International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 05 | May - 2022 Impact Factor: 7.185 ISSN: 2582-3930

Crowd Surveillance System

Gayatri Shinde, Pratibha Adhav, Lokesh Malkani, Pooja Pedgaonkar | Prof.Ms.Ekta Choudhari

Abstract

A huge number of studies related to features for improving the accuracy of object detection systems using the Convolutional Neural Network (CNN). The majority of projects/research is based on free parking space which is a slow model and which is based on the CNN model. In object detection technique monitoring the objects using by using the traditional manual technique, the video surveillance technique is feasible to use when there is a vast number of objects to monitor. To improve real-time problems whose accuracy should be fast and can be used in day-to-day life. To enhance the accuracy and speed of the CNN algorithm we can add an extra to the model which is known as YOLO (You Only Look Once). It enables the system that can recognize objects in the provided frame. Real-time object detector operation requires Graphics Processing Unit (GPU) for object detection [1].

Introduction

In recent years, the development of technologies is at its highest peak and traditionally doesn't support the practice of using Closed-Circuit Television System (CCTV) is not practically possible [2]. To build a realistic model we can create a GPU but it will not be a relevant option to use because it will be higher and eventually the cost of training will increase so, we can create a CNN that can operate on a conventional GPU for which training requires only once. Features of the most common activities will be identified using the image annotation technique. The main objective of this approach is its operating speed is faster to detect objects and also parallel computing is possible. As a result, we came up with an efficient solution to count crowds and detect crowd activities.

Literature Survey

The automation of data analysis and collection of samples or the behavior of the crowd in high demand is important to provide security and safety to society. Crowd density estimation is based on textures captured in the images and the basic idea is that the current images are classified into different crowd density classes. The classification is based on a self-organizing map (SOM) neural [3] in which feature vectors are composed as texture descriptors extracted from the occurrence of matrices and using window-centered classification [4]. At this stage estimating all pixels of the image is a time-consuming process to optimize its distributed algorithm for the Beowulf environment is executed parallel as a Virtual machine. In this algorithm, the master processor divides the image into n fragments then each image fragment is sent to a slave processor. Each processor performs the classification of images as texture classification than using a sequential algorithm. During this experiment pixel texture classification is carried out which is 15x15 window centered wherein four co-occurrences is been calculated. The approach takes part in account the geometric distortions caused by the camera's position under which surveillance is mapped on a finer line of textures and the result is 77.33?curate for estimation of the crowd using an image. The collection of crowd density data and analyzing it for automation purposes is desirable increasing for longterm site management using image-processing techniques with CCTV systems. The main method of this system is to use a reference image that is the area by pedestrians in a crowd image containing information about crowd density. This approach is considered the fastest to determine the crowd density. A reference image with only the background to classify pixels in a crowd image by subtraction as belonging to either background. Comparing the number of pixeled images and the number of people counted manually has a clear tendency linearly with an increasing number of

images. A deep convolutional consists of a core trainable segmentation engine which is an encoder network and corresponding decoder network under pixel-wise classification. The architecture of the encoder-decoder is identical to the 13 convolutional layers of the network [5]. The main purpose of designing SegNet is for mapping low-resolution features as input and which produces an accurate boundary localization. This system is helpful for understanding road scene applications which is the ability to model, shape, and understand the relationship such as roads, buildings, cars, etc. The key component of SegNet is the decoder as it is smaller and easier to train as fully convolutional has been removed which were present inVGG16 [5]. To find an effective convolution network (ConvNets) for the accuracy of large-scale images. ConvNets has become a commodity in the computer field which improved architecture. The significantly more accurate architecture not only achieves the classification and localization task but is also applicable to other image recognition datasets. ConvNetsis of fixed size 224 x 224 RGB image at the initial stage. Then this input is passed to several layers of stack convolutions and as a result, 3 x 3 is filtered out. A stack of convolution layers is followed by three fully-connected layers which are used to soft-max the layer and all the hidden layers are nonlinear. A trained ConvNet and input image are given to the test phase.

Architecture

The architecture of YOLOV4 consists of various parts, which are: first is the input which is the training images that will be fed to the network and will be processed in batches by the GPU. The version$ architecture used darknet53 which consists of 53 convolutional layers and activation functions like Leaky ReLu. Followed by the activation function batch normalization making the system a fully connected model. The input will be divided into several grid squares and the center of the gird is responsible for detecting that object. The main components of yolov4 are the backbone and neck which are used to extract features from the square girds. It mostly concentres on the previous input with the current input before proceeding to the next layer. The role of the neck is to collect the feature maps from all the layers. The neck consists of three layers which are as follows:

1. SPP (Spatial Pyramid Pooling Layer): The SPP allows only the fixed-size features of whatever size our feature map is.
2. FPN (Feature Pyramid Network): This layer extracts a single-scale image of arbitrary size as the input and output are of the proportional size of the feature map.
3. PaNet: This layer incorporates the model to enhance the segmentation.

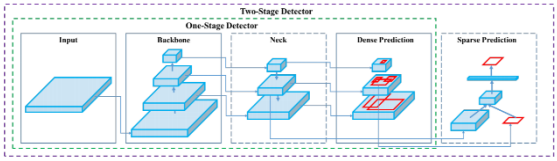


Fig 1. Architecture Diagram

The dense prediction layer is the last layer of the architecture which is also known as HEAD wherein the prediction is been made. Then the output is been shown on the console.

Methodology

The methodology used for implementing this system is the deep learning algorithm known as CNN. CNN (Convolutional Neural Network) is a class of deep learning networks used to analyze visuals. And to detect objects in real-time we have used YOLO (You Only Look Once) this algorithm only a single forward propagation.

