International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 06 | June - 2021 ISSN: 2582-3930

PEOPLE COUNTER USING OPENCV

Vipul Sehgal, Shiwank Pathak, Sumit Katyal, Hritik Sharma

HMR INSTITUTE OF TECHNOLOGY AND MANAGEMENT, CSE

---***---

Abstract - We offer a real-time people counting method based on OpenCV in this paper. In comparison to prior methods such as edge identification, morphological filters and SVM order that do not perform real time counting, our method trains the system to work in real time. It trains the system by streaming footage and displaying the results in real time. OpenCV is a computer vision library that is used to count people, analyze images, and for analysis deep learning object detector is also used. To improve the accuracy of the people counter, this method uses both object detection and tracking.

Key Words: opencv, computer vision, SVM, morphological filters

1.INTRODUCTION *(Size 11, Times New roman)*

The number of individuals in a crowded photograph is counted using a single image crowd counting approach. Due to extreme obstruction and complex backgrounds, traditional approaches are difficult to use. Crowd count detection has a variety of uses, including public safety, train scheduling, and traffic control. We employ OpenCV in our proposed technique, which is a programming language for doing standard computer vision and image processing tasks. People Counter or Crowd detection employing detection and regression algorithms, RGBD counting, and multitask tactics are examples of related work in this subject. These strategies were ineffective and inferior compared to our method. The main goal is to use OpenCV to create a basic real-time system for counting the number of individuals. This is primarily to safeguard people's safety in all conditions.

2 LITERATURE SURVEY:

In detection-based techniques, crowd counting is addressed as an object/person detection problem, assuming that a crowd is made up of individual things. Early studies used handcrafted characteristics to detect humans, but they were not robust to large-scale variation or occlusion in crowded scenes or clustered environments. Despite the recent success achieved by deep network-based object detectors, which have shown outstanding object detection results, regression-based algorithms still beat them when it comes to crowd counting. A video-based face recognition system that detects human faces. The work done here is effective in difficult conditions such as numerous shot videos and surveillance movies with low frame quality. It does not, however, work with those who are wearing masks. In order to better estimation of crowd counts, several studies have looked into RGBD crowd counting.

The majority of these studies focus on how to use depth information to improve person/head detection in crowd scenes. However, their detection module does not beat the regression module. Meanwhile, the depth map is underutilized because it is not directly supplied into the regression module, which does crowd count classification and density map estimation concurrently, but the accuracy is excellent.

Meanwhile, the depth map is underutilized since it is not explicitly fed into the regression module, which combines crowd count classification and density map estimation, but the accuracy in spotting faces is low.

2.1 SYSTEM ARCHITECTURE

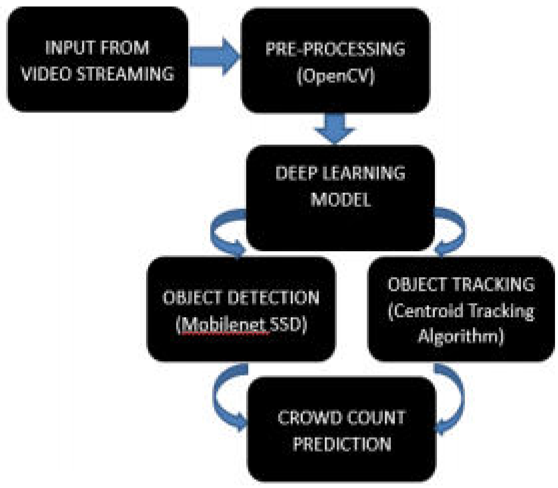


Diagram representing system architecture

For video input, a webcam is used. The expected frames per second (FPS) throughput rate is determined. After that, the frames are converted to RGB and scaled. OpenCV is used for image processing and conventional computer vision functions. OpenCV is also used to open and write video files, as well as infer deep neural networks and display the output frames on the screen. For greater accuracy, we use two phases of Object Detection and Object Tracking in the proposed system. Object detection is handled via the Mobile-Net Single Shot Detector. Because it is expensive, we only run it once every N frames. All of the pre-trained deep learning model files are stored in Mobile-Net. To keep track of the items, an object tracker is employed for each one that is detected. For tracking, we employ a mixture of correlation filters and a centroid tracking method. To begin, we utilize the bounding box coordinates to calculate the center, also known as the centroid. The

Euclidean distance between the existing and new centroids is then calculated, and object IDs are assigned. Those who enter the field are registered, while those who leave are deregistered. Dlib is the library that is used to implement object tracking. Python and OpenCV are utilized in this project. The audience count should be implemented and predicted.

2.2 METHODOLOGY:

2.1.1) Streaming Realtime Video

We use a webcam to detect objects and calculate the Frames Per Second (FPS) throughput rate. The first two restrictions to consider when working on this problem are competence with FPS and accuracy. It is also vital to consider that the detection area encompasses the contour of the person; if it does not, the person will not be detected or false detections will occur. Another essential consideration is the video resolution, as lesser resolutions allow for better detection and counting than higher resolutions.

2.2.2) Pre processing of frames

By scaling and switching to RGB, frames are pre-processed. OpenCV is a library for handling standard image processing and computer vision tasks. OpenCV will be used to infer deep neural networks, open and write video files, and display output frames on our screen.

