Deep learning based dense object counting with density map

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

POTTI SAI YATEESH (Reg. No. 37110580)

PRADEEP S (Reg. No. 37110583)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600 119

MARCH - 2021

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **POTTI SAI YATEESH (Reg. No. 37110580) and PRADEEP S (Reg. No.37110583)** who carried out the project entitled "**DEEP LEARNING BASED DENSE OBJECT COUNTING WITH DENSITY MAP**" under my supervision from **August 2019** to **APRIL 2020**.

Internal guide
Dr.K.Ashok Kumar M.E., Ph.D.,

Head of the Department
Dr. S. Vigneswari, M.E., Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner **External Examiner**

DECLARATION

I **POTTI SAI YATEESH** hereby declare that the Project Report entitled "**DEEP LEARNING BASED DENSE OBJECT COUNTING WITH DENSITY MAP"** is done by me under the guidance of**, Dr.K.Ashok Kumar, M.E., Ph.D.,** Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering**.**

DATE:

PLACE: CHENNAI **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.K.Ashok Kumar, M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Object detection (OD) in crowded scenes is a challenging task since objects are densely distributed and partially overlapped. In this paper, we propose a novel OD method by fully exploring the information provided by the image and its estimated density map. Our proposed OD method consists of two main stages. Initial object locations are firstly computed based on object spatial distribution information obtained from the estimated density maps. Inspired by the human visual attention mechanism, a saliency map which offers object boundaries is then employed to accurately estimate the bounding boxes with the support of the estimated initial object locations. We propose the task to estimate the density map of objects from single image with unknown perspective map. We follow the recent progress in object counting through density map estimation. Object density map estimation is usually suffered from scale variance of objects caused by unknown perspective. In addition, the background and irrelevant objects in the image lead to artifacts in the resulting density maps, which build up the error when heat maps are needed by aggregating density maps.

TABLE OF CONTENTS

	ABSTRACT	**v**
	LIST OF FIGURES	**viii**
CHAPTER No.	**TITLE**	**PAGE No.**
1.	**INTRODUCTION**	**1**
	1.1 INTRODUCTION	1
	1.2 RESEARCH AND SIGNIFICANCE	2
2.	**LITERATURE SURVEY**	**4**
	2.1 LITERATURE SURVEY-1	4
	2.2 LITERATURE SURVEY-2	*8*
	2.2 LITERATURE SURVEY-3	11
3.	**AIM AND SCOPE**	**15**
	3.1 AIM OF THE PROJECT	15
	3.2 SCOPE	15
	3.3 OBJECTIVES	15
	3.4 ADVANTAGES	16
	3.5 CROWD COUNTING	16
	3.5.1 DENSITY CROWD COUNTING	16
	3.5.2 REGRESSION BAASED COUNTING	16
	3.5.3 DENSITY MAP BASED COUNTING	17
	3.6 ABLATION STUDIES	18
	3.6.1 DENSITY MAPS	18
	3.6.2 KERNEL STUDIES	18
	3.6.3 REGULARIZATION	18
	3.6.4 SELF-ATTENTION MODULE	18
	3.6.5 LOCAL COUNTING PERFORMANCE	19
	3.6.6 VISUALIZATION	19

4.	**METHODOLOGY**	21
	4.1 EXISTING SYSTEM	22
	4.3 SYSTEM FUNCTIONALITIES	23
	4.4 HARDWARE SPECIFICATION	24
	4.5 SOFTWARE SPECIFICATION	24
	4.5.1 BACKEND TECHNOLOGIES	24
	4.5.2 FRONTEND TECHNOLOGIES	24
	4.6 MODULES	25
	4.6.1 DATASET	25
	4.6.2 LOAD WEIGHTS	26
	4.6.3 PREDICTION	27
	4.7 DESCRIPTION OF DIAGRAM	30
	4.8 ARCHITECTURAL DESIGN	31
	4.9 COUNTING BY DETECTION	32
	4.9.1 MONOLITHIC DETECTION	32
	4.9.2 PART BASED DETECTION	33
	4.9.3 SHAPE MATCHING	34
5.	**RESULTS AND DISSCUSSION**	36
6.	**CONCLUSION AND FUTUREWORK**	40
	REFERENCES	41
	APPENDIX	42
	A. SOURCE CODE	**48**
	B. JOURNAL PAPER	
	C. PLAGARISM REPORT	

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
4.1	BLOCK DIAGRAM	30
4.2	ACTIVITY DIAGRAM	31
5.1	INPUT SCREEN SHOT	36
5.2	GRAYSCALE IMAGE OF GIVEN INPUT	37
5.3	DENSITY CALCULATING IMAGE	38
5.4	DENSITY OF THE GIVEN IMAGE	39

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

Dense object counting tasks, including crowd counting, vehicle counting, and general object counting, aim to estimate the number of objects in the image. Counting tasks have practical usage for understanding crowded scenes. For example, crowd counting can be used to prevent accidents caused by overcrowding and estimate the crowd flows in station. And vehicle counting can be used for traffic management on roads or in parking lots. General object counting is useful for the management of goods in the supermarket, farms and factories.

Although counting tasks are important and useful, the real usage is still limited since dense object counting is challenging. One of the main challenges is scale variation. Since the scale of people varies dramatically in images and across different images, it is difficult to extract features for density regression. Another challenge is the occlusions among people since only a small part of each person may be visible in crowd images. Complex backgrounds may also hurt the counting performance, and the domain gap between scenes in datasets and the real world scenes also limits the usage of counting algorithms.

Current state-of-the-art methods use crowd density maps to achieve superior counting performance [1, 2, 3]. Density maps are an *intermediate* representation, where the sum over any region in the density map indicates the number of people in the region. First, the density maps are generated from the dot annotations, where each dot indicates a person's location. Second, given the input image, algorithms are designed to predict the density map which is then summed to obtain the count.

In practice, the method for generating the density maps is crucial for crowd counting. Improperly generated density maps may dramatically hurt the counting performance – the choice of the kernel bandwidth or kernel shape used to generate the density map is often dataset dependent, and such choices often do not work across different datasets.

In the era of deep learning, we may consider current density maps as a *hand-crafted intermediate representation*, which is used as a target for training deep networks to count. From the standpoint of end-to-end training, these hand- designed intermediate representations may not be optimal for the particular network architecture and particular dataset.

1.2 RESEARCH AND SIGNIFICANCE

In this, we call these two steps density map *generation* and density map *estimation*, respectively. Most works focus on density map estimation and ignore density map generation. Many different deep networks have been proposed to improve density map estimation, e.g., using different kernel sizes or image pyramids to handle scale variations, or using context or prior informationto handle occlusions. Although density map estimation is well-studied, the generation of density maps is often overlooked and uses handcrafted designs without adequate investigation and analysis. The simplest approach to obtain a density map is to convolve the annotation dot map with a Gaussian with fixed width, i.e., place a Gaussian on each dot. Other works scale the Gaussian bandwidth according to the scene perspective, or adaptively use the local congestion level (or distance to nearest neighbors) uses human-shaped kernels, composed of two Gaussians, but is less popular since the body of the person is often occluded in crowd images.

Current state-of-the-art methods use crowd density maps to achieve superior counting performance . Density maps are an *intermediate* representation, where the sum over any region in the density map indicates the number of people in the region. First, the density maps are generated from the dot annotations, where each dot indicates a person’s location. Second, given the input image, algorithms are designed to predict the density map , which is then summed to obtain the count.

In practice, the method for generating the density maps is crucial for crowd counting. Improperly generated density maps may dramatically hurt the counting performance – the choice of the kernel bandwidth or kernel shape used to generate the density map is often dataset dependent, and such choices often do not work across different datasets. In the era of deep learning, we may consider current density maps as a *hand-crafted intermediate representation*, which is used as a target for training deep networks to count.

From the standpoint of end-to-end training, these hand- designed intermediate representations may not be optimal for the particular network architecture and particular dataset.

In this paper, we take the first step towards learnable density map representations, which are jointly trained with the density map estimator (counter). In particular, we first generate a unique normalized kernel for each object (e.g. a person or a vehicle) given an image as the input. The scale and shape of kernels are automatically learned during the joint optimization with a counter.

CHAPTER 2
LITERATURE SURVEY

2.1 LITERATURE SURVEY-1

Title	Benchmark Data and Method for Real-Time People Counting in Cluttered Scenes Using Depth Sensors
Authors	Shijie Sun , Naveed Akhtar , Huansheng Song, Chaoyang Zhang, Jianxin Li , and Ajmal Mian
Published Year	2019
Efficiency	• Proved its efficiency in large volumes of extremely imbalanced data. • Construct a comprehensive feature vector. • Outperforms from previous schemes in terms of recall, precision, and time complexity.
Drawbacks	• Poor Application Performance. • Not based on real-time datasets. • Cannot exploit the new feature space.

Description	Vision-based automatic counting of people has widespread applications in intelligent transportation systems, security, and logistics. However, there is currently no large-scale public dataset for benchmarking approaches on this problem. This paper fills this gap by introducing the first real-world RGB-D people counting dataset (PCDS) containing over 4500 videos recorded at the entrance doors of buses in normal and cluttered conditions. It also proposes an efficient method for counting people in real-world cluttered scenes related to public transportations using depth videos. The proposed method computes a point cloud from the depth video frame and re-projects it onto the ground plane to normalize the depth information. The resulting depth image is analyzed for identifying potential human heads. The human head proposals are meticulously refined using a 3D human model. The proposals in each frame of the continuous video stream are tracked to trace their trajectories. The trajectories are again refined to ascertain reliable counting. People are eventually counted by accumulating the head trajectories leaving the scene. To enable effective head and trajectory identification, we also propose two different compound features. A

	thorough evaluation on PCDS demonstrates that our technique is able to count people in cluttered scenes with high accuracy at 45 fps on a 1.7-GHz processor, and hence it can be deployed for effective real-time people counting for intelligent transportation systems. We concatenate the above mentioned four geometric features into a vector in R13. Notice that, although we do consider varied areas of IH in the above mentioned features, the compound feature only accounts for the information that is local to individual rectangles. In the real-world scenarios, the relative locations of the rectangles (that we suspect to contain human heads) can provide useful information about a bounded object being a human head or not. Therefore, we further define NRDF to account for this additional information. For each rectangle in FH , we compute NRDF as a vector in the 3D- space that is directed towards the center of the rectangle from the center of its nearest rectangle in our current set of head proposals. This feature is further illustrated in Fig. 6. The resulting NRDF $\in$ R3 is concatenated with the above mentioned feature vector to finally arrive at our compound feature vector in R16.

	Background removal is a major task in many surveillance related problems. For our approach, reliable background subtraction is necessary for the success of subsequent processing of video frames. Therefore, we separately analyze the performance of our method for this task. We use the popular Gaussian mixture-based background segmentation method (MOG) and the K-nearest neighbors (KNN) based method to benchmark our technique. We note that other approaches for background subtraction also exist, however the selected baseline methods are chosen for their well-established effectiveness for the depth videos.To analyze the performance of our method for human head identification, we first manually labeled 12,148 rectangle proposals in height images for people entering the buses as ‘heads’ and ‘non-heads’. These proposals were generated automatically by the method in Sec. We can argue that the employed classifiers are able to identify human heads in the proposals successfully. We note that the classification performance depicted by is better for the people exiting buses than for the people entering buses. The reason behind this phenomenon is that while providing the ground truth we only

	labeled those proposal rectangles as 'heads' that bounded complete human heads. For the case of people entering the buses, many half- heads appeared in the frames due to queuing of people on bus doors. On scrutiny, we found that most of those heads resulted in false positive identifications in our experiment. However, this is not problematic for the overall approach because the final results rely more strongly on tracking of heads on multiple frames, and the half- heads eventually transform into complete heads in the subsequent video frames. We also provide the details of precision, recall and the f1-scores for our head identification experiment.

