

Real Time Crowd Counting using OpenCV

Ms. Subashree D, Shrushti Rohidas Mhaske, Sonal Rajesh Yeshwantrao, Ayush Kumar
Computer Science and Engineering Department,
SRM Institute of Science and Technology, Ramapuram, Chennai,
India (600089)

Abstract:- In this paper we propose a real time crowd counting method using OpenCV. With reference to the previous methods such as edge recognition, morphological filter, SVM order which are not real time applications, our method trains the system using video streaming and provides result in real time. OpenCV for people counting, image processing and deep learning object detector are used. This method leverages both object detection and tracking to improve the accuracy of the people counter.

I) INTRODUCTION

Single image crowd counting method evaluates the number of people in the crowded image. Conventional methods are challenging due to severe obstruction and complex backgrounds. Crowd count detection has various applications such as public safety, scheduling trains, traffic control etc. In our proposed method we make use of OpenCV is a programming language that can be used to perform standard computer vision and image processing tasks. Related work done in this field included crowd detection using detection and regression methods, RGBD counting and multitask strategies. These methods lacked accuracy. The main objective is to develop a simple real-time system to count the number of people by using OpenCV. This is mainly to ensure safety of people under all circumstances.

II) LITERATURE SURVEY

Crowd counting is treated as an object/person detection problem in detection-based techniques, which assumes that a crowd is made up of individual items. [1]. Handcrafted features were used in early works to detect people, but they were not resistant to extreme large-scale variance or occlusion in crowded scenes or clustered settings [2]. Despite the success of deep network-based object detectors in recent years, impressive object detection results, they still outperform regression-based methods when it comes to crowd counting [5]. A video-based face recognition [6] which works on human face detection. The work done here works efficiently on challenging scenarios like multiple shot videos and surveillance videos with low quality frames. However, it does not work with people wearing masks. Several studies have looked into RGBD crowd counting in order to better estimate crowd counts. The majority of these studies concentrate on using depth information to increase crowd scene person/head detection. [8] Their detection module, on the other hand, does not outperform the module for regression. Meanwhile, since the depth map is not explicitly fed into the regression module, it is underutilized [7] which performs crowd count classification

and density map estimation together, but the accuracy in detecting faces is low.

III) SYSTEM ARCHITECTURE

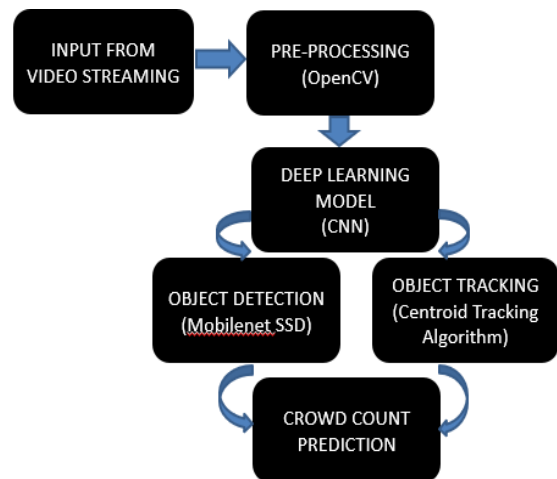


Fig.1 Architecture Diagram

A webcam is used for video input. The throughput rate of estimated frames per second (FPS) is calculated. The frames are then swapped to RGB and resized. Image processing and standard computer vision functions are performed using OpenCV. OpenCV is also used for opening and writing of video files, deep neural network inference and showing the output frames on screen. In the proposed system we use Object Detection and Object Tracking in two phases for more accuracy. Mobile-Net Single Shot Detector is used for Object Detection.

We run it only once after every N frames since it is expensive. Mobile-Net contains all the pre-trained deep learning model files. An object tracker is used for every object detected to keep a track of the objects. We use a combination of Correlations filters and centroid tracking algorithm for the tracking purpose. Firstly, we use the coordinates of bounding boxes to determine the center, which is also called centroid. Then the Euclidean distance between existing and new centroids is calculated and given object Ids. Objects that have entered the field are registered and those which have left are deregistered. Dlib is used for the implementation of object tracking. OpenCV and Python is used to implement and predict the crowd count.

IV) METHODOLOGY

1) Video Streaming

For object detection, we work with a webcam and calculate the Frames Per Second (FPS) throughput rate. When working on this problem the first two constraints to think of is capability with FPS and Accuracy.

2) Pre-Processing

Frames are pre-processed by resizing and switching to rgb. OpenCV is a library for performing common computer vision and image processing tasks. Deep neural network inference, opening and writing video files, and showing output frames to our screen will all be done with OpenCV.

4) Object Detection

It is a computer technology that deals with identifying instances of semantic objects of a certain class in images and videos and constructing bounding boxes around those objects. It is related to computer vision and image processing.

5) Object Tracking

We use the centroid tracking algorithm for this. The center is calculated using bounding boxes. The distance between new and existing centroids is then determined using Euclidean geometry. It also unregisters objects that have been removed from the field.

V) ALGORITHM

1) Deep Learning Algorithm

Deep learning is often regarded as a branch of artificial intelligence. It's an area that relies on analyzing computer formulas to learn and improve. Although AI makes use of simpler assumptions, Deep learning makes use of phoney neural networks that are meant to simulate how people consider and understand. Up until now, neural organizations were constrained by figuring power, and thus intricacy was restricted. Larger, more refined neural networks have been made possible by advances in Big Data processing, allowing computers to notice, understand, and respond to complex situations faster than humans. Image order, language interpretation, and discourse acknowledgment have all been aided by deep learning.

