DETECTING AND COUNTING FACES USING PYTHON- OpenCV AND NumPY

Vivek Chandra, Praseeda Atmuri, Aditya Reddy, Kilaru Sreenu

Abstract:

There are numerous potentials uses for real-time crowd counting, including subway crowd flow control and surveillance. A novel and quick method for estimating the number of people in crowded surveillance scenes is presented in this paper. This method is resistant to changes in background and illumination and can count people in real-time. A multi-scale head-shoulder detector is trained using the boosted classifier's cascade and combined rectangle features. With high accuracy, the detector can find a human in every frame. Then, in subsequent frames, human tracking is used to follow the people that have been detected and eliminate duplicates. Probes a certifiable video shows the proposed strategy can give an exact assessment progressively.

Keywords: Crowd Counting, Human Detection, Human Tracking.

1.1 Introduction:

The process of counting the number of people in videos or images is known as crowd counting. This technique can be used in a few areas that are relevant to our day-to-day lives, such as urban planning, health care, disaster management, public safety management, and defense. As a result, new research is being conducted in this area. Techniques based on supervised and unsupervised learning are the two main types of crowd techniques. There are a few techniques for crowd counting using convolutional neural networks and traditional methods. Occlusion, perspective and scale distortion, and non-uniform distribution are just a few of the limitations of crowd counting methods. Along with the count, density estimation provides an idea of the people's spatial distribution.

In this project, we will determine the number of people in the live video. Face detection in conjunction with image processing can be used to estimate the number of humans. We display a square box with the count on top for each human face that is detected. The Spyder IDE's Python programming language and libraries like OpenCV, NumPy and dlib were utilized here for human face detection.

The three main categories of CNN-based methods are as follows:

- Property of the network
- Training process
- Image view

a) Property of the network:

Approaches based on CNN can be categorized as follows:

a. Basic CNNs: Includes just essential CNN layers.

b. Scale-conscious models: Scalability is provided by architecture with multiple resolutions and columns.

c. A context-sensitive model: CNN structure includes both worldwide and neighborhood context-oriented data of a picture

d. Perform multiple tasks systems: estimating crowd density and crowd counting in addition to combining crowd velocity estimation and foreground-background subtraction.

b) Training process:

CNN-based approaches can be divided into

a. Patch-based training: Patches will be created from the input images.

b. Whole-image-based training: Use the input image as such.

c) Image View:

We can divide image view-based methods into two groups based on the perspective of the image.

a. Arial view based: The camera and object are perpendicular to one another.

b. Perspective view based: object and camera are parallel with one another.

1.2 Literature Survey:

The primary performance issue of detection-based crowd counting and occluded multiple objects is the low resolution of some of the images. When local features are extracted from the segmented images and the regression model is used to estimate the crowd count in each segment, regression-based counting performs better in this setting. Prior to this, global image features were used to develop regression-based methods that were unable to capture the information's region-wise distribution. Density-based methods, which overcome the drawback of regression-based methods and preserve location information, generate density values that are estimated using low-level features like pixels or regions. The density map estimation techniques may differ depending on the choice of the loss function and the type of prediction, so the predicted density maps may have distinct characteristics. Pixel- or region-wise prediction and loss functions are both possible. Image-wise prediction computations are relatively faster because they reuse computations. Due to the possibility of deviation in the mapping between density and image, these kinds of methods are ineffective because the actual count can frequently be inaccurate.

In computer vision, there have been numerous proposals for crowd-counting algorithms. Head or body detection was used in the early works to estimate the number of pedestrians. Such location-based strategies are restricted by serious impediments in very thick group scenes. Global counts are predicted by means of regressors that have been trained with low-level features like HOG, SIFT, Fourier Analysis, detections, and deeper representations outperform these low-level features.

Crowd counting has entered the deep CNN era in recent years. A thorough review of ongoing CNN-based techniques for swarm including can be found. Wang and co. Prepared an exemplary Alexnet-style CNN model to foresee swarm counts. Unfortunately, because it does not provide an estimate of the distribution of the crowd, this model has limitations for crowd analysis. Zhang and co. proposed an alternatively regressed deep convolutional neural network with two related learning objectives for crowd counting: Density and crowd count on a map. This learning with switchable objectives aids in the achievement of both objectives. However, this method has limited application because it requires perspective maps, which are difficult to come by in practice during the training and testing phases.

