Security and Surveillance using Computer Vision

Shashank Maan
M.I.E.T Meerut
Villa no 2
Saga Habitat, Meerut

Anamika Tyagi
M.I.E.T Meerut
Vill+ post- kurdi,
distt.-Saharanpur

Shubham Kumar
M.I.E.T Meerut
Village/ post -Tana
District - Shamli

Abhishek Kumar
M.I.E.T Meerut
H.No 749
Vill+Post- Jaula
Tehsil- Budhana
Distt.- Muzaffarnagar

ABSTRACT

Object detection and Action recognition is one of the key problems in computer vision. Deep learning has gained an enormous influence on how the world is adapting to Artificial Intelligence since the past few years.The project is mainly made for the purpose of security and surveillance. Action recognition is used to monitor the movements of the subject and object detection is used to detect the objects present in the scene.This project is created in the Python programming language with the help of real time computer vision libraries like OpenCv and YOLO. OpenCv is also used for machine learning, and image processing. We have given a dataset of objects which can detect up to 82 objects. OpenCv uses its dnn module to load pre-trained dataset in the code and match it with the data provided by the numpy matrices. Numpy is also a python library used to store the data in arrays. It works by taking the input as a video stream from the webcam/any cam and applying the algorithm on each frame.

Keywords

Computer vision, Convolutional neural network (CNN models), Objection Detection, Action recognition

1. INTRODUCTION

There are several issues in computer vision that were saturating on their accuracy before a decade. However, with the increase of deep learning techniques, the accuracy of those issues drastically improved. one in all the most important issues was that of image classification and activity recognition, which is dened as predicting the category of the image and gathering the information from the frame. a rather sophisticated downside is that of image localization, wherever the image contains one object and also the system ought to predict the category of the placement of the item within the image (a bounding box round the object). The additional sophisticated downside (this project), of object detection involves each classification and localization. Action recognition and action prediction are 2 additional sophisticated issues within the field of laptop vision. Action recognition is the recognition of human activity from a video containing complete action execution and also the Action prediction is the reason an individual's action from temporally incomplete video information. This project works on Objection Detection and Action Recognition. During this case, the input to the system is a true time video, and also the output is a bounding box adoring all the objects and also the action within the image, beside the category of objects and actions in every box.

Image is assessed from the input frame by grouping the pixels from lower level upto higher level till a picture is assessed and detected as shown in fig.(a) and that we will acknowledge the action by verificatory and connecting numerous links obtained by the rule from the input frame as shown in fig(b).When we point out Object Detection and Action Recognition primarily based upon Deep Learning , we tend to stumbled on 3 primary object detectors:-

1) R-CNN and its variants, which incorporates original R-CNN , Fast R-CNN , quicker R-CNN.
2) Single Shot Detector (SSDs).
3) YOLO(You Look Only Once).

R-CNNs are the primary Object Detectors primarily based upon deep learning associate degreed ar an example of a two-stage detector . It needs algorithms like selective search.

Fast R-CNN created terribly several enhancements to the initial R-CNN by increasing accuracy and reducing the time it took. quicker R-CNN became a far better Deep Learning primarily based object detector by removing the need of algorithms like selective search . It uses Region projected Network (RPN) that's absolutely convolutional and may predict the item bounding boxes . The outputs of the RPN are then passed to the R-CNN for final classification and labeling.

The R-CNNs are terribly correct however the most downside with them is their speed that is improbably slow - getting solely five independent agencies on a GPU . For resolution this downside of speed YOLO is employed , that treats Object Detection as a regression downside by taking a given image as input and learning bounding box coordinates adore category label possibilities at the same time. YOLO is one in all the foremost effective Object Detection algorithms that hold at intervals several of the foremost innovative concepts of laptop Vision

One of the classical issues of laptop vision is Object Detection wherever we tend to be needed to acknowledge what and wherever i.e. what objects are there within the given image and conjointly wherever they reside within the image . Object Detection downside is additional complicated than the classification downside that acknowledges what the item is however not wherever it resides within the given image. YOLO uses a very completely different approach of clever Convolutional Neural Network (CNN) for object detection in a time period . This rule uses one neural network for the total image . YOLO is additionally well-liked as a result of it's ready to offer high accuracy in time period moreover . It needs only 1 forward propagation pass through the neural network to form predictions. Then it shows the output as recognized objects along, beside the bounding boxes . It optimizes the item Detection performance.