LITERATURE SURVEY-2

Title	Scale Driven Convolutional Neural Network Model for People Counting and Localization in Crowd Scenes
Authors	SALEH BASALAMAH, SULTAN DAUD KHAN , AND HABIB ULLAH
Published Year	2019
Efficiency	• Outstanding Learning Capabilities • Effectively improve prediction accuracy. • Reduces the consumption of hardware resources

Drawbacks	• Can’t learn relation between factors. • Do not take account of the influence features • Maximizes the complexity of the problem
Description	Counting and localization of people in videos consisting of low density to high density crowds encounter many key challenges including complex backgrounds, scale variations, nonuniform distributions, and occlusions. For this purpose, we propose a scale driven convolutional neural network (SD-CNN) model, which is based on the assumption that heads are the dominant and visible features regardless of the density of crowds. To deal with the problem of different scales of heads in different regions of the videos, we annotate a set of heads in random locations of the videos to develop a scale map representing the mapping of head sizes. We then extract scale aware proposals based on the scale map which are fed to the SD-CNN model acting as a head detector. Our model provides a response matrix rendering accurate head positions via nonmaximal suppression. For experimental evaluations, we consider three standard datasets presenting low density to high density crowd scenes. Our proposed SD-CNN model outperforms the state-of-the-art methods in terms of both frame-level and pixel- level analyses. After generating scale-aware proposals, the next step is to classify each proposal into two classes, i.e, head and background. Our detection network follows the classical R-CNN mode [12] and instead of using selective search for proposal generation, we use scale-aware proposals. Before feeding to the network, we extend the bounding box of each proposal by a small margin and then image patch corresponding to each proposal is resized to Fit the input layer of the CNN. For the head detection, we keep the square- like aspect ratios < 2 for all bounding boxes. The classical R-CNN is based on AlexNet architecture which is

	pretrained on ImageNet [9] dataset. In addition to AlexNet, we used several other alternatives.From the experiment, we noticed that VGGS slightly outperforms AlexNet but was slower in both training and testing. In this section we discuss both qualitative and quantitative analysis of the results obtained from the experiments. We evaluate our SD- CNN framework using three publicly available datasets, UCSD dataset ,WorldExpo'10 and UCF-CC-50 . The summary of the datasets is described in Table 1. Generally, these datasets are annotated in a way that can only be useful for evaluating the performance of regression models. Typically, in these datasets, every individual pedestrian is annotated with a dot in the scene. These dot annotations are not suitable for training our SD-CNN model or other detection based methods. Therefore, we annotated each pedestrian with a bounding box that cover whole body of pedestrian. In the same way, we also annotated the head of each pedestrian using the bounding box. After annotation, we then trained different models discussed in Section IV on Titan Xp with learning rate at 0.01 and decrease it by a factor of 10 after the validation error reaches saturation point. In this section, We evaluate both qualitatively and quantitatively the localization performance of our framework. In order to quantify the localization error, we associate the center of estimated bounding box with the ground truth location (single dot) through 1-1 matching strategy. We then compute Precision and Recall at various thresholds and report the overall localization performance in terms of area under the curve. In order to estimate the location, we use the same density maps generated by state-of-the-art methods followed by non-maxima suppression algorithm. The results are reported in Table 5. It is obvious that our proposed

	model presents higher Precision and Recall rates as compared to the stateof- the-art methods. These results attribute to the fact that our model generates scale-aware proposals that capture wide range of head sizes in each image. It can also be observed that all other methods present lower rates for UCF-CC-50 dataset as compared to WorldExpo'10 and UCSD datasets. This is due to the fact the UCF-CC-50 dataset contains more dense images with heavy occlusions as compared to World- Expo'10 and UCSD datasets. We also show some qualitative results of our proposed method in Figure 5. From the Figure 5, it is obvious that the sample images from the UCSD dataset represent low density scene. The sample images taken from two different scenes of World- Expo'10 dataset represent medium densities and the images from UCF-CC-50 represent relatively more complex and extreme high density scenes.

2.3 LITERATURE SURVEY-3

Title	Device-Free People Counting in IoT Environments: New Insights, Results and Open Challenges
Author	Iker Sobron,Manuel Velez
Published Year	2018
Advantages	Low Deployment Cost Quick Calculation Time Very easy to implement for multi-class problem
Drawbacks	A lot of pre-development education Existing system is Opportunistic and uncontrollable Difficulties to obtain better performance
Description	In the last years multiple Internet of Things (IoT) solutions have been developed to detect, track, count and identify human activity from people that do not carry any device nor participate actively in the detection process. When WiFi radio receivers are employed as sensors for device-free human activity recognition, channel

	quality measurements are preprocessed in order to extract predictive features towards performing the desired activity recognition via machine learning (ML) models. Despite the variety of predictors in the literature, there is no universally outperforming set of features for all scenarios and applications. However, certain feature combinations could achieve a better average detection performance compared to the use of a thorough feature portfolio. Such predictors are often obtained by feature engineering and selection techniques applied before the learning process. This manuscript elaborates on the feature engineering and selection methodology for counting device-free people by solely resorting to the fluctuation and variation of WiFi signals exchanged by IoT devices. We comprehensively review the feature engineering and ML models employed in the literature from a critical perspective, identifying trends, research niches and open challenges. Furthermore, we present and provide the community with a new open database with WiFi measurements in several indoor environments (i.e. rooms, corridors and stairs) where up to 5 people can be detected. This dataset is used toexhaustively assess the performance of different ML models with and without feature selection, from which insightful conclusions are drawn regarding the predictive potential of different predictors across scenarios of diverse characteristics. Time statistics have been widely employed for feature construction in many application areas. Common statistical metrics such as mean, variance, moments and the like are combined to yield multiple predictors. Following this engineering approach, the authors in [24] defined a new feature, coined as the coefficient of variation of phase, where the ratio between the standard deviation and mean of the i-th CSI subcarrier phase \Hi(n) is computed within a time window. Human motion is detected when the averaged ratio of the coefficient of variation of phase falls within a predefined confidence interval. In the coefficient of variation of the normalized CSI amplitudes is computed following the same criterion. In order to quantify the multi- path propagation conditions due to the presence of an intruder, a new metric is derived. This feature consists of the ratio between the kurtosis and the mean of

	the coefficient of variation of CSI amplitudes. A precalibrated detection threshold is necessary for each scenario. Additionally, in the Rician K-factor was postulated as a possible predictor, however, it was neglected due to the lack of time and phase synchronization on commodity WiFi devices in order to extract accurate information for the Rician estimator. It has been mentioned previously, this architectural scheme is usually adopted for decreasing the volume of data transmitted from the IoT nodes to the Cloud. However, due to the fact that nodes can be connected to the Cloud via wired links, the adoption of Edge Computing is not motivated here by data traffic limitations, but rather by the required computational effort and by possible latencies that data processing can yield, which might be of relevance for certain time-critical applications demanding the counting service (e.g. intrusion detection). As a result, raw CSI data can be processed at each end device such that some elaborated features or testing decisions are eventually transmitted to the Cloud for further processing and/or storage. In this regard two computation levels are proposed: a) feature extraction and selection can be performed at each IoT node, so that the selected feature set is transmitted and fed to the ML model in the Cloud, whose (re-)training algorithm leverages a higher amount of computational resources; b) both FS and learning/testing are performed at the IoT devices, after which predictions are delivered to the Cloud and served therefrom to applications demanding the people counting service. We delve into the design of a device-free people counting IoT framework, which can be regarded as a cyber-physical system in the IoT context. Apart from the regular application of the WiFi systems, IoT nodes can act as sensors by collecting data about the surrounding radio environment. The sensed CSI, necessary for channel decoding in the wireless communication chain, can be converted into meaningful information about the human activity in a monitored area. WiFi IoT devices are usually connected to a local network where one or several routers grant connectivity to Internet. As a result, sensing data can be shared, gathered and

	analyzed in a cloud platform in order to provide ubiquitous device-free people counting services over IoT deployments.

CHAPTER 3
AIM AND SCOPE

3.1 AIM OF PROJECT

The primary aim of this proposed system sets the detection threshold and adjusting the camera. The detection results below the threshold, which is usually set from 0.2 to 0.4, will not be counted. For the sake of simplicity, we use the default value of 0.2 in this work. In the actual scene, the camera should be adjusted to the appropriate height and angle.

3.2 SCOPE

The proposed system overcomes the above issues with a novel real-time people counting approach dubbed YOLO-PC (YOLO based People Counting). In the proposed system, after the special pre-treatment, adaptive segmentation and feature extraction for the peoplecounting data, the feature vector is used as the inputs of the trained YOLO to classify and statistics of the total number of the people .

3.3 OBJECTIVES

The proposed system is a dynamic background subtraction module is first considered to segment moving objects from each captured video frame. In order to overcome light variations, a dynamic threshold value associated with detecting regions of interest from the differentiated image is iteratively calculated according to the distributions of background and foreground pixels in each frame. After obtaining the foreground regions, four states including new, leaving, merged and split are assigned to the detected moving objects according to their appearances in the current frame. In particular, targets identified as states of merge and split further pass through backward tracking for relieving the occlusion effects by investigating the centroid distances among objects in the previous frame. Finally, targets in four states are tagged to yield the results of people tracking and counting.

3.4 ADVANTAGES

- Low computational complexity with high efficiency
- Give more precise on the estimation
- Scale and confidence priors are discovered automatically
- Extremely fast counting strategy with high accuracy
- High performance is achieved

3.5 CROWD COUNTING

Traditional crowd counting algorithms are based on individual detection and tracking but these methods do not work well for dense scenes. Thus, global regression based methods are proposed to avoid the detection of individuals by directly regressing the number of people However, global regression ignores the spatial distribution of people, and thus density map based methods are proposed to further predict the spatial density map in images [8] and have achieve the outstanding performance.

3.5.1 DENSITY CROWD COUNTING

Individual detection and tracking based counting algorithms rely on the detection algorithms that do not work well under congested scenes. Most of these algorithms detect human heads and shoulders by detection or tracking algorithms. propose to count the number of people by detecting human heads and shoulders based on foreground segmentation.

3.5.2 REGRESSION BASED COUNTING

To avoid explicit detection of individuals, regression methods are proposed to estimate the number of people directly from low-level features like texture, color, and gradient. Chan *et al.* [12] propose to count by directly regressing from global features to crowd number using Gaussian process regression. A prior distribution is proposed in [13] to estimate homogeneous crowds. Multiple features are combined in [14] to improve the performance of crowd counting. However, the performance of traditional counting algorithms are still limited due to scale variation and occlusion in crowd images.

3.5.3 DENSITY MAP BASED COUNTING

Density map based methods are currently the most popular ap- proach to crowd counting since the performance can be dramat- ically improved by utilizing spatial information. Density maps are typically generated by blurring dot maps in which each dot indicates a person in an image [8]. Since density maps are inter- mediate representations, most algorithms generate density maps beforehand by convolving the dot maps with Gaussian kernels with either fixed or adaptive bandwidths. Then, different network architectures are designed to handle various challenges, such as scale changes, improving the quality of density maps, encoding more contextual information, or adapting to new scenarios.

To handle with scale variation of people,[4] proposes a multi-column neural network (MCNN) where each column has different kernel sizes to extract multi-scale features. Similarly, SANet [16] proposes to extract multi-scale features with scale aggregation modules, while Kang and Chan [5] propose to use image pyramid to deal with scale variation in crowd counting. Instead of fusing multi-scale features, switch-CNN [17] proposes to select a proper column with appropriate receptive field. A tree- structured CNN is proposed to handle the diversity of people in crowds [18]. [19] proposes a hierarchical encoder-decoder framework to encode multi-scale features, while [20] proposes an attention based framework to filter background and [21] proposes a novel feature fusion strategy.

Another way to improve the quality of density maps and the performance of crowd counting is to use refinement-based algorithms. Ranjan *et al.* [22] propose a two stage counting framework in which the high-resolution density map is estimated based on the low-resolution density map gen- erated in the initial stage, while Sam and Babu [23] propose a feedback mechanism to refine the predicted density map.

3.6 ABLATION STUDIES

3.6.1 DENSITY MAPS

We first compare the effectiveness of different traditional density maps and our generated density maps. For fixed bandwidth kernels, the bandwidths are set to 4 and 16.The learned density maps from ADMG/KDMG generally out- perform manually generated density maps including fixed kernels and adaptive kernels. KDMG achieves superior performance over ADMG, since the former preserves the true count in the learned density map.