Proposed System

It is not feasible to count/monitor crowd activities at various places like airports, events, or any other place by looking at them. Human identification is not an easy task in the case of a large and densely populated crowd. There is a need for an automated system that can provide us with some meaningful information from live or recorded

videos. To implement a crowd surveillance system, we have built two modules first crowd counting and the second one is of detecting activities like fire, gun, etc. The algorithm of the system is the same as the usual CNN-based system wherein images are given as input to the system. There are various steps involved in a crowd surveillance system which are as follows:

1. User Interface

To interact with the system, we have designed a web app using streamlit which is an open-source python framework. The user has to create his/her account first using the signup option and sign in to log into the system. On successful completion of the registration process, the main interface of the system will be shown which is the crowd counting option another one is to detect activities using the live camera of the system or the external system.

2. Image Annotation

In this step, we have marked the label on the images used for training purposes. For annotation of the image, we used a website known as roboflow. We have provided 500 plus images for training the crowd surveillance model. The image segmentation is of edge detection type. The raw are been labeled and class has been identified and the input images are now ready to be trained.

3. Train Model

The crowd surveillance model is trained using Google Colab which provides GPU for training the system models. First of all, we need to clone the Darknet repository which is an open-source neural network framework in C and CUDA and supports GPU computation. Then we have to upload a custom detector, labeled custom datasets, and a .cgf file. All the images are divided into 2 parts 90% of train data and 10% for test. And lastly, run the – '! make' command to build darknet and download the pre-trained weights generated by YOLOV4

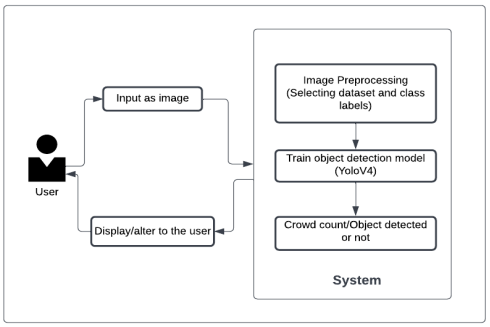


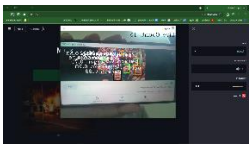
Fig 2. System Architecture

4. To detect object/crowd count

After completion of the registration and login procedure system will redirect users to the homepage wherein users need to select the crowd counting/object detection option. The system will ask for the permissions to enable the device camera. The camera will be capturing the gird of the screen to detect the activities and to count the density. The system will beep/alter the user if the grid reaches its threshold value.

Result

The user has to select the option of crowd count/object detection, then the device camera will be enabled. The camera will search for the grids from the camera frames and will detect the objects which the system pretends to detect or count the crowd. If the crowd count exceeds the maximum threshold value the system will alert the user through a beep and also if an object like a gun or activity like the fire is detected the system will start beeping. The output of the system is as follows: -



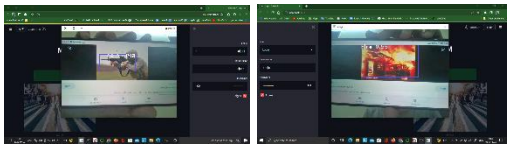


Fig 3. Output

Future Scope

As this system is used for security and surveillance purpose in the future, we can use the crowd surveillance system for various suspicious activities like fights, robbers, accident detection, etc. And to monitor crowd behavior and predict the next activity of the suspected object/person.

Conclusion

This system aimed to detect objects and to keep eye on crowd density used for security purposes. To alert the users if there are any abnormal objects are been detected and an interactive user interface for easy access to the system.

References

1. Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection”.
2. Marana, A. N., Cavenaghi, M. A., Olson, R. S., Drumond, F. L. “Real-time Crowd Density Estimation Using Images”, Proceeding of the IEEE.
3. Kohonen, T., "The Self-Organizing Map", Proceedings of the IEEE, vol.78, pp. 1464- 1480, 1990.
4. Jia Hong Yin, Sergio A. Velastin and Anthony C. Davies, “Image Processing Techniques for Crowd Density Estimation Using a Reference Image”, Proceeding of Second Asian Conference on Computer Vision (ACCV95), Singapore, 5-8 December, Vol. III, pp. 6-10.
5. V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481-2495, 1 Dec. 2017, DOI: 10.1109/TPAMI.2016.2644615.
6. S. Liu and W. Deng, "Very deep convolutional neural network-based image classification using small training sample size," 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 2015, pp. 730- 734, DOI: 10.1109/ACPR.2015.7486599.
7. R. S. de Moraes and E. P. de Freitas, "MultiUAV Based Crowd Monitoring System," in IEEE Transactions on Aerospace and Electronic Systems, vol. 56, no. 2, pp. 1332- 1345, April 2020, DOI: 10.1109/TAES.2019.2952420 [6] A Systematic Review of Location-Aware Schemes in the Internet of Things.
8. Haralick, R. M., "Statistical and Structural Approaches to Texture", Proceedings of the IEEE, vol. 67(5), pp. 786-804, 1979.
9. Yin J.H., Velastin S.A. and Davies A.C. (1995): “A Calibration Approach for Estimation of Crowd Density Using Image Processing”, accepted for 2nd International Workshop on Image and Signal Processing: Theory, Methodology, Systems, and Applications, 8-10 November, Budapest, Hungary.
10. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.
11. J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in CVPR, pp. 3431–3440, 2015.