{when we tend to|once we|after we} {try to|attempt to|try associate degreed} build our deep learning primarily based time period object detector with the assistance of OpenCV we need to: 1) Access our net cam / video stream in an economical manner and 2) Apply Object Detection to every frame. OpenCV library supports some models from deep learning that are TensorFlow , Torch , PyTorch and Caffe in keeping with the outlined list of supported layers. a number of the applications of the OpenCV are : 2D and 3D feature toolkits , identity verification System , Motion Understanding , Object Identification , Segmentation , Motion pursuit and additional

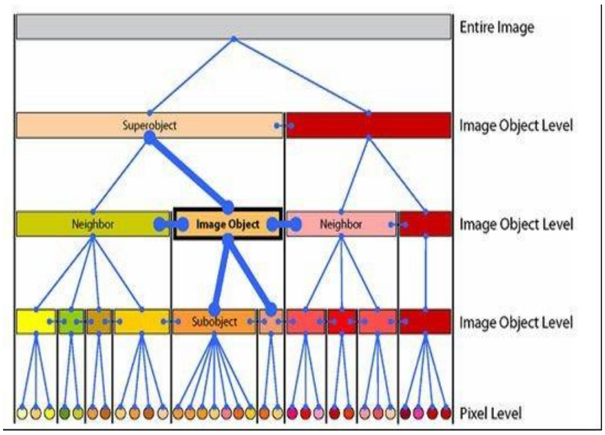


Fig.1 .A general representation of abstract features using pixel. The bottom level consists of raw pixels. As we go up in the hierarchy, pixels are grouped together to form low-level features, which, in turn, form groups to form high-level features.

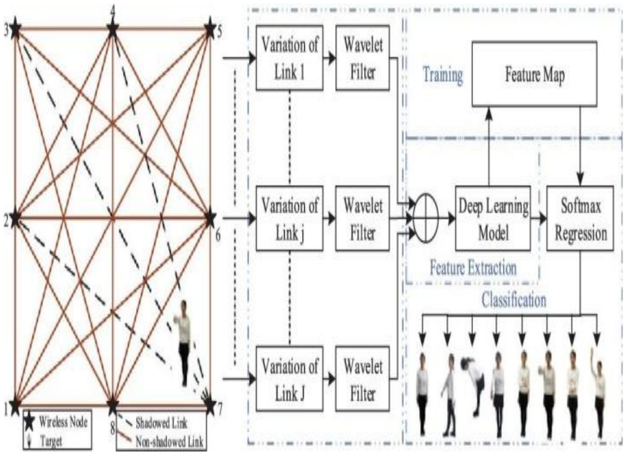


Fig.2 Action Recognition Neural network

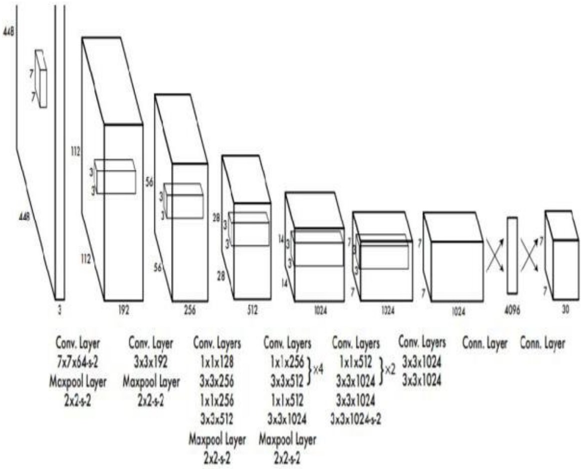


Fig.3 Layer wise presentation

2. WORKING OF NETWORKS

Working of networks based mainly on Convolutional Neural Networks and following points are described below 2.1)Theory of the CNN networks in a brief manner and working of the CNN on the basis of its Layers.