3.6.2 KERNEL STUDIES

The effect of kernel size *k* on the density maps produced by KDMG is investigated on ShTech A and B, and the results are presented in Figure 6 (a). For ShTech A, smaller kernels tend to yield better performance since the crowd is dense. In ShTech B, the best performance is achieved when kernel size is 7 since the crowd is less dense. We thus set kernel size to 5 and 7 for ShTech A and B, respectively. On the other crowd datasets, we use *k*=5 since they have similar average head size to ShTech A. We also set *k*=5 for the vehicle and general object counting datasets.

3.6.3 REGULARIZATION

The cosine similarity is used as a regularizer for spatial con- sistency. The effect of the weight in is investigated on ShanghaiTech A and ShanghaiTech B, and the results are shown in Figure 6 (b). On ShanghaiTech A, larger weights tend to generate better performance since people are close to each other in ShanghaiTech A. Under this circumstance, spatial regularization is effective to learn the spatial density distribution.

3.6.4 SELF-ATTENTION MODULE

To evaluate the effectiveness of the self-attention module in ADMG, we compare using the self-attention module with three variants: “image-att” generates attention from the input image;“direct fusion” directly fuses without attention; “naive- fusion” directly sums all density

maps. As in the fusion is more effective with self-attention (“self-att”) than with the input image (“image-att”). The possible reason is that the crowd information is directly obtainable from the density maps, whereas this information needs to be decoded and interpreted from the image, which introduces additional complexity and noise.

3.6.5 LOCAL COUNTING PERFORMANCE

To investigate the local counting performance, we evaluate the frameworks based on GAME metric on UCF-QNRF. The result is shown in Table 12. The local counting performance of ADMG is worse than the traditional method, while KDMG achieves better local counting performance than the baseline CSRNet. KDMG has better local performance because it composites a set of kernels with fixed $k \times k$ size, which keeps the local density regions smooth. In contrast, ADMG uses a series of convolution layers that tends to make the density regions more compact , which cause errors near boundaries of the GAME image patches.

3.6.6 VISUALIZATION

To better understand the generation framework, we compare the learned density maps on different datasets with traditional density maps . For small-bandwidth density maps and ADMG density maps (second column), the density only appear on part of the object, which results in sparse density maps. For density maps generated by adaptive kernels , the density is too smooth for sparse objects. The density maps generated by fixed kernel with bandwidth are similar to KDMG density maps , which can better cover the whole objects with less leakage to the background.

We next visualize the learned individual kernels for KDMG and fixed kernels with bandwidth 16 in Figure 8. Overall, the learned kernels are more flat than fixed kernels especially for very dense images in UCF-QNRF. Since the counters use max-pooling layers which will produce translation invariant features, it will be better to have the same density value for shifted patches.

We also fit the profiles of the learned kernels from KDMG to quadratic functions. In particular, we randomly selected 70 images for each dataset. Then, the images are split into several regions based on the height or width, and the average coefficient is calculated for each region. The correlation statistics and p-value are calculated using data from the 70 images. The magnitude of the quadratic coefficient (the coefficient is always negative) can

be used to represent the flatness of a kernel. For each dataset, the magnitude of the quadratic coefficient of a kernel versus its x- coordinate (SKU-110K dataset) or y-coordinate (other datasets) is shown in Figure 10, where the origin (0,0) is the top-left corner of the image. We also plot the best fit line (via linear regression), and test whether there is a significant correlation between the coefficient magnitude and the y-coordinate (or x-coordinate), i.e., whether the line's slope is significantly different from zero.

When the images contain dense crowds under perspective ef- fects (as in ShTech A, UCF-QNRF, PUCPR+, and TRANSCOS), we found that the quadratic coefficient's magnitude is negatively correlated with kernel's y-coordinate. In other words, for these scenes, the curvature of the kernel adapts to placement of peo- ple in the scene. For small people far from the camera (small y-coordinates), the quadratic coefficient has larger magnitude, yielding a sharper kernel. In contrast, for large people close to the camera (large y-coordinates), the coefficient has smaller magnitude, yielding a flatter kernel.

A possible explanation for using flatter kernels closer to the camera is that flat structures are easier to predict through the max-pooling layer (as discussed above), and the kernels are less likely to overlap since close people are more spread out in the image space. On the opposite, far people are more likely to have overlapping kernels, and thus a sharper kernel is used so that the densities in overlapping regions does not get too large, compared to the peak value on the person. For the overhead-view images or side-on view , there was no significant correlation between the kernel shape and its location, mainly because the object density and inter-object distance do not change significantly within the image. Finally, for ShTech B, there was a small positive correlation
between the coefficient magnitude and the y-coordinate people closer to the camera (large y-coordinates) had slightly sharper kernels (larger coefficient magnitude),
compared to people further away. Perhaps this is a side-effect caused by the large variations in camera orientations and heights for images in ShTech B (compared to the other datasets that are more homogenous), which also explains the small effect size (r=0.099). In summary, we found that the kernel-shape changes within the image, baon properties of the scene, and thus KDMG is able to optimize the kernel shape to match the scene and the particular density estimation network

CHAPTER 4
METHODOLOGY

4.1 EXISTING SYSTEM

Existing system follow a Bid generative adversarial networks framework as the basis of our method, which contains the encoder, generator (decoder) and discriminator. The structure can encode the selected image into a specific distribution, which is the necessary condition to interpolate two target images. The core framework of BiGAN indicates that the encoder and generator (decoder) will be trained separately. To train a general deep learning model to explore dynamic patterns of density maps, it is important to create available datasets. The property of samples in training data should contain different sizes, shapes, and time intervals, and they should also provide smooth and continuous dynamic information. As a visualization system, we want to balance the conciseness and computational capabilities and to introduce as few interactions as possible. The dynamic change may be difficult to be identified, particularly when there are slight changes and noises. We enhance the visual effect to improve our system by using blending, field representation, and sampling. The next parts introduce how to apply these technologies to implement a general framework.

The existing system works best with color (RGB) images containing dense crowd i.e. more than 500 heads in an image. The existing system comprises of four major components. The first component is a CNN-based head detector that provides a sparse location of heads and their sizes in the images. The second component is a feature classifier the image is first divided into equal-size rectangular patches, which are categorized as crowd or not crowd by a Support Vector Machine SVM classifier on speeded up robust features SURF features. The third component is a regression module that estimates the head count for each crowd patch based on its spatial coordinates and estimated head sizes. While notcrowd patches obviously have zero counts, it is possible that the head detector may fail to detect any head in some of the crowd patches. This is resolved by the fourth component in which the counts for these crowd patches are estimated by the spatially dependent weighted average of the counts from the neighboring eight patches. The final step is to sum all the individual patch estimates to get a total count for the entire image. The existing system does not assume that the crowd fills the entire image.

Since we follow a patch approach, some of the patches may not contain any crowd. It is important to identify such patches to avoid over estimation. To address this, we introduce a binary crowd/not-crowd classifier. The existing system only produces sparse detections, which leads to many patches having no heads detected in them at all while SVM may classify them as crowd patch. If there are no heads detected in a patch by CNN, the estimated head size would be estimate

4.2 SYSTEM CAPABILITIES

Clearly defined requirements are essential signs on the road that leads to a successful project. They establish a formal agreement between a client and a provider that they are both working to reach the same goal. High-quality, detailed requirements also help mitigate financial risks and keep the project on a schedule. According to the Business Analysis Body of Knowledge definition, requirements are a usable representation of a need. Creating requirements is a complex task as it includes a set of processes such as elicitation, analysis, specification, validation, and management. In this article, we'll discuss the main types of requirements for software products and provide a number of recommendations for their use.

Python is a free, open-source programming language. Therefore, all you have to do is install Python once, and you can start working with it. Not to mention that you can contribute your own code to the community. Python is also a cross-platform compatible language. So, what does this mean? Well, you can install and run Python on several operating systems. Whether you have a Windows, Mac or Linux, you can rest assure that Python will work on all these operating systems.

Python coding style comprises physical lines as well as logical lines or statements. A physical line in a Python program is a sequence of characters, and the end of the line terminates the line sequence as opposed to some other languages, such as C and C++ where a semi-colon is used to mark the end of the statement. A logical line, on the other hand, is composed of one or more physical lines. The use of a semi-colon is not prohibited in Python, although it's not mandatory. The NEWLINE token denotes the end of the logical line. A logical line that only contains spaces, comments, or tabs are called blank lines and they are ignored by the interpreter.

As we saw that in Python, a new line simply means that a new statement has started. Although, Python does provide a way to split a statement into a multiline statement or to join multiple statements into one logical line. This can be helpful to increase the readability of the statement. Following are the two ways to split a line into two or more lines: Explicit Line Joining, Implicit Line Joining.

Comments in any programming language are used to increase the readability of the code. Similarly, in Python, when the program starts getting complicated, one of the best ways to maintain the readability of the code is to use Python comments. It is considered a good practice to include documentations and notes in the python syntax since it makes the code way more readable and understandable to other programmers as well, which comes in handy when multiple programmers are simultaneously working on the same project.

4.3 SYSTEM FUCTIONALITIES

The code can only explain how it does something and not why it does that, but Python comments can do that. With Python comments, we can make documentations for various explanations in our code itself. Comments are nothing but tagged lines of codes which increase the readability of a code and make it self-explanatory. There are different ways of creating comments depending on the type of comment we want to include in our code. Following are different kinds of comments that can be included in our Python program.

Python has the documentation strings (or docstrings) feature which is usually the first statement included in functions and modules. Rather than being ignored by the Python Interpreter like regular comments, docstrings can actually be accessed at the run time using the dot operator. It gives programmers an easy way of adding quick notes with every Python module, function, class, and method. To use this feature, we use triple quotes in the beginning of the documentation string or comment and the closing triple quotes at the end of the documentation comment. Docstrings can be one-liners as well as multi-liners. Unlike some programming languages that support multiline comments, such as C, Java, and more, there is no specific feature for multiline comments in Python. But that does not mean that it is totally impossible to make multiline comments in Python. There are two ways we can include comments that can span across multiple lines in our Python code.

Python Block Comments: We can use several single line comments for a whole block. This type of comment is usually created to explain the block of code that follows the Block comment. Python

Block comment is the only way of writing a real comment that can span across multiple lines. It is supported and preferred by Python's PEP8 style guide since Block comments are ignored by Python interpreter or parser. However, nothing is stopping programmers from using the second 'non-real' way of writing multiline comments in Python which is explained below. Using Docstrings: Docstrings are largely used as multiline comments in Python by many programmers since it is the closest thing to having a multiline comment feature in Python.

One of the most crucial part of learning any programming language is to understand how data is stored and manipulated in that language. Users are often inclined toward Python because of its ease of use and the number of versatile features it provides. One of those features is dynamic typing. In Python, unlike statically typed languages like C or Java, there is no need to specifically declare the data type of the variable. In dynamically typed languages such as Python, the interpreter itself

predicts the data type of the Python Variable based on the type of value assigned to that variable. While it is not wrong to use docstrings when we need to make multiline comments, it is important to keep in mind that there is a significant difference between docstrings and comments. Comments in Python are totally ignored by the Python Interpreter, while docstrings, when used inside the Python function, can be accessed at the run time.

4.4 HARDWARE SPECIFICATION

Hardware : Minimum Requirement.
Disk Space : 32 GB or more,10 GB or more for Foundation Edition.
Processor :1.4 GHz 64 bit Memory 512 MB.
Display : (800 × 600) Capable video adapter and monitor.

4.5 SOFTWARE SPECIFICATION

4.5.1 Backend Technologies

- Python
- Numpy
- Sci-learn
- Eclipse IDE

4.5.2 FRONTEND TECHNOLOGIES

- Web Technologies
- Bootstrap

4.6 MODULES

4.6.1 DATASET

Data visualization is a technique that uses an array of static and interactive visuals within a specific context to help people understand and make sense of large amounts of data. The data is often displayed in a story format that visualizes patterns, trends and correlations that may otherwise go unnoticed. Data visualization is applied in practically every field of knowledge. Scientists in various disciplines use computer techniques to model complex events and visualize phenomena that cannot be observed directly, such as weather patterns, medical conditions or mathematical relationships. Data visualization provides an important suite of tools and techniques for gaining a qualitative understanding.