A 11-convolutional layer joins features from various CNN columns with varying receptive fields to regress crowd density, which is designed to capture scale variation and perspective. Before crowd patches become multi-column regressors, a patch-based switching architecture inspired by MCNN is proposed. Using patch-wise variations in density within a single image, the switch net is trained as a classifier to intelligently select the most suitable regressor for a specific input patch. They prioritize the precision of the predicted crowd count over the quality of the regressed density map by employing max pooling layers and l2 loss. Consequently, other higher-level cognition tasks that rely on these poor maps, such as counting and scene recognition, are negatively impacted. The most recent study, CPCNN, proposes a contextual Pyramid CNN for incorporating both global and local contexts, which can be learned from different densities. A Fusion-CNN that is made up of convolutional and fractionally-stride layers combines contextual information with high-dimensional feature maps that are taken from a multi-column CNN. CP-CNN and our method are both recent attempts to consider the quality of density maps. CP-CNN and our method are both recent attempts to consider the quality of density maps. Through adversarial training, we not only propose a patch-to-density translation but also introduce a novel regularizer to compel cross-scale model calibration and encourage different scale paths to collaborate.

2.1 Existing System with block diagram:

Video Streaming: We use a webcam to calculate the throughput rate in frames per second (FPS) for object detection. When working on this issue, the accuracy and FPS capability are the first two constraints to consider. Pre-Processing Frames undergo resizing and rgb conversion prior to processing. OpenCV is a library for common image processing and computer vision tasks. Profound brain network induction, opening and composing video documents, and showing yield edges to our screen will be generally finished with OpenCV. Object detection is a computer technology for locating instances of semantic objects belonging to a particular class in images and videos and creating bounding boxes around them. It has to do with image processing and computer vision. For this, we use the centroid tracking algorithm. Object tracking Bounding boxes are used to determine the center. Euclidean geometry is then used to determine the distance between the new and existing centroids. Additionally, it deregisters removed objects from the field.

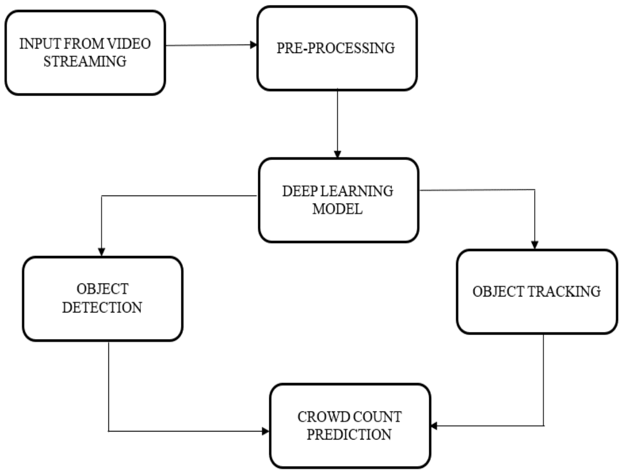

Fig 1: Block Diagram of Existing method

2.2 Problem Statement:

• An operation like max pool makes a Convolutional neural network significantly slower.

• If the computer does not have a powerful GPU, the training process will take a long time if the CNN has multiple layers. Additionally, it is difficult to comprehend.

• Coordinate frames, which are an essential part of human vision, are absent from CNNs.

• To process and train a convolutional n

Objective: The goal is to get the bounding box by using one or more methods, such as the coordinates of the face in the image. The number of faces that will be computed depends on the different areas covered by the number of coordinates.

2.3 Proposed System with Block Diagram:

In this project, we proposed a better way to count the number of human faces in the live stream that takes less time and is simpler. Additionally, this approach is simpler than the previous one. The system can be trained using the small dataset that we propose.

Open CV, NumPy, and dlib are just a few of the Python libraries we use in this model.

Step-1:

The system's camera or the webcam that is connected to the system starts recording video.

Step-2:

The camera begins locating the coordinates of multiple faces in the video frame as soon as it starts.