2.1 Theory of CNN :

One of the most popular algorithms of deep learning at present is the Neural Network . It has been proven over time that the Neural network performs better in terms of speed and accuracy as compared to other algorithms . They have various variants like CNN , RNN , AutoEncoders etc.

What is a neuron?

Neural Networks are inspired by the neural architecture of the human brain where the basic building block is known as neuron . Its functionality is similar to the biological neuron which takes some input and fires some output. In purely mathematical terms, Neuron is like a mathematical function which takes some input and applies a function on it to give the output . In neurons the function is known as the activation function. For understanding the neural networks we are required to understand what is a layer in a neural network. A layer is a collection of neurons which takes input and provides an output with the help of the activation function.

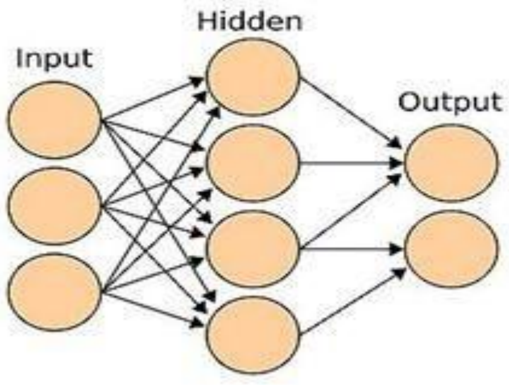


Fig .4 Simple Network Architecture

The left most layer of the network is the input layer and the rightmost layer of the network is the output layer . The middle layer is known as the hidden layer because the values of this layer are not seen in the training set. Every Neural Network consists of 1 input layer and one output layer. The number of hidden layers varies in different neural networks based upon the complexity of the problem to be solved

Each middle layer can have different activation functions based upon the type of the problem.Convolutional neural networks(CNN) is one of the variants of the neural networks which is used in the field of computer vision . The CNN consists of the following hidden layers : convolutional layers, pooling layers , fully connected layers and normalization layers

Instead of using activation function convolution and pooling functions are used as activation functions in CNN.

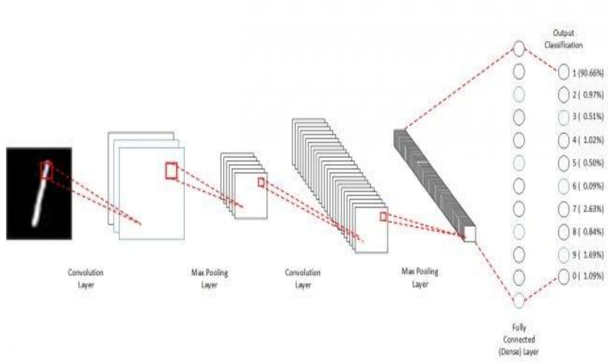


Fig .5 Image classification using CNN

Convolution : It takes two inputs one is the image and the other one is the filter and provides us with an output image . In layman terms it takes in an input signal and applies the filter on it which means multiplying the input signal with the kernel to get the modified signal i.e. mathematically it provides us with dot(.) product of the input function and the kernel function.

Pooling : It is a sample-based discretization process . it is used to dowm-sample an input by reducing its dimensionality and allowing for assumptions to be made about features contained in the sub region. Two types of pooling are max and min pooling.

3. EXPERIMENTAL PROCEDURE

3.1 Preparing Database

The input is given as a frame from the real time video itself. The frames are then converted to gray scale as data information is important for the network and data matching and not the color information. Also, the images are resized to 32x32. Since the images from data sets are larger, pyramid reduction is done to make them 32x32 in size and to reduce the searching time. The image pyramid is a data structure designed to support efficient scaled convolution through reduced image representation. It consists of a sequence of copies of an original image in which both sample density and resolution are decreased in regular steps.

