INTRODUCTION

Object recognition is one the critical challenge that majority of the computer vision research is focused on. Mainly, object detection and tracking is very popular with growing demands in surveillance, defence front, traffic control and medical imaging. It is also a result of the powerful algorithms in sub-fields of machine learning, computer vision as well as due to innovations in hardware that support highly data intensive computations in magnitude of couple of minutes. The ultimate purpose underlying vision based object detection is to comprehend the kind of objects in the image, the characteristics of those, their location in space and additionally involve motion or trajectory of the object while moving. Of the many objects commonly considered for detection, human as an object has got a lot of attention and produced many applications.

The more general application of multiple object detection that may include targets like chair, car, tree i.e. non-living objects in addition to humans or animals is more complex one. This task has the challenges of accurately detecting the object categories as a result of inherent difficulties of multiple objects, varying size in the scene, overlapping of objects and deformities in the target object that vary across images. To overcome this problem when attempting a large detection mission of objects spanning a number of classes, the cue that helps is contextual recognition.

Public benchmark datasets for object detection and classification available Online, involve millions of images with objects belonging to thousands of different categories. Some of those datasets are ImageNet database, Microsoft COCO database and Pascal VOC database. The datasets have annotations made already to each object present. The object classification datasets may have one or more of the three different kinds of images- iconic target dataset, which contain target object visual occupying the central part without other background (context); non-iconic dataset, which contain object instances along with contextual elements and humans; last, iconic scene dataset that involve various objects present in the frame either belonging to the target class or not but absence of humans. Though iconic image types seem to be easier to detect and also track in case the video contains only iconic images of one specific target, it is the non-iconic image data are good candidates for generalization containing contextual information. Having context data in images makes the object recognition algorithm to learn well and output good classification rate when it is tested against a new image dataset.
containing many generic naturally captured clicks containing various objects.

The problem of detection and tracking of objects from video data has been implemented using many approaches. The main challenges involved are accurate detection of object in occlusions, ambiguous background object movement and real-time tracking without comprising quality of output produced. There is much scope in detection of generic objects from videos. In order to detect an object accurately, the object features must be extracted and matched (feature extraction) or the object structure must be learnt automatically through training (machine learning).

Many works have exploited different feature set to achieve good detection performance. Few of the researches involve tracking of objects by means of detection, wherein the mobile object is continuously detected in frames and labelled. One such algorithm makes use of colour histogram probability and centroid features for object detection [1]. Optical flow motion vectors[2] of moving objects like vehicle, pedestrian have been chosen to detect objects, both living and non-living while in movement. This technique works based on motion vector comparison between two successive frames corresponding to the same object to find the object movement assuming intensity of object remains fixed.

Texture features like Local Binary Pattern(LBP)[6] of the image have been considered for feature extraction for detecting objects. LBP features are computed by means of pixel neighbourhood operations. A popular feature descriptor for object detection involves Histogram of Oriented Gradients. HOG features are shape descriptors that represent an object in terms of intensity gradients in specific directions. In [7], the authors have exploited HOG features citing their properties of invariance with respect to transformations like rotation, deformities as well as lighting conditions. In another work [8], Gabor wavelet has been utilized. It produces magnitude responses of high frequency parts of a binary image. To detect multiple objects of same class but varying in colour, Gabor wavelet is estimated on binarized images.

In this work, we have used the corner features that are most suitable to be detected in videos using Shi-Tomasi detector and Lucas-Kanade algorithm. For object class recognition, multi-class Support Vector Machine is utilized.

**Related Work**

This section briefly describes the existing systems for object detection and tracking in videos and the techniques used by their authors for the same.

Mingjie Liu et. al[1] have proposed a framework for tracking moving objects in videos based on confidence score computation and data association. Tracking-by-detection approach is adopted for detection of individual moving object in video with formation of tracks by integration of detections in successive frames. Detection is carried out based on extraction of three features namely location, shape and appearance. ResNet-50, a pre-trained Convolutional Neural Network for obtaining object appearance features was implemented. An ensemble model integrating all three features for affinity computation, a measure representing the degree of accurate association of object detections with tracklets is used.