The train-test split procedure is appropriate when you have a very large dataset, a costly model to train, or require a good estimate of model performance quickly. The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred to as the training dataset. The second subset is not used to train the model; instead, the input element of the dataset is provided to the model, then predictions are made and compared to the expected values. This second dataset is referred to as the test dataset. Train Dataset: Used to fit the machine learning model. Test Dataset: Used to evaluate the fit machine learning model. The objective is to estimate the performance of the machine learning model on new data: data not used to train the model. by default, the program ignores the original order of data. It randomly picks data to form the training and test set, which is usually a desirable feature in real-world applications to avoid possible artifacts existing in the data preparation process. To disable this feature, simply set the shuffle parameter as False (default = True).

The skimage.io image package is used to read the image from the file. Rescale operation resizes an image by a given scaling factor. The scaling factor can either be a single floating point value, or multiple values - one along each axis. Resize serves the same purpose, but allows to specify an output image shape instead of a scaling factor.

4.6.2 LOAD WEIGHTS

Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set. Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process. So to sum up, the idea of PCA is simple — reduce the number of variables of a data set, while preserving asmuch information as possible. Data dimensionality reduction is part of data preprocessing in the entire intrusion detection system. Various studies have shown that not only serious redundancy among the characteristic dimensions of network data but also high correlation exists among the data of each dimension. Redundancy and correlation between feature dimensions not only reduce the response time of the intrusion detection system but also affect the learning efficiency of the training process. Therefore, dimensionality reduction of high-dimensional data is particularly necessary. Reducing the dimension of the dataset can not only improve the learning performance of the detection system; it can also reduce the redundancy of the dataset.. A residual block consists of two or three sequential convolutional layers and a separate parallel identity (repeater) shortcut connection, which connects the input of the first layer and the output of the last one. Each block has two parallel paths. The left path is similar to the other networks, and consists of sequential convolutional layers + batch normalization. The right path contains the identity shortcut connection (also known as skip connection). The two paths are merged via an element-wise sum. That is, the left and right tensors have the same shape and an element of the first

tensor is added to the element of the same position of the second tensor. The output is a single tensor with the same shape as the input. In effect, we propagate forward the features learned by the block, but also the original unmodified signal. The network can decide to skip some of the convolutional layers thanks to the skip connections, in effect reducing its own depth. The residual blocks use padding in such a way that the input and the output of the block have the same dimensions.

An inception block starts with a common input, and then splits it into different parallel paths (or towers). Each path contains either convolutional layers with a different-sized filter, or a pooling layer. In this way, we apply different receptive fields on the same input data. At the end of the inception block, the outputs of the different paths are concatenated.

4.6.3 PREDICTION

ImageDataGenerator class allows allow rotation of up to 90 degrees, horizontal flip, horizontal and vertical shift of the data. We need to apply the training standardization over the test set. ImageDataGenerator will generate a stream of augmented images during training. We will define Exponential Linear Unit (ELU) activation functions A single fully-connected layer after the last max pooling. The padding='same' parameter. This simply means that the output volume slices will have the same dimensions as the input ones.

Batch normalization provides a way to apply data processing, similar to the standard score, for the hidden layers of the network. It normalizes the outputs of the hidden layer for each mini-batch (hence the name) in a way, which maintains its mean activation value close to 0, and its standard deviation close to 1. We can use it with both convolutional and fully connected layers. Networks with batch normalization train faster and can use higher learning rates.

There are different types of AI/ML models throughout com- puter science history. Among these many AI/ML models, the algorithms behind these models work differently. In general, one can say that ML algorithms can be categorized under three different parts, namely, supervised, unsupervised, and reinforce- ment learning. Since we consider supervised ML algorithms in this paper, labeled data becomes crucial in this context. As a result, the data generation part becomes an important aspect of this paper. Considering all these, we generate a labeled data set for 5G cellular networks. While selecting the fields of the proposed data set, we are inspired by the 5G specifications pub- lished by the 3GPP consortium.Traditional density map generation approaches treat generation and estimation as two separate steps. Usually, a dot annotation map (the input) is convolved with a Gaussian kernel to form a smooth heat map which is called a density map. When the images contain dense crowds under perspective effects .

we found that the quadratic coefficient's magnitude is negatively correlated with kernel's y-coordinate. In other words, for these scenes, the curvature of the kernel adapts to placement of peo- ple in the scene. For small people far from the camera (small y-coordinates), the quadratic coefficient has larger magnitude, yielding a sharper kernel. In contrast, for large people close to the camera, the coefficient has smaller magnitude, yielding a flatter kernel.

A possible explanation for using flatter kernels closer to the camera is that flat structures are easier to predict through the max-pooling layer (as discussed above), and the kernels are less likely to overlap since close people are more spread out in the image space. On the opposite, far people are more likely to have overlapping kernels, and thus a sharper kernel is used so that the densities in overlapping regions does not get too large, compared to the peak value on the person. For the

overhead-view images or side-on view , there was no significant correlation between the kernel shape and its location, mainly because the object density and inter-object distance do not change significantly within the image. Finally, for ShTech B, there was a small positive correlation between the coefficient magnitude and the y-coordinate people closer to the camera had slightly sharper kernels (larger coefficient magnitude), compared to people further away. Perhaps this is a side-effect caused by the large variations in camera orientations and heights for images in ShTech B (compared to the other datasets that are more homogenous), which also explains the small effect size . In summary, we found that the kernel-shape changes within the image, based on properties of the scene, and thus KDMG is able to optimize the kernel shape to match the scene and the particular density estimation network. We next visualize the learned individual kernels for KDMG and fixed kernels with bandwidth

Overall, the learned kernels are more flat than fixed kernels especially for very dense images in UCF-QNRF. Since the counters use max-pooling layers which will produce translation invariant features, it will be better to have the same density value for shifted patches Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set. Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process. So to sum up, the idea of PCA is simple — reduce the number of variables of a data set, while preserving asmuch information as possible. Data dimensionality reduction is part of data preprocessing in the entire intrusion detection system. Various studies have shown that not only serious redundancy among the characteristic dimensions of network data but also high correlation exists among the data of each dimension. Redundancy and correlation between feature dimensions not only reduce the response time of the intrusion detection system but also affect the learning efficiency of the training process.

The two paths are merged via an element-wise sum. That is, the left and right tensors have the same shape and an element of the first tensor is added to the element of the same position of the second tensor. The output is a single tensor with the same shape as the input. In effect, we propagate forward the features learned by the block, but also the original unmodified signal. The network can decide to skip some of the convolutional layers thanks to the skip connections, in effect reducing its own depth. The residual blocks use padding in such a way that the input and the output of the block have the

same dimensions. An inception block starts with a common input, and then splits it into different parallel paths (or towers). Each path contains either convolutional layers with a different-sized filter, or a pooling layer. In this way, we apply different receptive fields on the same input data. At the end of the inception block.

4.7 DESCRIPTION OF DIAGRAM

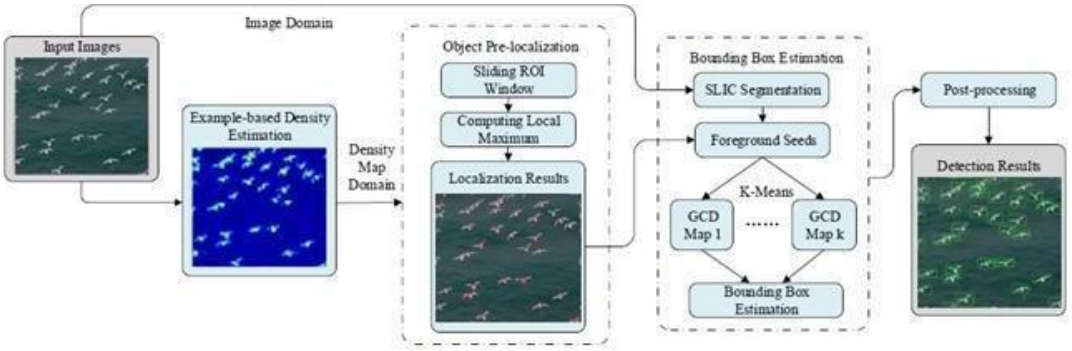


Fig 4.1: BLOCK DIAGRAM

An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components. What is a diagram? What are the types of diagrams for architecture? The Dragon1 open EA Method makes it very clear: if a diagram does not show a concept, principle or part of a principle, it is NOT an architecture diagram, because it does not show (a part of) the architecture. There are many kinds of architecture diagrams, like a software architecture diagram, system architecture diagram, application architecture diagram, security architecture diagram, etc. For system developers, they need system architecture diagrams to understand, clarify, and communicate ideas about the system structure and the user requirements that the system must support. It's a basic framework can be used at the system planning phase helping partners understand the architecture, discuss changes, and communicate intentions clearly.