3.2 Training and Testing

We pretrain our convolutional layers on the ImageNet 1000-class competition dataset. For pretraining we use the first 20 convolutional layers from Figure © followed by an average-pooling layer and a fully connected layer. We train this network for accuracy of 88% . We use the Darknet framework to train the dataset. We then converted the model to perform detection to show that adding both convolutional and connected layers to pretrained networks can improve performance. Our final layer predicts both class probabilities and bounding box coordinates. We normalize the bounding box width and height by the image width and height so that they fall between 0 and

1. We parametrize the bounding box x and y coordinates to be offsets of a particular grid cell location so they are also bounded between 0 and 1. We use a linear activation function for the final layer

$$\varphi(x) = \{ x, \text{if } x > 0 \}$$
$$\{ 0.1x, \text{otherwise} \}$$

Sum-squared error is also optimized in the output of our model. We use sum-squared error because it is easy to optimize. In every image many grid cells(group of pixels) do not contain any object. This reduces the confidence of the boxes. This can lead to model instability, causing training to disunite.. Sum-squared method's error also equally weights error in large boxes and /small boxes. Error metric of the dataset reflects that small deviations in large boxes matter less than deviations in small boxes. To partially address this we predict the square root of the bounding box width and height instead of the width and height directly. During training we optimize the following, multi-part loss function:

$$\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$
$$+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \sum^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum (p_i(c) - \hat{p}_i(c))^2$$

where 1 obj i denotes if object appears in cell i and 1 obj ij denotes that the jth bounding box predictor in cell i is responsible for that prediction. Note that the loss function only penalizes the classification error if an object is present in that grid cell. We train the network for about 135 iterations on the training and validation data sets from PASCAL VOC 2007 and 2012. Throughout training we use a batch size of 64, a momentum of 0.9 and a decay of 0.0005. Our learning rate schedule is as follows: For the first iteration we slowly raise the learning rate from 10−3 to 10−2 . If we start at a high learning rate our model often departs due to unstable gradients. We continue training with 10−2 for 75 iterations , then 10−3 for 30 iterations, and finally 10−4 for 30 iterations. YOLO is fast at test time since it only requires a single network evaluation, unlike classifier-based methods. Often it is clear that in which grid cell an object falls into and the network only predicts one box for each object.

4. PROPOSED MODEL OF HUMAN ACTION DETECTION

The main objective of this approach is to detect the actions of multiple individuals in real-time for surveillance applications. The human actions are detected by a frame-based human detector. The action classifier is composed of three CNNs model networks that operate on the shape, motion history and their combined cues. According to the subaction descriptor, the classifier predicts the regions to produce three outputs for each action. The outputs of the classifiers go through a post-processing step to render the final decisions and detect the action.

4.1 Tracking by Detection

The main goal of this project and of this paper is real-time action detection in surveillance video. For human action detection, we adopt some existing methods to provide a stable human action region for subsequent action recognition. A processing time of 20-30 ms for each frame, a stable bounding box for the human action region, and a low false detection rate are the important factors for human detection and tracking.

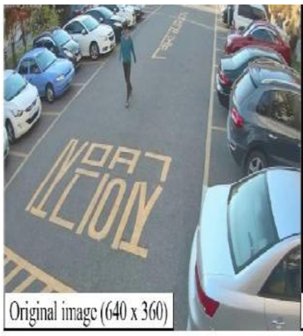


Fig 6.1 **Fig 6.2**

The size of mini motion map is computed with the following equation:

$$\text{size}_{\text{mni-map}} = \frac{\text{size}_{\text{original}} - \text{size}_{\text{detection}}}{\text{stride}}$$

And the default value of size detection is (64,128).