Hungarian algorithm for data association was implemented by the authors for first time track generation. The tracking was further improved by using confidence score computed to identify wrong associations of object and tracklet, with hierarchical feature updates associated to unconfident tracklets if the score is below certain threshold. The system performance is Stefan Duffner et. al [9] have presented a novel algorithm for the purpose of non-rigid object tracking. The algorithm works on pixel level granularity making use of Hough transform object detector. The detections are made by building a pixel model using color and gradient features adaptive to appearance variations. At the same time, a Bayesian probability segmentation technique was executed for extracting the foreground segment of the video frame. A shape model was also proposed to refine the roughly obtained foreground object to improve the tracking, using canonical shape reconstruction from segmentation map. Tracking was performed at two levels-first segmentation map tracking involving location of the centre and integration of position estimate obtained by the correlation of shape model with the segmentation map.

Mark Heimbach et. al[4] have used Kalman filter technique to achieve tracking of moving objects in videos in presence of noise and occlusions. The authors have primarily employed the Histogram of Oriented Gradients (HOG) as the object descriptor, due to the detections unaffected by rotational motion, deformities as well as illumination. The Kalman filter was chosen for tracking the objects using the location and speed as the object state. The noise and occluded object locations have been considered to rectify the tracks generated by the Kalman filter using comparison between HOG features and Kalman filter errors. The track updation to account for correct object location when occluded

Lychkov et.al [4] proposed a novel object tracking system that addresses the issue of losing track of object’s feature points. The authors use the concept of regeneration in living beings to detection of objects in video. The sparse optical flow features of the object in motion are extracted in order to detect the movement. The tracking is based on matching object location at present instant with the trajectory being generated as per observed relative displacement. The epimorphosis based regeneration technique of object’s feature points is exploited with elimination of feature points less than user defined threshold and new feature point replacement. The work includes the affine transform to adapt the tracking process with scaling as well as rotation.

Swalaganata et.al [5] has used a hybrid method for tracking of moving objects from video inputs, which is combination of Camshift algorithm and Kalman filter. The tracking is based on detection per frame of the video. The bounded box is used to mark the target object to be tracked as part of detection process. The first tracking technique is Camshift based on colour histogram probability and centroid establishment with Kalman filter as an enhancement to it. Since the occlusions in the frames are not handled by Camshift method, the authors thus rely on improving tracking by prediction using Kalman filter. Runze et.al [6] constructs a new framework for tracking of mobile objects called Tracking Learning Detection framework. Tracking module is used for frame-wise tracking of object movement using optical flow cues and the detection module
works by using ensemble and nearest neighbour classifier for recognition based on texture features. The cross correlation concept in the detection module for matching the object model with target in the video is also employed. Track updating with consecutive detections of the object in consecutive frames is done by the learning module.

Jaganatthan et al [7] proposed a framework for real-world 3D objects detection based on Histogram of Oriented Gradients (HoG) features and subsequently mapping them into respective object classes (classification) for use in energy efficient embedded devices. The dataset included three object categories of ‘pedestrian’, ‘vehicle’ and ‘traffic signal’. HoG features relating to each object are extracted from feature planes of the image constructed. AdaBoost cascaded tree classifier along with meanshift algorithm is used for classification. In addition, second classifier i.e. 7 layer CNN is also implemented. The classification accuracy is tested and compared in both classifiers used. The CNN classifier outperformed the AdaBoost classifier.

PROPOSED METHODOLOGY

The objective of our work is to implement an object detection system which is well trained to detect multiple generic objects in video frames and can generalize the detection in presence of occlusions in real time based on good feature descriptors and classifier.

System Architecture

The proposed technique adopts the feature extraction technique used in [2], which consists of two methods namely Shi-Tomasi algorithm and Lucas-Kanade method. The classification is achieved using multi-class Support Vector Machine (SVM). Fig. 1 describes the system architecture of system architecture of our generic object detection and classification system.

The Shi-Tomasi algorithm detects objects based on extracting corner features [9]. Lucas-Kanade method [10] is used as it is based on object detection using optical flow features, suitable for detecting static objects as well as tracking moving objects in videos. Following is description of the proposed methodology:

Object Detection and Classification System

Step 1. Read the input video containing multiple objects including moving and static for detection.

Step 2. Pre-processing: Pre-process the input video by performing the two steps below:

1. Extract and store individually every frame of the video from the beginning.
2. Apply morphological filtering operation to remove noise and refine the frame.

Step 3. Feature Extraction: Extraction of distinct object features to compute the feature vector for detection in successive frames of the video.