Step-3:

It begins capturing the human faces that are present in the video frame once it has detected the face coordinates.

Step-4:

The count keeps going up as more and more faces are captured in the frame.

Step-5:

The highest number is the number of faces in the video, and each face will be identified by a numerical number.

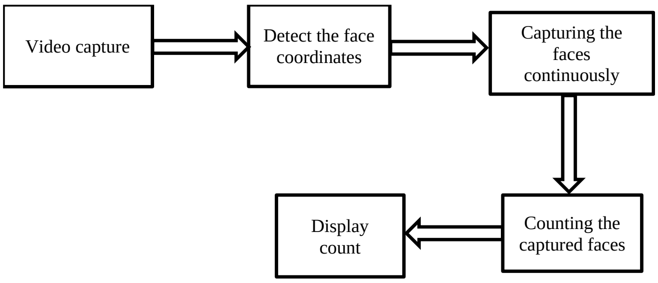

Fig 2: Block Diagram of proposed

3.1 Software used:

Spyder:

Spyder is a scientific environment that was developed by and for scientists, engineers, and data analysts. It is free and open source, written in Python, and designed for Python. It is one of a kind because it combines the capabilities of a comprehensive development tool for advanced editing, analysis, profiling, and debugging with those of a scientific package for data exploration, interactive execution, deep inspection, and beautiful visualization.

4.1 Results:

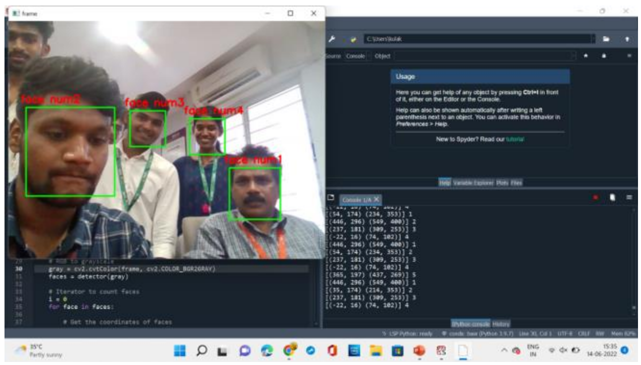

Figure 1: faces are detected and counted and number of faces detected are four.

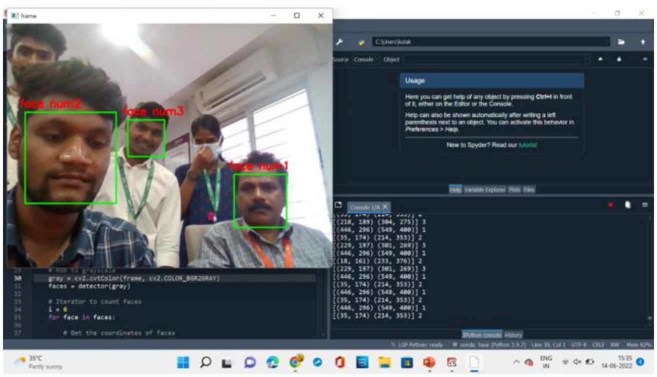

Figure 2: one person is wearing mask so it cannot detect the face and number of faces detected are three.

Let us say that if we can connect the buzzer to this project after a certain number of faces have been taken, we will be able to set off the buzzer and prevent people from entering the area.

In the alternate manner as opposed to giving specific number as information, we can compute thickness and can actuate the bell and prevent individuals from entering the region.

References:

[1] Ryan, David, Simon Denman, Clinton Fookes, and Sridha Sridharan. "Crowd counting using multiple local features." In 2009 Digital Image Computing: Techniques and Applications, pp. 81-88. IEEE, 2009.

[2] Chan, Antoni B., and Nuno Vasconcelos. "Counting people with low-level features and Bayesian regression." IEEE Transactions on Image Processing 21, no. 4 (2012): 2160-2177.

[3] Idrees, Haroon, Imran Saleemi, Cody Seibert, and Mubarak Shah. "Multi-source multi-scale counting in extremely dense crowd images." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2547-2554. 2013.