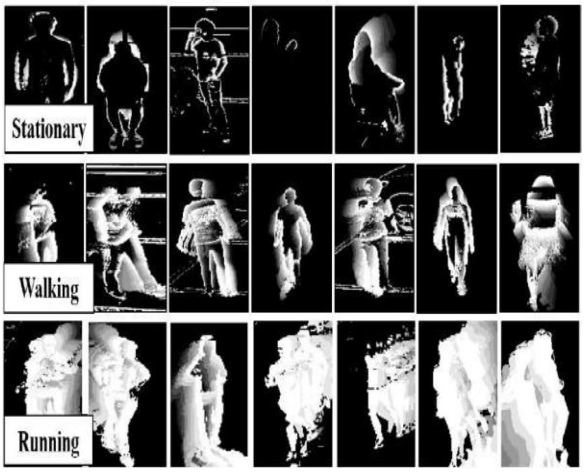


Fig.7 Examples of different sub actions

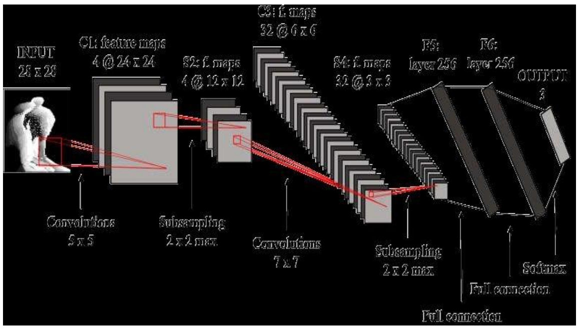


Fig . 8 Architecture of a CNN. The architecture comprises two convolutional layers, two subsampling layers, two fully connected layers, and one softmax regression layer.

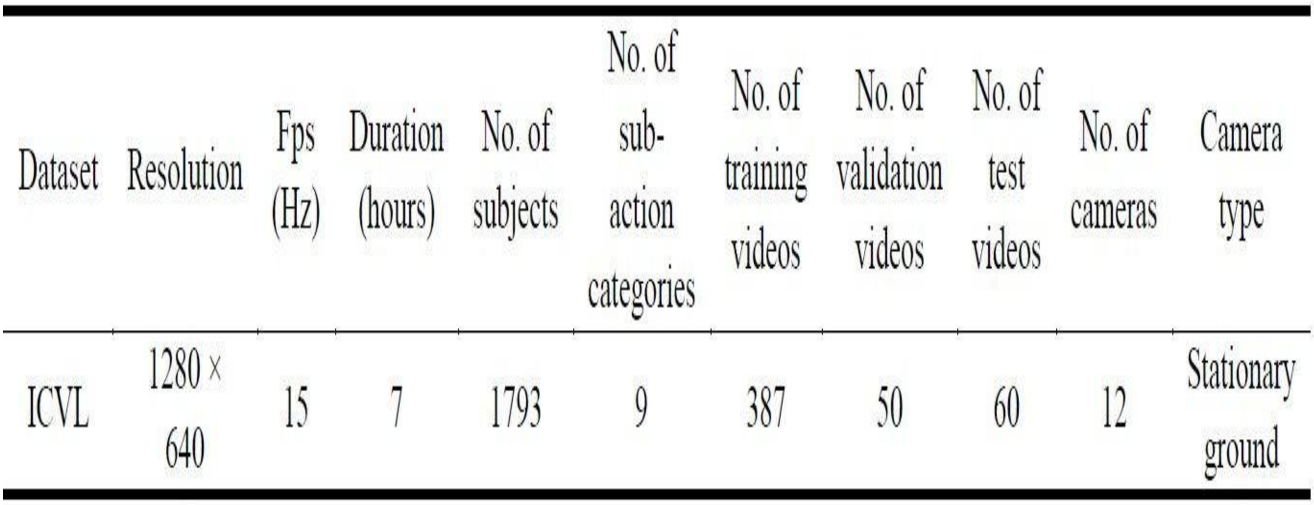
5. EXPERIMENTAL RESULT

In this section, the proposed approach for real-time action recognition in surveillance In this section, the proposed approach for real-time action recognition in surveillance videos is evaluated in terms of the recognition rate and processing time. The average processing time was computed as the average time required to localize and recognize an actor in a test video.