2) Centroid-based Tracking Algorithm

Centroid-based tracking is a tracking algorithm that is simple to understand but extremely effective. Since it is based on the Euclidean distance between one current object centroids and the second new object centroids between subsequent frames in a film, this object tracking algorithm is known as centroid tracking. The centroid tracking algorithm uses (x, y) coordinates for every detected object in each frame, assuming that some sets of the bounding box are transferred. Bounding boxes must be calculated for each frame of the film, or, to put it another way, for each object identified by the camcorder. Following the assignment of bounding boxes within the frame with their (x, y) coordinates, the centroid of each bounding box is determined, and each bounding box is given a unique ID. The centroid of an object is computed in each subsequent frame using the bounding box definition that we discussed earlier. However, giving a new unique ID for each detection of the thing which hinder the objective of object tracking, so we'll see if we can compare the centroid of the new object to

that of an existing object to resolve this, and to do so, we'll use the distance formula to measure the Euclidean distance between the two objects.

3) OpenCV

OpenCV (Open Source Computer Vision Library) is a programming library for computer vision and artificial intelligence. The aim of OpenCV was to give PC vision applications a logical framework and to speed up the use of machine learning in business processes. Since OpenCV is a BSD-approved project, it makes it easier for organizations to use and change the code. The library contains 2500 enhanced figures and includes a detailed game plan for both masterpieces as well as cutting-edge PC vision and AI computations. These figurines are frequently used to recognize and see faces, perceive objects, depict human activities in chronicles, monitor camera improvements, track moving articles, separate 3D models of things, generate 3D point fogs from sound framework cameras, enter pictures to give a significant standard image of a whole scene, find tantamount pictures from a picture database, and so forth, destroy red eyes from photos taken with stripe, monitor eye progressions, see the view and find markers to overlay it with augmented reality, and so on. OpenCV has a consumer community of over 47 thousand people, with a download count of over 18 million.

4) Single Shot Detector Algorithm

Single Shot Detector (SSD) Algorithm is used for object detection. Mobilenet SSD contains all the pretrained deep learning model files. SSD comprises of two parts:

i) Extract feature maps

ii) Apply convolution filter to detect objects

The Single Shot Detector (SSD) is intended to be independent of the base network, allowing it to work on top of any base network, including VGG, YOLO, and Mobile-Net.

Mobile-Net was integrated into the SSD framework to address the challenges of running high-resource and power-consuming neural networks on low-end devices in real time.

VI) RESULT AND DISCUSSION

On a standard CPU, this system has the ability to run in real-time. It makes use of deep learning object detectors to increase the accuracy of person detection. For increased tracking accuracy, it also employs Centroid tracking and correlation filters are two distinct object tracking algorithms, allowing it to detect new people as well as recover people who might have been "stuck" during the tracking process. This system can also be used for vehicle traffic count.

VII) CONCLUSION AND FUTURE ENHANCEMENT

Using OpenCV and Python, we built a people counter. It is possible to incorporate a model that calculates the distance between the bounding boxes and thus improves the precision of the violation. The performance of object detection in image processing is required for a growing number of real-time applications, and we can detect any type of object with this application. We will use various types of extraction process techniques in the future for various purposes, and this technique can be used in airports, shopping malls, businesses, parks, and so on.

VIII) REFERENCES

- [1] W. Liu, M. Salzmann, and P. Fua, "Context-aware crowd counting," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 5099–5108.
- [2] V. A. Sindagi and V. M. Patel, "HA-CCN: Hierarchical attentionbased crowd counting network," IEEE Trans. Image Process., vol. 29, pp. 323–335, 2020.
- [3] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, and X. Yang, "Crowd counting via adversarial cross-scale consistency pursuit," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 5245–5254.
- [4] T. Zhao, R. Nevatia, and B. Wu, "Segmentation and tracking of multiple humans in crowded environments," IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 7, pp. 1198–1211, Jul. 2008.
- [5] W. Ge and R. T. Collins, "Marked point processes for crowd counting," in Proc. CVPR, Jun. 2009, pp. 2913–2920.
- [6] M. Li, Z. Zhang, K. Huang, and T. Tan, "Estimating the number of people in crowded scenes by MID based foreground segmentation and head shoulder detection," in Proc. 19th Int. Conf. Pattern Recognit., Dec. 2008.
- [7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", in arXiv:1506.01497v3 [cs.CV] 6 Jan 2016
- [8] Desen Zhou and Qian He, "Cascaded Multi-Task Learning of Head Segmentation and Density Regression for RGBD Crowd Counting", Digital Object Identifier 10.1109/ACCESS.2020.2998678, June 10, 2020
- [9] Lingbo Liu, Zhilin Qiu, Guanbin Li, Shufan Liu, Wanli Ouyang and Liang Lin, "Crowd Counting with Deep Structured Scale Integration Network", 2019 IEEE/CVF International Conference on Computer Vision (ICCV)
- [10] Diping Song, Yu Qiao and Alessandro Corbetta, "Depth Driven People Counting Using Deep Region Proposal Network", Proceedings of the 2017 IEEE International Conference on Information and Automation (ICIA) Macau SAR, China, July 2017