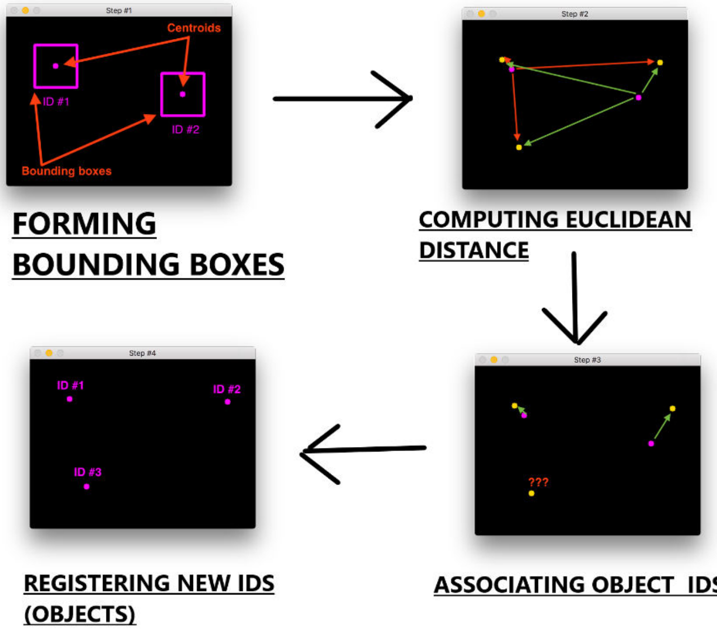
2.3.3)Object detection

Object detection is a process involving computer technology that identifies instances of semantic items of a specific class in photos and videos and creates bounding boxes around them. It is a part of image processing and computer vision. As the name specifies, we detect the presence of objects in real time video and object detection is a major part of our hybrid model.

Above figure represents Detection of a person

2.2.4)Object tracking

For this, we employ the centroid tracking technique. Bounding boxes are used to calculate the center. Using Euclidean geometry, the distance between new and old centroids is calculated. Objects that have been removed from the field are also unregistered.



The above diagram specifies the overall process that are underwent in the object tracking process.

ALGORITHMS USED:

1)Deep learning Algorithms:

Deep learning is frequently considered a subset of artificial intelligence. It's a field that learns and improves by examining computer formulas. Deep learning uses phony neural networks to replicate how people think and understand, whereas AI uses simpler assumptions. Until now, neuronal organizations were confined by calculating power, limiting their complexity. Advances in Big Data processing have enabled larger, more precise neural networks, allowing computers to perceive, analyze, and respond to complicated situations faster than humans. Deep learning has helped with image order, language interpretation, and discourse recognition.

2)Object Tracking Algorithms:

Centroid-based tracking is a simple-to-understand but very effective tracking technique. This object tracking algorithm is called as centroid tracking because it is based on the Euclidean distance between one current object centroids and the second new object centroids between subsequent frames in a video. The centroid tracking technique assumes that some sets of the bounding box are exchanged and uses (x, y) coordinates for every identified object in each frame. For each frame of the film, or, to put it another way, for each object spotted by the camcorder, bounding boxes must be calculated.

The centroid of each bounding box is computed following the assignment of bounding boxes within the frame with their (x, y) coordinates, and each bounding box is given a unique ID. Using the bounding box definition we described before, the centroid of an item is determined in each following frame. However, creating a new unique ID for each detection of the thing will obstruct the goal of object tracking, so we'll see if we can resolve this by comparing the centroid of the new object to that of an existing object, and we'll do so by using the distance formula to calculate the Euclidean distance between the two objects.

3)**OPENCV:**

OpenCV (Open Source Computer Vision Library) is a computer vision and artificial intelligence programming library. The goal of OpenCV was to provide a logical foundation for PC vision applications and to accelerate the usage of machine learning in commercial operations. Because OpenCV is a BSD-approved project, it's easier for businesses to utilize and modify the code. There are 2500 improved and added figurines in the library including updated data, as well as a comprehensive game plan for both processes, as well as cutting-edge computer vision and artificial intelligence calculations. 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, and so on. We can also overcome various defects like reddening of eyes in the photo clicked by stripe, ,monitor eye progressions etc. using augmented reality. OpenCV has a user community of around 47 thousand people, with over 18 million downloads.

4)SSD ALGORITHM:

For object detection, the Single Shot Detector (SSD) Algorithm is utilized. All of the pretrained deep learning model files are stored on the Mobilenet SSD. There are two pieces to a solid-state drive (SSD):

i)Create a feature map

ii) Detect objects using a convolution filter

The Single Shot Detector (SSD) is designed to be network-agnostic, allowing it to run on top of any base network, such as VGG, YOLO, and Mobile-Net.

Mobile-Net was incorporated into the SSD architecture to overcome the difficulties of running high-resource, power-hungry neural networks in real time on low-end devices.

2.3 RESULT:

This system is capable of running in real time on a regular CPU. Deep learning object detectors are used to improve the accuracy of human detection. It also employs Centroid tracking and correlation filters, which are two independent object tracking techniques that allow it to discover new persons as well as recover persons who may have become "stuck" during the tracking process, for enhanced tracking accuracy. This approach can also be used to count the number of vehicles on the road.

3. CONCLUSIONS

We created a people counter with OpenCV and Python. It is possible to include a model that_determines the distance between the bounding boxes, improving the accuracy of the violation. Item detection performance in image processing is becoming increasingly important for a rising variety of real-time applications, and we can detect any type of object with this application.

FUTURE ENHANCEMENTS:
In the future, we will utilize a variety of extraction process techniques for a variety of objectives, and this approach can be employed in airports, shopping malls, enterprises, parks, and other places. Also due to the increase in COVID-19 cases our model can be used to ensure whether there is no overcrowding in a region or not and can be used for prevention of gathering of many people in a particular region.

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