5.1 Action Detectionon ICVL Dataset

One important characteristic of the ICVL dataset is that it includes different sub-actions of multiple individuals occurring simultaneously at multiple space locations in the same scene. There are diverse types of sub-action categories on the ICVL dataset: sitting, standing, stationary, walking, running, nothing, texting, smoking, and others. In terms of annotation, sub-action annotations are provided for each action, e.g A person is marked simultaneously by three sub-actions: standing, walking, and smoking. The ICVL dataset collected approximately 7 h of ground-based videos across 12 non-overlapping indoor and outdoor scenes, with a video resolution of 1280 × 640 at 15 Hz. Snapshots of 12 scenes. One of the fundamental issues in sub-action labeling is deciding the starting moment and ending moment of an action. For instance, the action standing up has a posture conversion from sitting to standing.

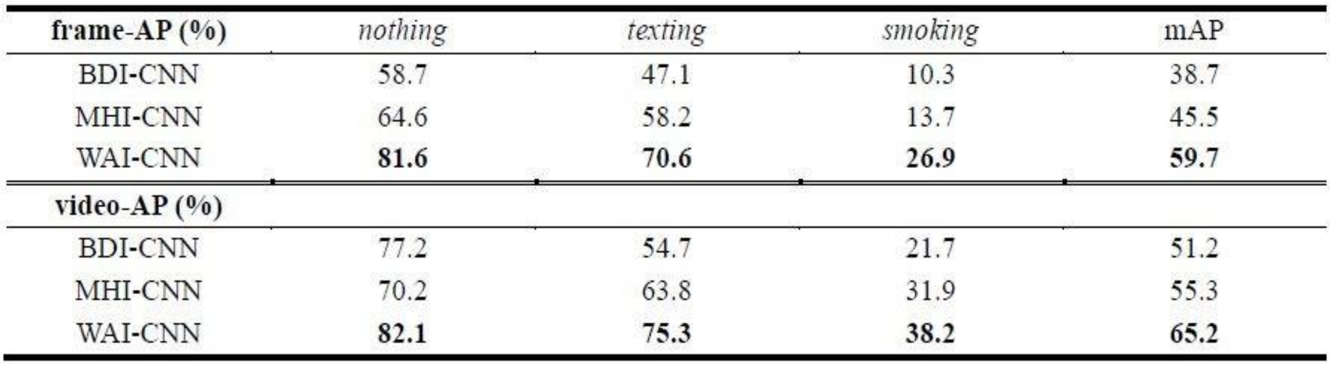
Tabel 1. Statistics of the ICVL dataset. The dataset focuses on action detection in video surveillance and was collected in real-world environments.



