International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-5, January 2021

Multiple Detection and Tracking of Multi Class Vehicles using Locality Sensitive Histogram.

Bhavya R, Geetha K S

Abstract: Multiple object detection and tracking in a cluttered background is most important in vision-based applications. In this paper, the goal is to develop a classifier that detects and tracks multiple objects thereby ensuring robustness and accuracy. Locality Sensitive Histogram feature extraction is used, which adds contributions from all the pixels in an image. These features extracted are trained using decision tree classifier which performs with an accuracy of 97%. Experimental results demonstrate the objects tracked and detected under different scale and pose variations. Evaluation and comparison of the proposed method with various other techniques is performed using performance parameters. Results depict that the proposed technique outperforms with increased accuracy and is the top performer.

Keywords: Decision tree, Detection, Locality Sensitive Histogram, Tracking

I. INTRODUCTION

Multiple object detection and tracking still pose harder challenges as it is required to estimate and track objects in every frame with cluttered background, changing appearance and abrupt motion. Despite the demonstrated success of various other algorithms, an image analysis tool called, Histogram is widely used in many applications. The tracking algorithms represent the target objects with various features in particular feature space to address the challenging factors [4]. While few algorithms operate as binary classification problem, other group consider a different feature space to represent target object [2], [3]. In this paper, the multiple object detection and tracking is performed by using locality sensitive histogram. Locality Sensitive Histogram is an effective method which estimates the histogram by considering the contributions from all the pixels in an image [1]. It functions much similar to the conventional way of histogram calculation, which uses only the local neighborhood pixels.

Manuscript received on January 20, 2021.
Revised Manuscript received on January 26, 2021.
Manuscript published on January 30, 2021.
* Correspondence Author
Bhavya Rudraiah*, Research Scholar, Dept. of ECE, R V College of Engineering, Bangalore, INDIA. Email: bhavyanadgouda@gmail.com
Dr. Geetha K S, Professor, HOD, Dept. of ECE, R V College of Engineering, Bangalore, INDIA. Email : geethaks@rvce.edu.in

In the conventional method, number of times the occurrence of each intensity value is counted by adding ones to the respective bins where as LSH adds a floating-point to the bins. This value varies exponentially with respect to the location of the pixel where the histogram is considered.

This method proves its effectiveness in calculating the histogram where the pixels contributions are of lower weights as they contain more of occluding noise, background information. Therefore, it proves to be efficient and accurate by elimination of background noise. The decision tree classifier which is feasible, is used to perform the classification of data. It is relatively faster in constructing the tree than any other classification methods. With this trained classifier, the tracking is achieved by sliding window method. This approach demonstrates successfully tracking objects under different appearances. The remaining content of the paper is as follows: The proposed multiple object detection system is explained in section 2. The implementation results, comparison and discussion of using different datasets is explained in section 3. Section 4 briefs the conclusion and future work.

II. PROPOSED SYSTEM

Multiple detection and tracking of vehicles involve two major steps: Feature Extraction and Classification[5]. In this section, the proposed architecture of multiple detection and tracking is shown in Fig.1 It explains the system architecture which utilizes Locality Sensitive Histogram (LSH) and Decision Tree classifier to perform classification of positive and negative samples. This classifier track objects detected using sliding window method.

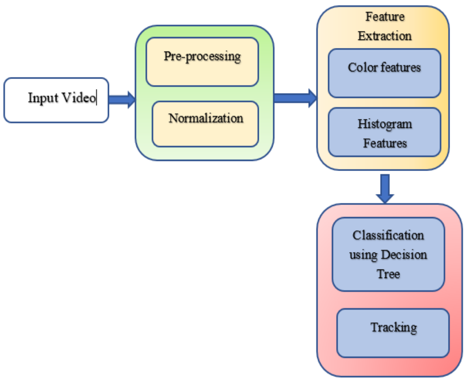

Fig 1: Proposed System Architecture.