4.8 ARCHITECTURAL DESIGN

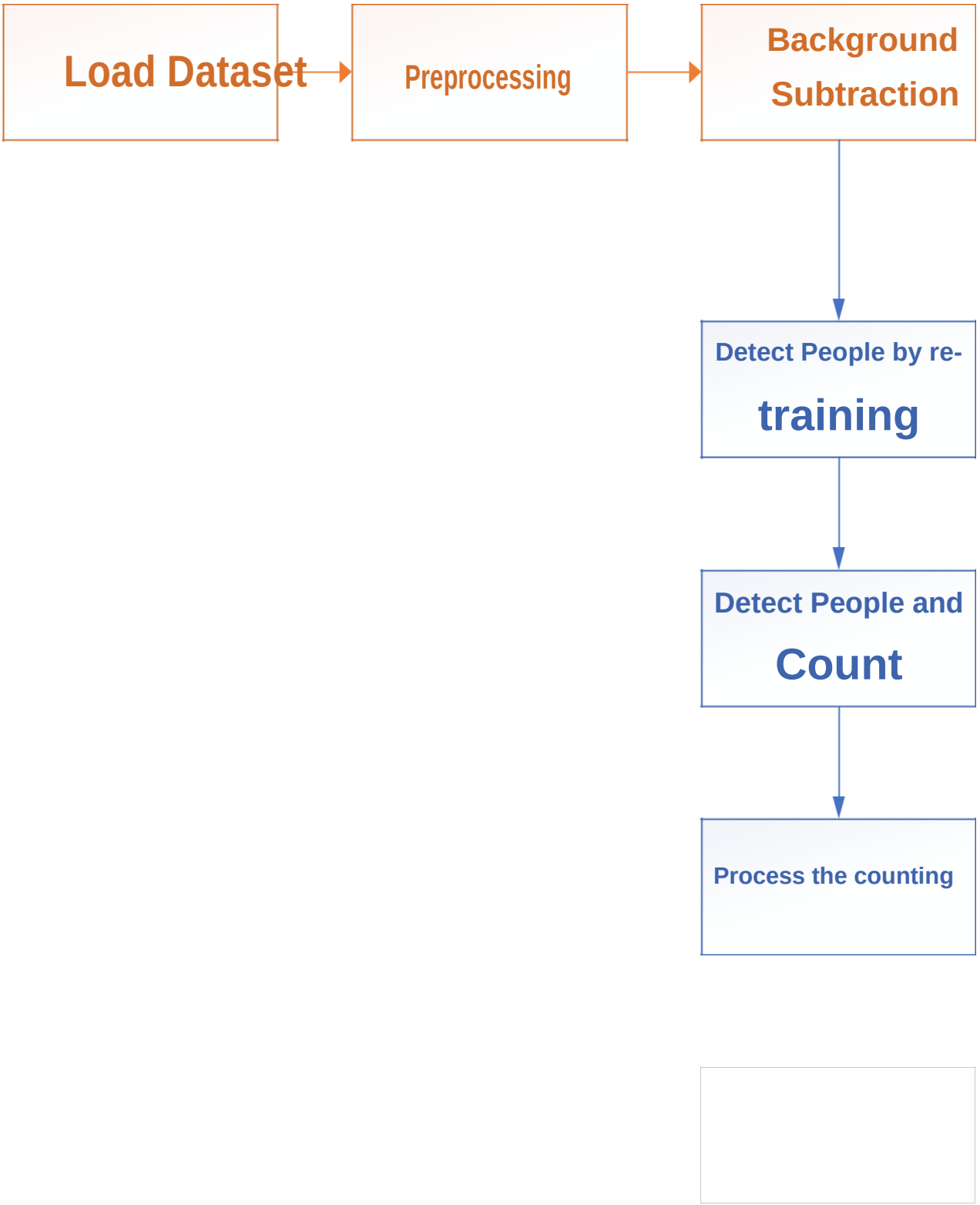


Fig 4.2: Activity Diagram

4.9 COUNTING BY DETECTION

4.9.1 MONOLITHIC DETECTION

It trains the classifier using the full-body appearance that's available in the training images using typical features such as Haar wavelets, gradient-based features such as a histogram of oriented gradient (HOG), etc. Learning approaches such as SVMs, random forests have been used that employ a sliding window approach. But these are limited to sparse crowds. To deal with dense crowds, part-based detection is often more useful. CrowdNet is a combination of deep and shallow, fully convolutional neural networks. This feature helps in capturing both the low-level and high-level features. The dataset is augmented to learn scale-invariant representations. The deep network is similar to the well-known VGG-16 network. It captures the high-level semantics needed for crowd counting and returns the density maps . To train a general deep learning model to explore dynamic patterns of density maps, it is important to create available datasets. The property of samples in training data should contain different sizes, shapes, and time intervals, and they should also provide smooth and continuous dynamic information. As a visualization system, we want to balance the conciseness and computational capabilities and to introduce as few interactions as possible. The dynamic change may be difficult to be identified, particularly when there are slight changes and noises. We enhance the visual effect to improve our system by using blending, field representation, and sampling. The next parts introduce how to apply these technologies to implement a general framework. We describe a procedure for initializing the structure of a mixture model and learning all parameters. Parameter learning is done by constructing a latent SVM training problem. We train the latent SVM using the coordinate descent approach described in together with the data-mining and gradient descent algorithms that work with a cache of feature vectors. The desired output of an object detection system is not entirely clear. The goal in the PASCAL challenge is to predict the bounding boxes of objects. In our previous work [17] we reported bounding boxes derived from root filter locations.

Yet detection with one of our models localizes each part filter in addition to the root filter. Furthermore, part filters are localized with greater spatial precision than root filters. It is clear that our original approach discards potentially valuable information gained from using a multiscale deformable part model.

4.9.2 PART BASED DETECTION

Rather than taking the whole human body, this technique considers a part, say head or shoulders and applies a classifier to it. Head solely isn't sufficient in estimating the presence of a person reliably, therefore head + shoulder is the preferred combination in this technique. A new dataset of images is used comprising of 1198 images with 330,000 annotations to train the model. A Multi-Column CNN architecture maps the image to its crowd density map. This model utilizes filters with various receptive fields. The features learned by each column CNN are adaptive to variations in people/head size due to perspective effect or image resolution. Here, the density map is computed accurately based on geometry-adaptive kernels. The crowd density variations are taken into consideration to improve the accuracy and localisation of the predicted crowd count. It relays patches from a grid within a crowd scene to independent CNN regressors on a switch classifier. A particular regressor is trained on a crowd scene patch if the performance of the regressor on the patch is the best. A switch classifier is trained alternately with the training of multiple CNN regressors to correctly relay a patch to a particular regressor. Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through fintech applications like cloud computing.

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a

lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It's achieving results that were not possible before.

In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

4.9.3 SHAPE MATCHING

Ellipses are considered to draw boundaries around humans, and then a stochastic process is used to estimate the number and shape configuration. The Nanonets API allows you to build Object Detection models with ease. You can upload your data, annotate it, set the model to train and wait for getting predictions through a browser based UI without writing a single line of code, worrying about GPUs or finding the right architectures for your deep learning models. Several techniques have been used to come up with the right solution to the above question. Initially, computer scientists developed basic machine learning and computer vision algorithms like detection, regression, and density-based approaches to predict crowd density and density maps. Nonetheless, these methods are also bound with various challenges such as variations in scale and perspective, occlusions, non-uniform density, etc. Later, when Convolutional Neural Networks proved its capability in various computer vision tasks by overcoming these failures, researchers shifted their attention to it, in order to exploit its features in deriving the algorithms. When the crowd has loads of people stacked up at one place, then it's termed as the dense crowd, and when the people are sparsely placed, it's a sparse crowd. The methods and techniques which we would be exploring a deal with both dense and sparse crowd. Sparse crowd counting is relatively easy in comparison to Dense crowd counting, hence the algorithms need to work harder for a dense crowd. Counting by detection is not very accurate when the crowd is dense and the background clutter is high. To overcome these problems, counting by regression is used wherein the features extracted from the local image patches are mapped to the count. Here, neither segmentation nor tracking of individuals is involved. One of the earliest attempts involves extracting the low-level features such as edge details, foreground pixels, and then apply regression modelling to it by mapping the features and the count. A majority of the previous approaches ignored the spatial information persisting in the images.

However, this approach focuses on the density by learning the mapping between local features and object density maps, thereby incorporating spatial information in the process. This avoids learning each individual separately and therefore tracks a group of individuals at a time. The mapping described could be linear or nonlinear.

CHAPTER 5
RESULTS AND DISCUSSION

As a result of this system we are going to say about the people counting in this project and in this we have used deep learning to run this code and in this we have used convolutional neural network algorithm is used for counting purpose and for remaining modules we have written the code in the code in the program.

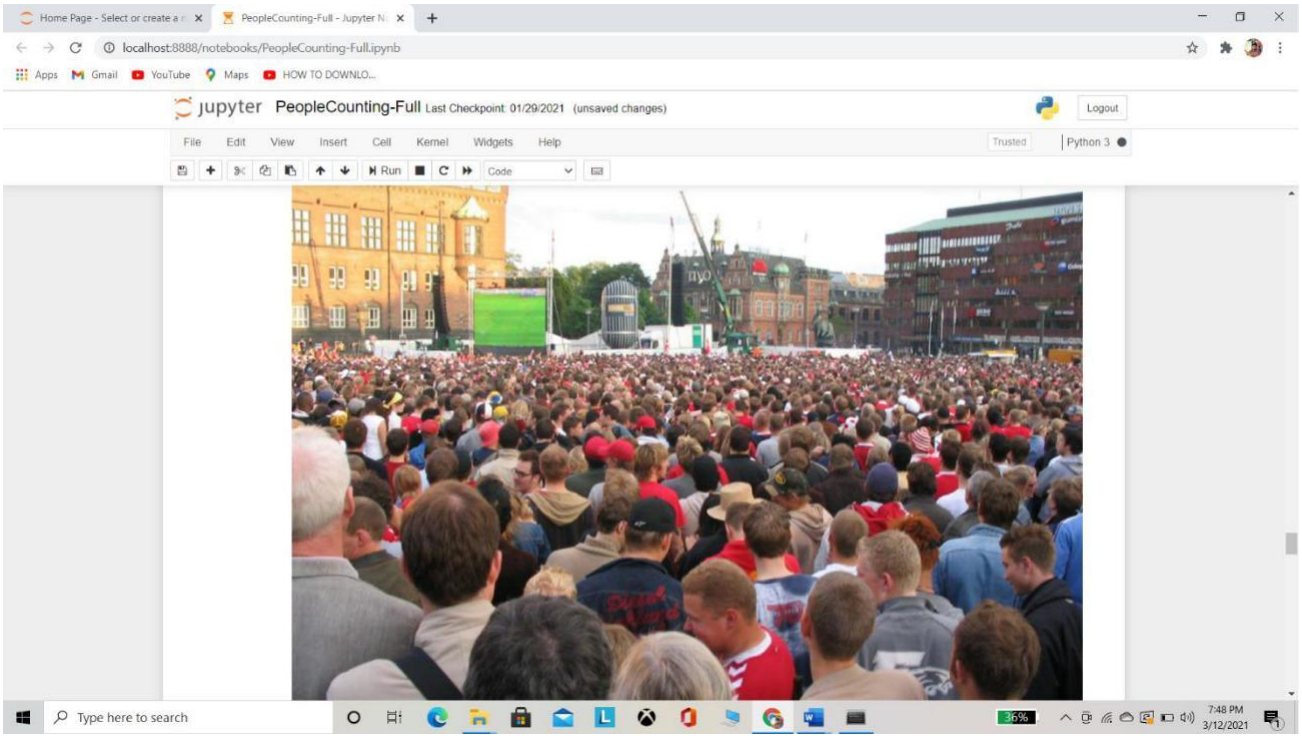


Fig 5.1 INPUT SCREEN SHOT

The image tells about the input of the the project from this we will calculate the density of the persons in the image and it will give most accurate than so many softwares in this it is the output of theour project by this we will get the output density of an image.

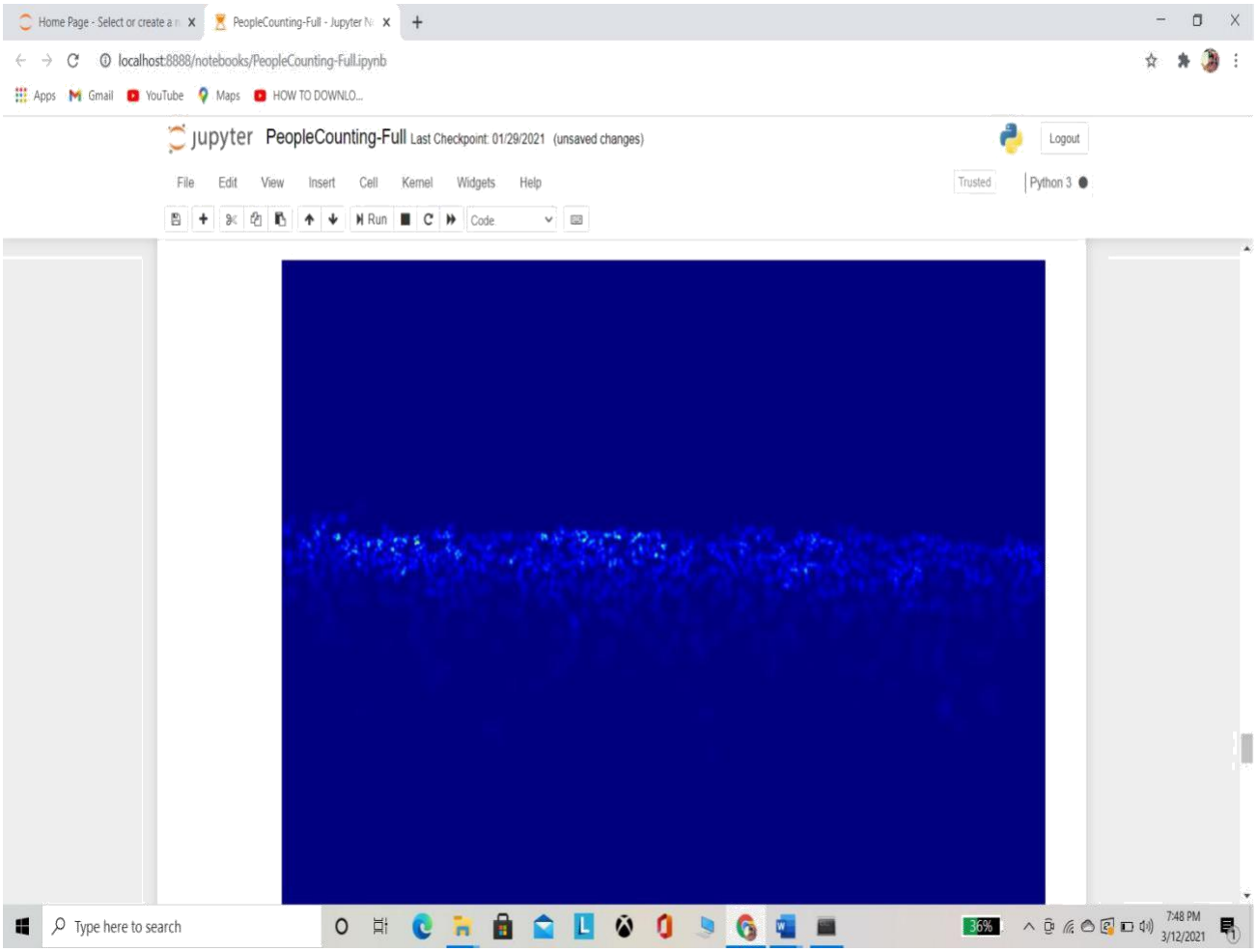


FIG 5.2 GRAYSCALE IMAGE OF GIVEN INPUT IMAGE

This screenshot explains about the grayscale image which it would be converted to count the people in the image of the screenshot 4.1 from which it converted into machine understanble Language and then it will calculate the density of the peopleand it will give the result by converting the grayscale we use coding to convert the normal image into an grayscale image by this we can convert the image into grayscale then it will count the density of the people and estimate it.