[4] Kong, Dan, Douglas Gray, and Hai Tao. "A viewpoint invariant approach for crowd counting." In 18th International Conference on Pattern Recognition (ICPR'06), vol. 3, pp. 1187-1190. IEEE, 2006.

[5] Cho, Siu-Yeung, Tommy WS Chow, and Chi-Tat Leung. "A neural-based crowd estimation by hybrid global learning algorithm." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 29, no. 4 (1999): 535-541. [27] Chen, Ke, Shaogang Gong, Tao Xiang, and Chen Change Loy. (2013) "Cumulative attribute space for age and crowd density estimation." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2467-2474.

[6] Chen, Ke, Chen Change Loy, Shaogang Gong, and Tony Xiang. (2012) "Feature mining for localised crowd counting." In BMVC, vol. 1, no. 2, p. 3.

[7] Li, Yuhong, Xiaofan Zhang, and Deming Chen (2018).. "CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1091-1100

[8] Kang, D., Ma, Z., & Chan, A. B. (2018). Beyond Counting: Comparisons of Density Maps for Crowd Analysis Tasks—Counting, Detection, and Tracking. IEEE Transactions on Circuits and Systems for Video Technology, 29(5), 1408-1422.

[9] Deepak Babu Sam, Shiv Surya, and R. Venkatesh Babu. Switching convolutional neural network for crowd counting. In IEEE Conference on Computer Vision and Pattern Recognition, pages 4031–4039, 2017.

[10] Z. L. Lin and L. S. Davis. Shape-based human detection and segmentation via hierarchical part-template matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32:604–618, 2010.

[11] M. Wang and X. Wang. Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In CVPR, 2011.

[12] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, 1:90–97 Vol. 1, 2005.

[13] S. An, W. Liu, and S. Venkatesh. Face recognition using kernel ridge regression. 2007 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–7, 2007.

[14] G. J. Brostow and R. Cipolla. Unsupervised bayesian detection of independent motion in crowds. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 1:594–601, 2006.

[15] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. 2008 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–7, 2008.

[15] K. Chen, C. C. Loy, S. Gong, and T. Xiang. Feature mining for localised crowd counting. In BMVC, 2012.

[16] A. Bansal and K. S. Venkatesh. People counting in high density crowds from still images. CoRR, abs/1507.08445, 2015.

[17] D. Kong, D. Gray, and H. Tao. A viewpoint invariant approach for crowd counting. In ICPR, 2006.

[18] V. Rabaud and S. J. Belongie. Counting crowded moving objects. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 1:705–711, 2006

[19] V. A. Sindagi and V. M. Patel. A survey of recent advances in cnn-based single image crowd counting and density estimation. Pattern Recognition Letters, 2017.

[20] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao. Deep people counting in extremely dense crowds. In ACM Multimedia, 2015.

[21] C. Zhang, H. Li, X. Wang, and X. Yang. Cross-scene crowd counting via deep convolutional neural networks. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 833–841, 2015.

[22] L. Boominathan, S. S. Kruthiventi, and R. V. Babu. Crowdnet: A deep convolutional network for dense crowd counting. In ACM Multimedia, 2016.

[23] D. B. Sam, S. Surya, and R. V. Babu. Switching convolutional neural network for crowd counting. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[28] V. A. Sindagi and V. M. Patel. Generating high-quality crowd density maps using contextual pyramid cnns. 2017 IEEE International Conference on Computer Vision, 2017.

[37] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. Single image crowd counting via multi-column convolutional neural network. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 589–597, 2016.

Author:

1. Chandra Vivek,
Electronics And Communication Engineering,
Institute of Aeronautical Engineering, Dundigal,
vivekchandra.1324@gmail.com

2. Aditya Reddy Onteddu,
Computer Science And Engineering,
Mallareddy Institute Of Engineering And Technology,
adityareddyonteddu123@gmail.com

3. Praseeda Atmuri,
Electronics And Communication Engineering,
Institute of Aeronautical Engineering, Dundigal,
praseedaathmuri@gmail.com

4. Kilaru Sreenu,
Electronics And Communication Engineering,
Institute of Aeronautical Engineering, Dundigal,
Kilarusreenu@gmail.com