The data set used for the proposed system as input consists of images including multiple vehicles of several categories and non-vehicles with different appearance, illumination variations. It includes around 10000 images of varying aspect ratio and size. The images are preprocessed to maintain uniformity. The images of varying aspect ratio are downscaled into 64x64 size and later subjected to extract the features. After pre-processing and normalization, a unique method called Locality Sensitive Histogram (LSHs) algorithm is utilized to extract the features of the required object, which is explained briefly in next section followed by classification and tracking of objects.

A. *Locality Sensitive Histogram (LSH)*

LSH at pixel p, is calculated by:

$$H_p^E(b) = \sum_{q=1}^{W} \alpha^{|p-q|}.Q(I_q,b), \;\; b = 1,\dots\dots B \qquad (1)$$

where α ε (0,1) is a weight controlling parameter, which decreases as the pixel moves away from the target. For a 1D image, the LSH is computed as:

$$H_p^E(b) = H_p^{E,left}(b) + H_p^{E,right}(b) - Q(I_p,b) \qquad (2)$$

Where

$$H_p^{E,right}(b) = Q(Ip,b) + \alpha.\; H_{p+1}^{E,right}(b) \qquad (3)$$

$$H_p^{E,left}(b) = Q(Ip,b) \quad + \alpha.\; H_{p-1}^{E,left}(b) \qquad (4)$$

Using the above-mentioned equations, the right side LSH and left side LSH around a particular pixel *p* is respectively calculated and it is summed up to compute the complete LSH around that pixel [1]. This combines the contributions from all the pixels and thereby considering the weights which reduces exponentially as a function of distance from pixel *p*. The number of additions and multiplications performed at each pixel location appears to be equal illustrating the reduction in complexity of computing LSH per pixel [1].

For 2D or multi-dimensional images, computation is performed separately in each dimension and calculated sequentially. It proves efficiency by capturing intensity information locally. In order to capture color information, full color space can be used to represent color image. But this increases number of bins exponentially. Hence RGB color space with selective quantization is used to represent frequently occurring colors. This step reduces number of colors replacing with the closest colors. Spatial binning is used to reduce the feature vector size into 16x16.

B. *Decision Tree*

Decision tree is one of the best supervised learning algorithm and predictive modelling approach. The most important feature is its capability to break complex-decision making processes into simpler and thus predicting accurate results. It is a tree like structure, having decision nodes and leaf nodes. Each of the decision node has two or more branches. The decision node performs test on attribute, and each branch gives an outcome mapped to a class label called as leaf node. Given a set of attributes tested against tree. It traces the path from root to leaf node and holds classification rules. The Decision tree algorithm is designed as follows: Input: Dataset, D - set of training tuples and associated class labels, list of attributes and splitting criterion [5].

Starting from a node N, if the tuple belongs to a class, then the label is returned to the leaf node with class labelling. If the list is null, leaf node is returned with major class. Splitting criteria continues and for each splitting output the tuples are partitioned and grow as subtrees. If the set is empty, then leaf node is labelled with majority class else returned to node N[5]. In-order to train the classifier, the larger datasets were split into smaller and smaller subsets. The standard scaler function now scales the data equally around the center and is normally distributed with each feature. The dataset is further split into training and test data. This is carried out to avoid overfitting. The decision tree classifier is trained further with the feature vectors of the derived input images. The performance of trained classifier is around 94%- 98%. With this classification accuracy and trained classifier, the objects appearing in each image is subjected to tracking which is explained in next section.

III. TRACKING OF OBJECTS

In this paper, the objects are tracked using feature-based tracking approach. The classifier trained with the feature vectors which represent the object are used to identify and track the object. Sliding window method is used to detect and track object. This method clearly identifies the locality of the object in an image. It uses a rectangular box to slide across the image with fixed width and height.

For all the windows defined, the image classifier slides across the image and detects whether the object of interest is present or not. All the windows with varying sizes are listed which includes the object of interest and its features. This results in several overlapping detections that is resolved into single detection identifying the locality of the objects in the frame. This is achieved by using heatmap which combines several overlapping detections that includes the object and thereby eliminating the false positives. Threshold of certain limit is chosen to combine the overlapping detections and remove false positives.