Image Pre-Processing and Annotation

Image Pre-Processing involves processing or cleaning of images. This step focuses on removal of noise and distortion, sharpening, intensity normalization, etc. The VOC dataset is refined with only person images and annotated according to the format of YOLO model. A text file is created for each image in the same directory with the same name that contains object number and object coordinates on this image, for each object in new line. The object number is an integer number of object from zero to total number of classes – 1, and object coordinates are float values relative to width and height of image, it can be equal from (0.0 to 1.0]. The ID card images are only pre-processed.

Training and Testing YOLO model

After pre-processing and annotation, the person dataset is divided into training and testing datasets. We train the YOLO model using training dataset until we get a better mean Average Precision (mAP). After the training, it is tested with testing dataset. YOLO is a full convolutional network consisting built using darknet-53. It detects objects at three different strides (8, 16 and 32) which helps to detect smaller objects. The input image is divided into S*S uniform grid, and each cell is composed of (x, y, w, h) and confidence C (Object). The coordinates (x, y) represent the position of the center of the detection boundary box relative to the grid. (w, h) is the width and height of the detection boundary box. Each grid predicts the probability of C categories. The confidence score is the probability of the model to include the target object and the accuracy of the prediction detection box. Pr(Object) stands for whether there is a target object falling into this cell. If there is confidence, it is defined as:

$$C(\text{Object}) = \text{Pr}(\text{Object}) \times \text{IOU(Pred, Truth)}$$

If the cell does not have a object, the confidence score is zero C (Object) = 0. IOU is the overlapping rate of the generated candidate bound and ground truth bound, that is, the ratio of their intersection and union.

Step 4. Classification: To classify the detected objects into their respective categories with the help of a classifier. Here, multi-class SVM classifier is used to output object class by supervised learning. This involves two steps-

1. Training the CNN with feature vectors and target classes of objects detected from training video dataset.
2. Testing the CNN classification performance over videos from testing video dataset.

The output of the classification is the class that a detected object belongs to. In case of multiple objects of different classes in a single frame, a vector containing the classes of all the detected in that frame is produced as output.
Feature Extraction for Recognition of Objects

This section describes in detail the theoretical as well as mathematical background related to the techniques used for extracting the features and object recognition based on those features in our system.

Shi-Tomasi Corner Detector

It is an improvised version of Harris corner detector, which extracts corner features from images that are the regions of high texture variations with small changes in position. A brief description of Harris corner detection technique follows-

The detection is done by scanning the whole image through equal sized small windows. If $I(x,y)$ is the intensity of a pixel located at position $(x,y)$ in an image window and when the window is moved slightly by a displacement $(u,v)$ then, the intensity change is denoted as $I(x+u, y+v)$. Since the aim of this corner detector is to find out regions or windows in the image whose intensity changes greatly with small displacements, it is mathematically expressed as-

$$E(u,v) = \sum_{x,y} w(x,y) I(x+u, y+v) - I(x,y)^2$$  .... (1)

Where ‘$w$’ is a weight function. The result $E(u,v)$ produces the high intensity variation windows in the image.

After using Taylor’s series and simplification of Equation 1, the following is the result

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$  .... (2)

A score based on Eigen values of matrix $M$ is used to find out suitable corners.

$$S = |M| - k(\text{trace}(M))^2$$  .... (3)

If $\lambda_1, \lambda_2$ are Eigen values of $M$, then $|M| = \lambda_1 \lambda_2$ and trace $(M) = \lambda_1 + \lambda_2$.

A high score $S$ represents one of the corners of the object in the image.

A small change in score calculation is accommodated in Shi-Tomasi Corner Detector, which leads to finding out only the most suitable ‘n’ corners of the object rather than extracting every corner feature. Shi-Tomasi score for corner detection follows-

$$S = \min(\lambda_1, \lambda_2)$$  ....... (4)

If the score, $S$ exceeds a threshold it is considered as a corner [9].

Lucas-Kanade Optical Flow

The estimation of optical flow in an image is carried out to track the motion of objects in an image when either the objects themselves are motion or if the camera is in motion. The optical flow is apparent movement of any entity from one frame to another in a video. Lucas-Kanade algorithm[10] works on the basis of assumptions of optical flow theory. Throughout the video frames, the intensity of any pixel belonging to same object remains unchanged, the pixels in a neighbourhood have similar motion and the displacements of pixels between successive frames is very small.