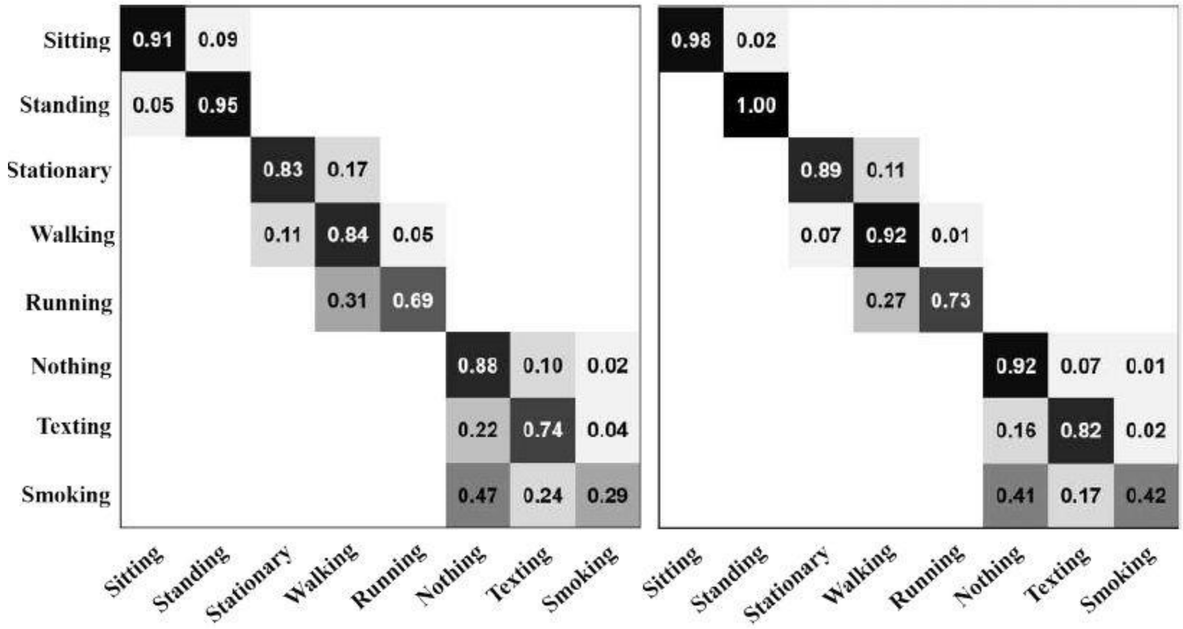
Dataset	Resolution	Fps (Hz)	Duration (hours)	No. of subjects	No. of sub-action categories	No. of training videos	No. of validation videos	No. of test videos	No. of cameras	Camera type
ICVL	1280 × 640	15	7	1793	9	387	50	60	12	Stationary ground

The statistics of the dataset are summarized in Table 1. The ICVL dataset consists of 387 training videos, 50 validation videos, and 60 test videos (5 videos for each camera in validation and test set).

Table 2: Results of the ablation study on the gesture level of ICVL dataset. Frame-AP and video-AP are reported for BDI-CNN, MHI-CNN, and WAI-CNN. WAI-CNN performed significantly better under both metrics, showing the significance of the combined cues for the task of gesture-level sub-action recognition. The leading scores of each label are displayed in bold font.



frame-AP (%)	*nothing*	*texting*	*smoking*	mAP
BDI-CNN	58.7	47.1	10.3	38.7
MHI-CNN	64.6	58.2	13.7	45.5
WAI-CNN	**81.6**	**70.6**	**26.9**	**59.7**
video-AP (%)				
BDI-CNN	77.2	54.7	21.7	51.2
MHI-CNN	70.2	63.8	31.9	55.3
WAI-CNN	**82.1**	**75.3**	**38.2**	**65.2**

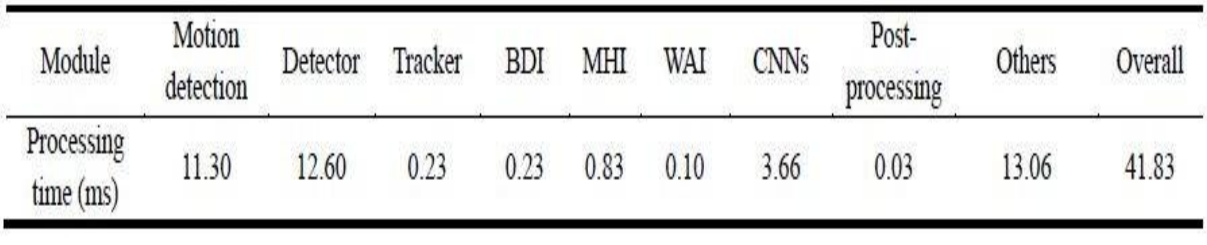


	Sitting	Standing	Stationary	Walking	Running	Nothing	Texting	Smoking
Sitting	0.91	0.09						
Standing	0.05	0.95						
Stationary			0.83	0.17				
Walking			0.11	0.84	0.05			
Running				0.31	0.69			
Nothing						0.88	0.10	0.02
Texting						0.22	0.74	0.04
Smoking						0.47	0.24	0.29