The above-mentioned step is followed multiple times with varying scale values thereby generating many search windows of different scales. But in order to track the desired object of interest, pipeline is defined. This pipeline creator function is loaded with multiple parameters to load a video and process every frame of all the images with the mentioned techniques, thus creating the video with processed frames. By running the pipeline on the required input video stream, using the heat map and repeated detections in each frame, the object is detected by rejecting outliers. The detected objects then appear with the bounding boxes.

International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-5, January 2021

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The experimental results and evaluation of the implemented method are explained in this section.

The technique used for tracking and detecting multiple objects is implemented using Python 3.7 on an Intel i7 Nvidia GE force CPU.

The performance evaluation and efficiency are analyzed by considering several videos comprising of multiple category vehicles as input. These videos are a collection from several repositories. The videos were chosen based on some challenging factors like pose variation, scale variation, difference in appearance and motion. The screenshots are listed in Fig.2, depicting the tracked object in different frames.

The first, second and third column in Fig. 2 represents the video frames, the heatmap image and the final image where the object is tracked respectively using the proposed technique. The tracked object is highlighted by blue colored bounding box-features representing multiple objects tracked.

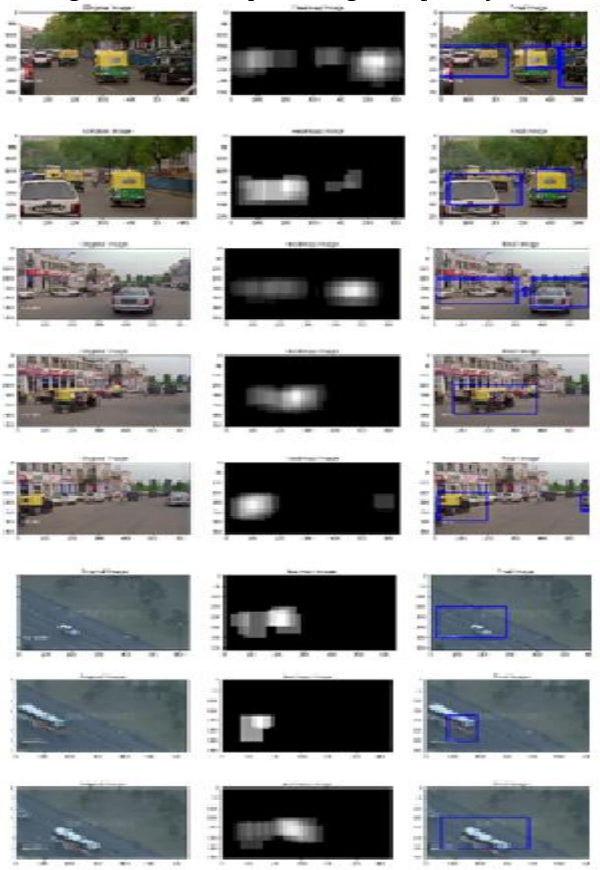

Fig 2: Detection and Tracking of moving objects using proposed technique. The blue box in the above figure represents the tracked object in each frame

The performance of proposed method along with different classifiers using different features are tabulated and plotted for comparison in Table I and Fig.3, respectively. Table I represents the performance of different classifiers using different features for tracking. Fig.3 represents the performance plot of different classifiers using different features along with the proposed technique. It clearly demonstrates that accuracy; precision, recall and f-measure are considerably high and remarkable with the proposed technique. For experimental purpose, different classifiers like k-Nearest Neighborhood, J48, Multi-Layer Perceptron, Support Vector Machine, Decision tree are considered. KNN, J48 and MLP track objects by considering Color, Texture and Shape features resulting with an accuracy of 63% - 65% and 45%[2].