A pixel with intensity $I(x,y,t)$ in a frame at time $t$ after moving with a small displacement $(dx, dy)$ in consecutive frame with a difference of time $dt$ is expressed as follows-

$$I(x,y,t) = I(x + dx, y + dy, t + dt)$$  .... (5)

The optical flow equation for movement of objects in an image is given as-

$$\frac{\partial I}{\partial t} + dx \frac{\partial I}{\partial x} + dy \frac{\partial I}{\partial y} + f_x = 0$$  .... (6)

Lucas-Kanade method outputs optical flow motion vectors by solving Equation (6):

$$\begin{bmatrix} u \\ v \end{bmatrix} = \sum_{xf} \begin{bmatrix} f_x I_x & f_x I_y \\ f_y I_x & f_y I_y \end{bmatrix}^{-1} \begin{bmatrix} -f_x f_t I_x \\ -f_y f_t I_y \end{bmatrix}$$  .... (7)

Where $(u, v)$ represent the displacement of the object between consecutive frames.

Classification of the Recognized Objects

The last step in our system is the classification of detected objects into respective categories or classes. Multi-class Support Vector Machine classifier is used to perform this classification.

Support Vector Machine is a machine learning, supervised learning algorithm. It relies on training itself with a number of sample inputs in the form of a n vector $(x_1, x_2, x_3, x_4, ..., x_n)$ along with the target class vector of same size $(c_1, c_2, c_3, c_4, ..., c_n)$ specified in order to learn the pattern involved in the data. In testing phase, the classifier attempts to generate the target class of the unobserved test input. The SVM constructs a hyper-plane with all inputs that are basically the feature vectors against their classes specified so that the best fit including all...
the input-output points is produced. The output is generated by locating the position of test feature vector onto the hyper-plane giving the target class prediction.

RESULTS AND DISCUSSION

The system is implemented in Matlab Tool, which facilitates application of complex mathematical computations through easy built-in functions for performing object detection and classification using its Computer Vision and Machine Learning toolboxes.

The video processing is smoothly done by operating on individual frames after their extraction. The Image Processing toolbox containing a lot of filtering and refining functions needed for pre-processing of an image/frame. A total of 10 videos are collected containing generic objects of different classes. The dataset is divided into two categories- training dataset of 10 videos and testing dataset of 8 videos required for support vector machine classification.

The below figure 2 describes histogram of the frames extracted from the video 1 where different objects are recognized after the extraction of features, in the same manner different objects detected from video 2 and video 3 are shown in figure 3 and figure 4 respectively.

The dataset contains videos where the objects to be detected can make up background or foreground of the scene like vehicles. Occlusions are also present in the input videos, involving both the scene-object and inter-object occlusions. The objects are detected based on easily tractable corner features, basically high intensity regions readily detected with small movement in the scene. The movement can be due to either mobile object or moving camera. The multi-class support vector machine is given as input the feature vectors of all objects with the respective classes in training. The classifier output obtained when the feature vector of an object in test video is given is the target class that is predicted by support vector machine.

The accuracy of the proposed technique is analyzed with total number of percentage of frames detected with object vs. the total number of frames. Here graph is generated by taking four input video from various sources with different sizes and different background, for each video we could able to get an accuracy of 89%. Where frames for each video vary from 100 to 1000 and accuracy for each video varies from 80% to 90% in recognizing the objects.

CONCLUSION

This paper presents a novel approach for Object Recognition using Lucas-Kanade technique and Support Vector Machine based Classification in Video Surveillance Systems. This paper is providing precise recognition of objects and estimation of their location from an unknown place. First the recognition of object is archived using Shi-Tomasi and Lucas-Kanade techniques, whenever the object is recognized from extracted frames of the input video the background subtraction will be applied. Then the classification of the objects into their respective categories can be achieved using support vector machine classifier by supervised learning. Finally the accuracy of detected frames of the object for the input video is measured in this work we achieved an accuracy of 89% by considering different input videos. In the future the algorithm can be applied over tracking of the objects on live surveillance systems using the proposed approach and research work will be extended to develop improve the accuracy of the proposed work, so that we can be able develop intelligent surveillance systems.

References

The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788

How to cite this article:
DOI: http://dx.doi.org/10.24327/ijrsr.2019.1010.4124

*******