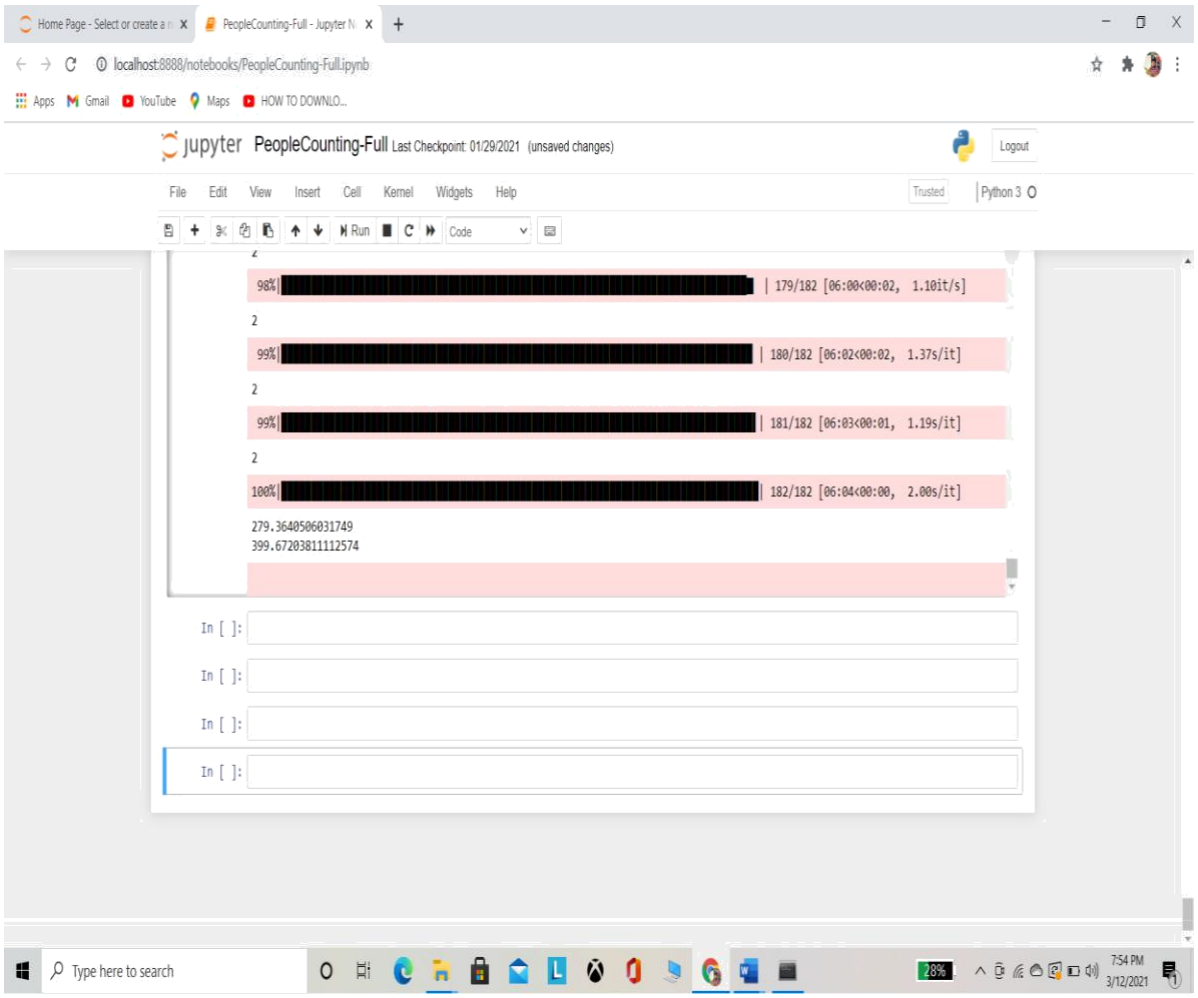


FIG 5.3 DENSITY CALCULATING IMAGE

This images shows about the density calculation and the mean squared error and in which how much accurate the result it is giving the output and it will load in percerntages and it will calculate. The errors and density of the each input it will calculate and it will show the output density as density equals to and it will show the output in this it will give much accurate than the other methods and in this convolutional neural network algorithm is used to calculate the density of the every images.

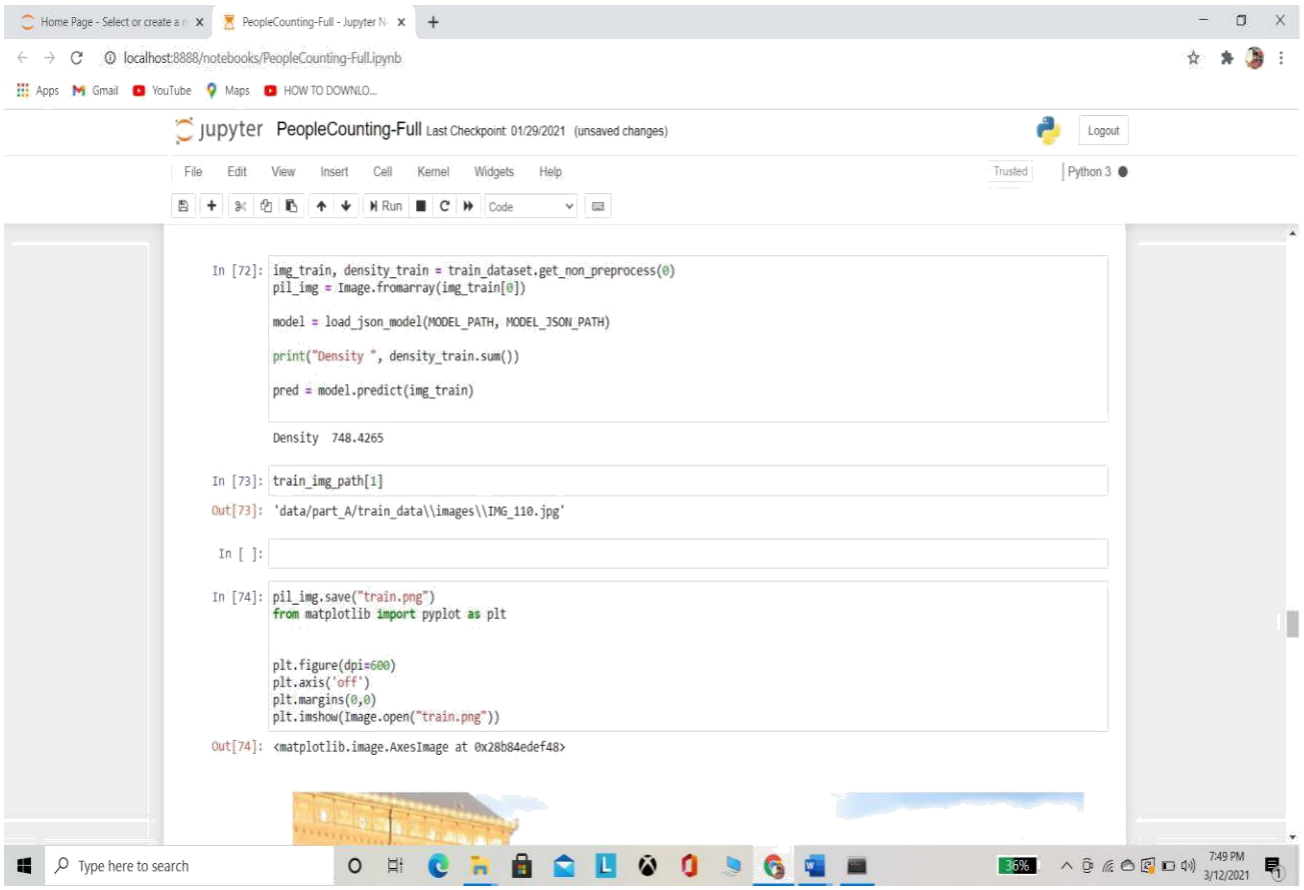


FIG 5.4 DENSITY OF THE OF THE GIVEN INPUT

This is the output screen of the project in this it will show the density of an image And then it will calculate the errors.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

We propose an accurate small object detection method by exploring from both the image and its estimated density map. Our method takes advantage of the object spatial distribution information in density map but avoids its drawback of obscure object boundary. Although our method is trained by dotted annotation datasets, the estimated bounding box fits the object accurately due to the sufficient boundary information provided by saliency map. The proposed focus prior map can be used as the focus feature for image analysis or used as an effective prior to improve the performance of salient object detection.

6.2 FUTURE WORK

Deep learning models have often achieved increasing success due to the availability of massive datasets and expanding model depth and parameterisation. However, in practice factors like memory and computational time during training and testing are important factors to consider when choosing a model from a large bank of models. Training time becomes an important consideration particularly when the performance gain is not commensurate with increased training time as shown in our experiments. Test time memory and computational load are important to deploy models on specialised embedded devices, for example, in AR applications. From an overall efficiency viewpoint, we feel less attention has been paid to smaller and more memory, time efficient models for real-time applications such as road scene understanding and AR. This was the primary motivation behind the proposal of SegNet, which is significantly smaller and faster than other competing architectures, but which we have shown to be efficient for tasks such as road scene understanding.

REFERENCES

[1] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun, “Saliency optimization from robust background detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR). IEEE, 2014, pp. 2814–2821.

[2] Na Tong, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang, “Salient object detection via bootstrap learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR). IEEE, 2015, pp. 1884–1892.

[3] Na Tong, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang, “Salient object detection via bootstrap learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR). IEEE, 2015, pp. 1884–1892.

[4] Tobias Franke, Paul Lukowicz, and Ulf Blanke, “Smart crowds in smart cities: real life, city scale deployments of a smartphone based participatory crowd management platform,” Journal of Internet Services and Applications, vol. 6, no. 1, pp. 27,2015.

[5] Y. Wang, Y. Zou, J. Chen, X. Huang, C. Cai, "Example-based visual object counting with a sparsity constraint," in IEEE International Conference on Multimedia and Expo, 2016.

[6] Kang, Z. Ma, and A. B. Chan, “Beyond counting: com- parisons of density maps for crowd analysis tasks-counting, detection, and tracking,” *IEEE Trans. CSVT*, 2018.

[7] N. Liu, Y. Long, C. Zou, Q. Niu, L. Pan, and H. Wu, “Ad- crowdnet: An attention-injective deformable convolutional network for crowd understanding,” in *CVPR*, June 2019.

[8] C. Liu, X. Weng, and Y. Mu, “Recurrent attentive zooming for joint crowd counting and precise localization,” in *CVPR*, 2019, pp. 1217–1226.