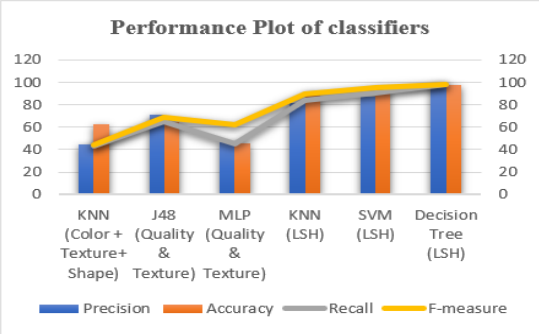

Fig 3 Performance plot of different classifiers with different features.

This results in missing the objects in certain frames, due to improper thresholding and false positives. Using the proposed method, LSH, which proves to be an efficient feature that can be extracted, KNN outperforms well with an accuracy of **91%** similarly[2] Decision tree classifier and Support Vector Machine also tracks effectively with an accuracy of **97**% and **94%** respectively. These classifiers perform precisely in identifying the objects, when Locality Sensitive Histogram is used as feature for extraction than other features. The performance of these classifiers mainly depends on the feature selection and tuning during implementation. With these, the system can accurately detect the object present in an image. The threshold and heatmap method finalize one window that detects the object. The classifier with the proposed technique effectively performs tracking, thereby considering the challenging factors.

Table I Performance comparison of proposed method with different classifiers and features.

Classifiers (Features)	**Precision**	**Accuracy**	**Recall**	**F-measure**
KNN (Color + Texture+ Shape)	44.99	63.01	43.62	44.29
J48 (Quality & Texture)	71.4	65	65	69
MLP (Quality & Texture)	47	45.3	45	62.1
KNN (LSH)	**81.48**	**90.56**	**83.87**	**89.79**
SVM (LSH)	**87.5**	**94.33**	**90.625**	**95.082**
Decision Tree (LSH)	**98.97**	**97.46**	**97.82**	**98**

V. CONCLUSION

In this paper, an approach to detect and track multiple category vehicles using Locality Sensitive Histogram is proposed. This method proves its effectiveness by classifying the images with an accuracy of **97%,** yet leading to an effective tracking method. Through well-known measures like precision, recall and F-measure, the proposed method is experimentally validated. The results obtained are remarkably high than compared to other methods. As Robust detection and tracking is still posing new challenges, the future work can be focused and directed in considering these challenges with new combination of datasets in several vison-based applications.

REFERENCES

1. S. He, Rynson W.H. Lau, Q.Yang, J Wang, M Yang, "*Robust Object tracking via Locality Sensitive Histograms*". In IEEE Transactions On Circuits And Systems For Video Technology, 2016.
2. T.Mahalingam, M. Subramaniom, "*A Robust Single and Multiple moving object detection , tracking and Classification*", Applied Computer and Informatics, 2018.
3. A Arinaldi , J A Pradana, A A Gurusinga, "*Detection and classification of vehicles for traffic video analytics*", Elsevier, INNS Conference on Big Data and Deep Learning 2018.
4. A. Yilmaz, O.Javed , M.Shah, "*Object tracking : A Survey*", ACM Computer Survey, Dec.2006.
5. S S. Sarikana , A. M Ozbayoglub , O Zilcia, "*Automated Vehicle Classification with Image Processing and Computational Intelligence*", Complex Adaptive Systems Conference, Engineering Cyber Physical Systems, CAS October 30 – November 1, 2017.

AUTHOR PROFILE

Bhavya R is a Research Scholar at Dept of ECE, RV College of Engineering, Bangalore, INDIA. She has completed her Master of Technology in VLSI Design and Embedded Systems in 2010 from Visvesvaraya Technological University. She has served as an Assistant Professor for 5 years in Engineering Colleges. Her Research Interests includes Image Processing, Signal Processing and Video Processing.

K S Geetha is Professor and Head of Department of Electronics and Communication Engineering, R V College of Engineering, Bangalore, India. She has received her B. E and MTech in Electronics Engineering from National Institute of Engineering, Mysore, India. Her research interests include Digital Signal Processing, Image and Video processing, Large Area Flexible Microelectronics. Publications which include various international journals and international conference proceedings.