	Sitting	Standing	Stationary	Walking	Running	Nothing	Texting	Smoking
Sitting	0.98	0.02						
Standing		1.00						
Stationary			0.89	0.11				
Walking			0.07	0.92	0.01			
Running				0.27	0.73			
Nothing						0.92	0.07	0.01
Texting						0.16	0.82	0.02
Smoking						0.41	0.17	0.42

Fig 9. Confusion matrices of the ICVL dataset at the frame-based and video-based measurement for the action-detection task when using appearance-based temporal features with a multi-CNN classifier. The horizontal rows are the ground truth, and the vertical columns are the predictions. Each row was normalized to a sum of 1. (a) The confusion matrix at the frame-based measurement, and (b) the confusion matrix at the video-based measurement

Table 3. Average processing time of the proposed action detection model



Module	Motion detection	Detector	Tracker	BDI	MHI	WAI	CNNs	Post-processing	Others	Overall
Processing time (ms)	11.30	12.60	0.23	0.23	0.83	0.10	3.66	0.03	13.06	41.83

6. CONCLUSION AND DISCUSSION

In this paper we provided an overall view of how we can use AI and Computer Vision in different fields like robotics , security and surveillance , driverless cars etc .

This paper represented a new approach to real time action detection and object detection in video surveillance systems. The given dataset in Object Detection is capable to detect the objects with higher accuracy but in low fps and the dataset provided in action recognition is capable to detect sub action to group into and identify the accurate activity of the subject. The given algorithm requires a good gpu to run and to achieve smooth fps while detection.As it uses a single shot detector it boosts its activity recognition algorithm to achieve higher frames per second. In future work we plan to place a gps in the algorithm to increase its security and to easily communicate with the user in emergency as well. The technology we used is also very interesting in recent decades due to its various applications in emerging fields like machine learning , deep learning , robotics etc.

7. REFERENCES

[1] Li Deng and Dong Yu "Deep Learning : methods and applications" by Microsoft research [Online] available at: http://research.microsoft.com/pubs/209355/N OW-Book-Revised-Feb2014-0nline.pdf

[2] McCulloch, Warren , Walter pitts,"A Logical Calculus of Ideas Immanent in Nervous Activity", Bulletin of Mathematical biophysics 5(4):115-133(1943)

[3] An introduction to convolutional neural networks [Online]available\at:http://white.standford.edu/tech/index.php/An_I ntroduction_to_Convolutional_Neural_Networ ks

[4] http://github.com/opencv/opencv/wiki/Deep-L eerning-in-OpenCV

[5] Adrian Rosebrock,"Object detection with deep learning and opencv",pyimagesearch

[6] Akshay Mangawati , Mohana , Mohammed Leesan , H. V. Ravish Aradhya, "Object Tracking Algorithms for video surveillance applications"International Conferenceon communication and signal processing (ICCSP),India,2018,pp.0676-0680 .

[7] Mohana and H. V. R . Aradhya, "Elegant and efficient algorithms for real time object detection, counting and classification for video surveillance applications from single fixed camera," 2016 International Conference on Circuits, Controls,Communications and Computing (I4C), Bangalore, 2016, pp. 1-7.

[8] Apoorva Raghunandan, Mohana, Pakala Raghav and H. V. Ravish Aradhya, “Object Detection Algorithms for video surveillance applications” International conference on communication and signal processing (ICCSP), India, 2018, pp. 0570-0575.

[9] Manjunath Jogin, Mohana, “Feature extraction using Convolution Neural Networks (CNN) and Deep Learning” 2018 IEEE International Conference On Recent Trends In Electronics Information Communication Technology,(RTEICT) 2018, India.

[10] Arka Prava Jana, Abhiraj Biswas, Mohana, “YOLO based Detection and Classification of Objects in video records” 2018 IEEE International Conference On Recent Trends In Electronics Information Communication Technology,(RTEICT) 2018, India.