A. Source Code


```
"import numpy as np "
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "metadata": {},
   "outputs": [],
   "source": [
    "import pandas as pd "
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 4,
   "metadata": {},
   "outputs": [],
   "source": [
    "from tensorflow.python.keras.utils.data_utils import Sequence"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 5,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "'1.18.1'"
      ]
     },
     "execution_count": 5,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "np.__version__"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 6,
   "metadata": {},
   "outputs": [],
   "source": [
    "import glob"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 7,
   "metadata": {},
   "outputs": [],
   "source": [
    "import os"
   ]
  },
  {
```

```
 "cell_type": "code",
 "execution_count": 8,
 "metadata": {},
 "outputs": [],
 "source": [
  "import keras"
 ]
},
{
 "cell_type": "code",
 "execution_count": 9,
 "metadata": {},
 "outputs": [],
 "source": [
  "from tqdm import tqdm"
 ]
},
{
 "cell_type": "code",
 "execution_count": 10,
 "metadata": {},
 "outputs": [],
 "source": [
  "from sklearn.model_selection import train_test_split"
 ]
},
{
 "cell_type": "code",
 "execution_count": 11,
 "metadata": {},
 "outputs": [],
 "source": [
  "import json"
 ]
},
{
 "cell_type": "code",
 "execution_count": 12,
 "metadata": {},
 "outputs": [],
 "source": [
  "from keras import backend as K"
 ]
},
{
 "cell_type": "code",
 "execution_count": 13,
 "metadata": {},
 "outputs": [],
 "source": [
  "from tensorflow.python.keras.models import load_model, model_from_json"
 ]
},
{
 "cell_type": "code",
 "execution_count": 14,
 "metadata": {},
 "outputs": [],
 "source": [
  "from tensorflow.python.keras.applications.vgg16 import VGG16, preprocess_input"
 ]
},
{
```

```
  "cell_type": "code",
  "execution_count": 15,
  "metadata": {},
  "outputs": [],
  "source": [
   "from tensorflow.python.keras.losses import mean_squared_error"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 16,
  "metadata": {},
  "outputs": [],
  "source": [
   "from tensorflow.python.keras.models import Model"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 17,
  "metadata": {},
  "outputs": [],
  "source": [
   "from tensorflow.python.keras.layers import Conv2D, UpSampling2D,
BatchNormalization, Activation\n"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 18,
  "metadata": {},
  "outputs": [],
  "source": [
   "from tensorflow.python.keras.initializers import RandomNormal "
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 19,
  "metadata": {},
  "outputs": [],
  "source": [
   "from tensorflow.python.keras.optimizers import SGD, Adam"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 20,
  "metadata": {},
  "outputs": [],
  "source": [
   "import tensorflow as tf"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 21,
  "metadata": {},
  "outputs": [],
  "source": [
   "from tensorflow.python.keras import backend as K"
  ]
 },
```

```
{
 "cell_type": "code",
 "execution_count": 22,
 "metadata": {},
 "outputs": [],
 "source": [
  "import cv2"
 ]
},
{
 "cell_type": "code",
 "execution_count": 23,
 "metadata": {},
 "outputs": [],
 "source": [
  "import h5py"
 ]
},
{
 "cell_type": "code",
 "execution_count": 24,
 "metadata": {},
 "outputs": [],
 "source": [
  "import PIL.Image as Image"
 ]
},
{
 "cell_type": "code",
 "execution_count": 25,
 "metadata": {},
 "outputs": [],
 "source": [
  "from tensorflow.python.keras.preprocessing.image import ImageDataGenerator"
 ]
},
{
 "cell_type": "code",
 "execution_count": 26,
 "metadata": {},
 "outputs": [],
 "source": [
  "from matplotlib import pyplot as plt\n",
  "from matplotlib import cm as CM"
 ]
},
{
 "cell_type": "code",
 "execution_count": 27,
 "metadata": {},
 "outputs": [],
 "source": [
  "tf.compat.v1.disable_eager_execution()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 28,
 "metadata": {},
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
```

```
   "text": [
    "['ground-truth', 'ground-truth-h5', 'images']\n"
   ]
  }
 ],
 "source": [
  "print(os.listdir(\"data/part_A/train_data\"))"
 ]
},
{
 "cell_type": "code",
 "execution_count": 29,
 "metadata": {},
 "outputs": [],
 "source": [
  "DATA_PATH = \"data/part_A/train_data\""
 ]
},
{
 "cell_type": "code",
 "execution_count": 30,
 "metadata": {},
 "outputs": [],
 "source": [
  "TEST_PATH = \"data/part_A/test_data\""
 ]
},
{
 "cell_type": "code",
 "execution_count": 31,
 "metadata": {},
 "outputs": [],
 "source": [
  "MODEL_PATH = \"csrnet_shanghaitechA_task1.model\""
 ]
},
{
 "cell_type": "code",
 "execution_count": 32,
 "metadata": {},
 "outputs": [],
 "source": [
  "MODEL_JSON_PATH = \"csrnet_shanghaitechA_task1.json\""
 ]
},
{
 "cell_type": "code",
 "execution_count": 33,
 "metadata": {},
 "outputs": [],
 "source": [
  "def create_training_image_list(data_path):\n",
  "    \n",
  "    DATA_PATH = data_path\n",
  "    image_path_list = glob.glob(os.path.join(DATA_PATH, \"images\", \"*.jpg\"))\n",
  "    return image_path_list"
 ]
},
{
 "cell_type": "code",
 "execution_count": 34,
 "metadata": {},
 "outputs": [],
```

```
"source": [
 "def get_train_val_list(data_path):\n",
 "    DATA_PATH = data_path\n",
 "    image_path_list = glob.glob(os.path.join(DATA_PATH, \"images\", \"*.jpg\"))\n",
 "    train, val = train_test_split(image_path_list, test_size=0.1)\n",
 "\n",
 "    print(\"train size \", len(train))\n",
 "    print(\"val size \", len(val))\n",
```

B.JOURNAL PAPER

Deep learning based dense object counting with density map

POTTI SAI YATEESH [1] , PRADEEP S [2] , ASHOK KUMAR K[3]

[1][2] UG Student, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

[3]Associate professor, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

ABSTRACT: Assessing the quantity of people in a public spot gives valuable data to video-based reconnaissance and checking applications. On account of sideways camera arrangement, tallying is either accomplished by distinguishing people or by measurably setting up relations between estimations of basic picture highlights to the quantity of individuals. Ongoing individuals tallying from video records is a fundamental structure block for some applications in shrewd urban areas. In the current System, a head indicator can be utilized to appraise the spatially fluctuating head size, which is the key component utilized in our mind tallying methodology. We influence the best in class convolutional neural organization for the meager head identification in thick group. After sub-partitioning the picture into rectangular patches, we first utilize a SURF highlight based SVM double classifier to name each fix as group/not-swarm and kill all not-swarm patches. In the current framework, task as a rule experiences numerous issues, similar to the absence of continuous handling of the recorded recordings or the event of mistakes because of superfluous individuals being tallied. The proposed framework defeats the above issues with a novel continuous individuals tallying approach named YOLO-PC (YOLO based People Counting). In the proposed framework, after the extraordinary pre-treatment, versatile division and highlight extraction for the human checking information, the element vector is utilized as the contributions of the prepared YOLO to characterize and measurements of the absolute number individuals.

Keywords: People Counting , YOLO-PC, CNN

1 INTRODUCTION

Dependably assessing the quantity of individuals present in a public spot (for example road, shopping center, tram station) over the long run might be of basic significance for both wellbeing and monetary reasons. For example, a specific number of people in a given setting may mirror an abnormal and possibly risky circumstance. Then again, observing the quantity of individuals in a territory of a shopping center will give important data not exclusively to streamlining the working hours of the shops yet additionally for assessing the appeal of the shopping region. With the approach of savvy cameras and the expanding opportunities for robotized reconnaissance and checking, the mechanization of the human tallying task additionally turns out to be mechanically doable. In this work we modify the current article location framework YOLO by proposing the purported YOLO-PC(People Counting dependent on YOLO). YOLO-PC broadens the first YOLO framework utilizing a profound learning way to deal with accomplish a higher exactness in individuals checking. Contrasted with other existing article location frameworks, for example, R-CNN, Fast R-CNN, and Faster R-CNN, YOLO has been picked as the base technique in our YOLO-PC, as a result of its low calculation overhead and its capacity to identify objects continuously. Regarding Performance continuously, YOLO-

PC retrains a profound convolutional neural organization to distinguish individuals at in excess of 40 fps (outlines each second) with the help of a GPU. As to individuals tallying measure, the picture is partitioned into a 9*9 networks and limits are noted. This prompts more recognized frameworks and a more noteworthy normal certainty esteem. Besides the YOLO-PC picks an alternate limit territory as indicated by the genuine application situation to include individuals in a contrapuntal way which further improves the tallying exactness. YOLO-PC can likewise overlook the immaterial people who might be in announcements or other unimportant territories Experimental outcomes show that YOLO-PC can tally individuals rapidly with a higher exactness at the passageway or ways out of spots like lifts, arenas, shopping centers and so on

The initial step comprises in setting the identification edge and changing the camera. The location results underneath the limit, which is generally set from 0.2 to 0.4, won't be tallied. For straightforwardness, we utilize the default estimation of 0.2 in this work. In the real scene, the camera ought to be acclimated to the proper tallness and point. In the subsequent advance, we distinguish individuals through re-preparing a convolutional neural organization. YOLO partitions the picture into a 7*7 framework and for every network cell predicts two jumping boxes just as the certainty esteem for those crates. We accept that this division isn't adequate and we point that our calculation will be more effective in distinguishing individuals to accomplish higher tallying exactness. At the end of the day, YOLO-PC works better with more recognized boxes and higher certainty esteems. To this end, YOLO-PC utilizes 9*9 lattice and 3 bouncing boxes. We set up three arrangements of examinations of 4 minutes video each utilizing various edges (i.e., 0.2, 0.3 and 0.4). The acquired outcomes are promising since YOLO-PC distinguishes more boxes and accomplishes higher certainty esteems for those containers contrasted with YOLO. All the more explicitly, YOLO-PC distinguishes in excess of 10% of the containers, the certainty normal worth is in excess of 50% higher when the limit is 0.2. The third step comprises in recognizing and checking individuals dependent on a suitable limit determination. YOLO-PC chooses at least one lattice cells as the territory limit from 243(9*9*3) cells and picks an alternate limit as per the real circumstance. In the event that individuals turn left by some place, the limit of the left region of the video ought to be chosen, the estimation of the limit is around 113 and individuals through that limit will be tallied. Essentially, if individuals turn directly by some place, the estimation of the limit is around 129.

In the event that individuals go straight by some place, the estimation of the limit is around 121. YOLO-PC can be more precise in recognizing the progression of individuals as some garbage obstruction, like individuals in the boards and disconnected backs, can be overlooked as a result of the limit choice. The checking data of the past advance is currently handled in the fourth step. In the chose limit zone, the cases number collects and continually refreshes, we allude to this number as S. The estimation of S at a second t in a video addresses the quantity of identified individuals at that point, which is exact. The estimation of S in a timeframe addresses the complete number of identified individuals. Since it requires some investment for individuals to move in the limit region, the estimation of S is rehashed, in other words, a similar individual has been distinguished commonly. As per the examinations, each individual has been distinguished around multiple times when going through the chose limit zone, so the anticipated number is S/18 at the default edge. In the fifth and last advance, we yield the constant individuals tallying data. YOLO-PC can straightforwardly show the realtime individuals including data in the video pictures, including the current number, FPS, certainty esteem and so forth YOLO-PC can likewise save continuous data and proceed to refresh,

and afterward yield them through certain interfaces.

II RELATED WORK

The problem of counting people has been handled by many examination endeavors, utilizing a few heterogeneous methodologies. In any case, an investigation in important writing uncovers that each approach is on a basic level persuaded by the uncommon states of the current issue. Numerous strategies utilize face/individual discovery calculations, while others join mass/object location and following plans. The issue is normally alluded to as "individuals tallying" or "swarm checking" when it is applied to huge, open zones. Regular applications center chiefly around reconnaissance or traffic. The intrigued peruser may discover an investigation zeroing in on checking. To assess person on foot stream checking, ***Hsieh et al. [1]*** introduced a framework that utilizes Kinect. Their methodology depended on morphological handling and extraction of associated parts to extricate districts of interest. They guaranteed that their framework delivered continuous outcomes, having wonderful precision. ***Ryan et al. [2]*** proposed a methodology that pre-owned neighborhood includes and worked on independent closer view mass sections. Thusly, they got a complete group gauge as the amount of the gathering sizes. They asserted that their methodology was adaptable to concealed group volumes, while it required a little preparing informational index. ***Zhang et al. [3]*** introduced a framework that utilized a vertically mounted Kinect. They abused profundity data as a methods for eliminating the impact of the appearance variety. They saw that since the head is in every case nearer to the Kinect sensor than different pieces of the body, individuals tallying task equivalents to locate the reasonable nearby least locales. Accordingly, they built up a solo water filling strategy, ready to discover these locales while being powerful and scale-invariant. ***Zhao et al. [4]*** utilized an ordinary face location calculation, upgraded by profundity data procured by a Kinect sensor. At that point, they followed individuals and checked directions. Among their perceptions we ought to stress the affectability of face recognition to changing lighting conditions. ***Brostow et al[5]*** followed straightforward picture includes and assembled them into bunches which addressed autonomously moving elements with a probabilistic methodology. ***Rabaud et al [6]*** proposed an exceptionally parallelized adaptation of the notable KLT tracker. A given video was at first handled into a bunch of highlight directions. at that point, a strategy for spatially and transiently molding was applied on the last mentioned. This portrayal was at last taken care of to an ordinary article descriptor. ***Celik et al. [8]*** explored a few methodologies for point of view contortion adjustment and proposed a strategy that depended on forefront object extraction, a viewpoint rectification and a certainty rate that guided a weighted middle channel to refine the tallies. ***Denman et al. [9]*** proposed a scene invariant group tallying calculation, whose objective was to work on different aligned cameras. Highlights between perspectives were standardized and locales of covering were redressed. They additionally explored a few highlights, for example, object size, shape, edges and keypoints and a few relapse models like neural organizations, K-closest neighbors and so forth They accomplished best outcomes by joining every accessible component. ***Chan et al. [10],*** expecting to ensure the security of guineas pigs, proposed a two-venture calculation, where the group was fragmented into parts of homogeneous movement, utilizing the combination of dynamic surfaces movement model. At that point a bunch of straightforward all encompassing highlights was extricated from each fragmented district. The correspondences among highlights and the quantity of individuals per portion were gotten the hang of utilizing Gaussian Process relapse. Thusly, they didn't utilize neither article acknowledgment nor following. At last he proposed an ongoing tallying calculation. They

utilized an observation camera with ordinary mounting. Their procedure joined component coordinating and point line distance approaches at the item and the identification line. They characterized a recognition line and checked individuals that enter or exit

III EXISTING SYSTEM

The current framework works best with shading (RGB) pictures containing thick group for example in excess of 500 heads in a picture. The current framework includes four significant segments. 1. The primary part is a CNN-based head indicator that gives a scanty area of heads and their sizes in the pictures. 2. The subsequent segment is a component classifier the picture is initial separated into equivalent size rectangular patches, which are arranged as group or not group by a Support Vector Machine SVM classifier on speeded up hearty highlights SURF highlights. 3. The third part is a relapse module that assesses the head mean each group fix dependent on its spatial facilitates and assessed head sizes. While notcrowd fixes clearly have zero checks, it is conceivable that the head finder may neglect to recognize any head in a portion of the group patches. 4. This is settled by the fourth segment in which the means these group patches are assessed by the spatially reliant weighted normal of the tallies from the adjoining eight patches. The last advance is to whole all the individual fix appraisals to get a complete mean the whole picture. The current framework doesn't accept that the group fills the whole picture. Since we follow a fix approach, a portion of the patches may not contain any group. It is imperative to distinguish such fixes to dodge over assessment. To address this, we present a parallel group/not-swarm classifier. The current framework just delivers inadequate discoveries, which prompts numerous patches having no heads identified in them at all while SVM may arrange them as group fix. In the event that there are no heads identified in a fix by CNN, the assessed head size would be zero.

IV PROPOSED SYSTEM

The proposed framework sets the location edge and changing the camera. The discovery results underneath the edge, which is generally set from 0.2 to 0.4, won't be tallied. For straightforwardness, we utilize the default estimation of 0.2 in this work. In the genuine scene, the camera ought to be changed in accordance with the fitting tallness and point.

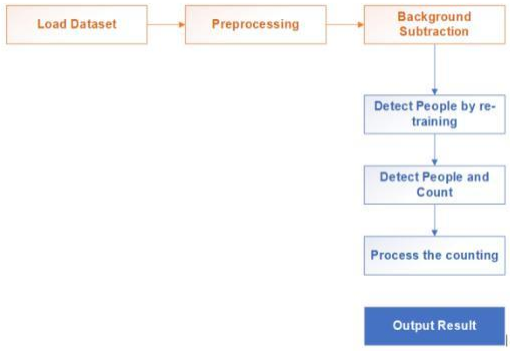


FIG 1 OVERVIEW OF THE PROPOSED SYSTEM

The proposed framework distinguishes individuals through re-preparing a convolutional neural organization. YOLO isolates the picture into a 7*7 network and for every framework cell predicts two jumping boxes just as the certainty esteem for those cases. We accept that this division isn't adequate and we point that our calculation will be more effective in distinguishing individuals to accomplish higher tallying precision. The proposed framework is a powerful foundation deduction module is first considered to portion moving items from each caught video outline. To defeat light varieties, a powerful edge esteem related with recognizing areas of interest from the separated picture is iteratively determined by the conveyances of foundation and closer view pixels in each casing. Subsequent to acquiring the closer view areas, four states including new, leaving, blended and split are allocated to the recognized moving articles as per their appearances in the current casing. Specifically, targets recognized as conditions of consolidation and split further pass through in reverse following for calming

the impediment impacts by exploring the centroid distances among objects in the past edge. At last, focuses in four states are labeled to yield the aftereffects of individuals following and tallying.

V MODULES DESCRIPTION

5.1 PREPROCESSING

Numerical activities: By thinking about the picture number juggling tasks, use of one of the standard numerical or intelligent tasks to at least two pictures is the choice. The administrators are applied in a stepby-step way. That is, the estimation of the yield pixel relies just upon the estimation of the info pixel. Consequently, the size of the picture should be the equivalent. The significant bit of leeway of utilizing numerical activities is that, it is exceptionally quick and easy to execute. The same numerical activities, legitimate tasks are oftentimes used to consolidate at least two double pictures. In the situation of advanced pictures, the consistent administrator is typically applied in somewhat insightful way Convert shading pictures to grayscale to lessen calculation intricacy: in specific issues you'll see it valuable to lose pointless data from your pictures to diminish space or computational intricacy.

For instance, changing your shaded pictures over to grayscale pictures. This is on the grounds that in numerous items, shading isn't important to perceive and decipher a picture. Grayscale can be adequate for perceiving certain articles. Since shading pictures contain more data than highly contrasting pictures, they can add pointless intricacy and occupy more room in memory (Remember how shading pictures are addressed in three channels, which implies that changing it over to grayscale lessens the quantity of pixels that should be handled). One significant requirement that exists in some AI calculations, like CNN, is the need to resize the pictures in your dataset to a bound together measurement. This infers that our pictures should be preprocessed and scaled to have indistinguishable widths and statures before took care of to the learning calculation.

Another regular pre-handling strategy includes expanding the current dataset with bothered variants of the current pictures. Scaling, revolutions and other relative changes are run of the mill. This is done to amplify your dataset and uncover the neural organization to a wide assortment of varieties of your pictures. This makes it almost certain that your model perceives objects when they show up in any structure and shape.

5.2 EDGE DETECTION

Edges are critical nearby changes of force in an image.Edges commonly happen on the limit between two distinct areas in an image.The clear edge in the picture is the vertical line between the dark paper and the white paper. To our eyes, there is a very abrupt change between the dark pixels and the white pixels. Yet, at a pixel-by-pixel level, is the progress actually that unexpected? On the off chance that we focus in on the edge all the more intently, as in this picture, we can see that the edge between the highly contrasting zones of the picture is anything but an obvious line.

Our edge discovery technique in this workshop is Canny edge recognition, made by John Canny in 1986. This technique utilizes a progression of steps, some joining different sorts of edge recognition. The skimageskimage.feature.canny() work plays out the accompanying advances: A Gaussian haze (that is portrayed by the sigma boundary, see presentation) is applied to eliminate commotion from the picture. (So in the event that we are doing edge recognition through this capacity, we ought not play out our own obscuring step.) Sobel edge identification is performed on both the x and y measurements, to discover the force slopes of the edges in the picture. Sobel edge discovery processes the subsidiary of a bend fitting the angle among light and dull territories in a picture, and

afterward finds the pinnacle of the subordinate, which is deciphered as the area of an edge pixel.

Pixels that would be featured, yet appear to be excessively far from any edge, are eliminated. This is called non-most extreme concealment, and the outcome is edge lines that are more slender than those delivered by different techniques. A twofold limit is applied to decide expected edges. Here incidental pixels brought about by clamor or milder shading variety than wanted are killed. On the off chance that a pixel's inclination esteem – in light of the Sobel differential – is over the high limit esteem, it is viewed as a solid contender for an edge. In the event that the inclination is beneath the low edge esteem, it is killed. In the event that the slope is in the middle, the pixel is viewed as a feeble possibility for an edge pixel. Last identification of edges is performed utilizing hysteresis. Here, frail up-and-comer pixels are inspected, and in the event that they are associated with solid up-and-comer pixels, they are viewed as edge pixels; the excess, non-associated feeble up-and-comers are killed.

5.3 PREDICTION

A natural property of articles on the planet is that they just exist as significant elements over specific scopes of scale. A straightforward model is the idea of a part of a tree, which bodes well just at a scale from, say, a couple of centimeters to probably a couple of meters. It is pointless to examine the tree idea at the nanometer or the kilometer level. At those scales it is more applicable to discuss the atoms that structure the leaves of the tree, or the woodland where the tree develops. Likewise, it is simply important to discuss a cover over a specific scope of coarse scales. At better scales it is more fitting to think about the individual drops, which thusly comprise of water particles, which comprise of iotas, which comprise of protons and electrons and so on .

The scale-invariant component change (SIFT) is an element location calculation in PC vision to identify and portray neighborhood highlights in images.Applications incorporate article acknowledgment, mechanical planning and route, picture sewing, 3D displaying, motion acknowledgment, video following, singular ID of natural life and match moving. Also, start the checking cycle.

VI CONCLUSION

In the proposed work introduced a pressed ongoing individuals tallying approach named YOLO-PC. YOLO-PC improves the first convolutional construction of YOLO, and uses the smoothed out fire layer to supersede the 3 x 3 convolutional layer and the loaded model with less boundaries is procured through planning by more divisions of cells, YOLO-PC achieves extra bouncing boxes and higher acknowledgment sureness. Gotten together with a limit choice method, individuals excluding goes to be progressively proper with higher area, just as checking accuracy. The proposed framework is a YOLO based constant individuals checking approach utilizing limit determination. YOLO-PC beats YOLO as it re-trains YOLO organization, which empowers it to distinguish more boxes and accomplish higher normal certainty esteem. The limit determination in YOLOPC makes the checking more focused on and its outcome precise and quick. Taking everything into account, this strategy is exceptionally compelling and it is additionally ready to perceive immaterial individuals and overlook them in the tallying cycle. YOLO-PC has a wide scope of uses as it can help the advancement of numerous parts of the keen urban communities.

REFERENCES

[1] C.T. Hsieh,H.C. Wang, Y.K. Wu, L.C. Chang and T.K. Kuo. “A Kinectbased people-flow counting system” In Proc. of Int. Symp. on Intelligent Signal Processing and Communications Systems (ISPACS), IEEE, 2012.

[2] D. Ryan, S. Denman, C. Fookes and S. Sridharan. “Crowd counting using multiple

local features". In Proc. of Digital Image Computing: Techniques and Applications (DICTA), IEEE, 2009.

[3] X. Zhang, J. