharma, G. Adascan: Adaptive scan pooling in deep convolutional neural networks for human action recognition in videos. CPVR 2017
7. Wang W, Zhang W, Guo H, et al. A safety based approaching behavioral model with variousdriving characteristics. Transportation Research Part C-Emerging Technologies, Dec 2011.
8. AnkitDilipYawale, V. B. Raskar, "Review of Pedestrian Detection by Video Processing in Automotive Night Vision System", IJESRT,2017.
9. A. Angelova, A. Krizhevsky, and V. Vanhoucke Pedestrian detection with a largefieldof-view deep network. ICRA,2015.
10. D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In ComputerVision– ECCV2014.

AUTHORS PROFILE

Dr. Sandhya N. BE, M.Tech, Ph.D, currently working as Associate professor in Department of CSE, Dayananda Sagar Academy of Technology & Management. I have 16 years of teaching experience including 10 years of research focus.I have published 15 international publications. I have been an editorial Member of Journals. My areas of interests are in the domain of Image Processing using AI and ML techniques, Data Mining and Analysis and Computer Networks and Mobile Networks. I have received awards for Best Professor in Computer Science and Engineering Studies, Dewaang Mehta National Education Leadership Awards 2019. I have received Best Paper award in 24th CSI Convention, 2010. I have Patent filed in the field of IOT. I am a life time member of ISTE.

Anirudh Marathe, a graduated student of Computer Science and Engineering from Dayananda Sagar Academy of Technology and Management on October 2020 and received B.E/B.Tech degree in Computer Science and Engineering for the academic year 2016-2020. My areas of interest are research in the domain of Artificial Intelligence and Machine Learning. I have also carried out internship in the field of Artificial Intelligence which is object detection in images using Artificial Intelligence techniques during my academic year. And I have also made two projects based on Web Technology and Computer graphics during my academic year.

JS Dawood Ahmed, a graduated student of Computer Science and Engineering from Dayananda Sagar Academy of Technology and Management on 2020 October and received B.E/B.Tech degree in Computer Science and Engineering for the academic year 2017-2020. My areas of interest are research in the domain of Artificial Intelligence, Machine Learning, Web technologies & Networking and I have also carried out internship in the field of Networking & Cyber Security which is hacking wireless devices by using various tools of Kali Linux & others. And I also made two projects based on Web Technologies and one on Computer Graphics during my academic year.

Aman Kumar Suman,a student of Department of Computer Science and Engineering, graduated from the Dayananda Sagar Academy of Technology and Management on October 2020 and received B.E/B.Tech degree in Computer Science and Engineering for the academic year 2016-2020. My area of interest for research is the domain of Artificial Intelligence and Machine Learning. I have also carried out an internship project in Full Stack Web development which is web based in a given car rental portal by using XAMPP server during myacademic year. I have also carried out a project in Web Technology and Computer Graphics during my academic year.

Harshith R, a student of Department of Computer Science and Engineering, graduated from the Dayananda Sagar Academy of Technology and Management on October 2020 and received B.E/B.Tech degree in Computer Science and Engineering for the academic year 2016-2020. My area of interest for research is the domain of Artificial Intelligence and Machine Learning. I have carried out an internship project in the field of Artificial Intelligence which is 'GRE Admission Prediction' during my academic year. I have also carried out a project in Web Technology and Computer Graphics during my academic year. Presently I am working as a Java developer.

.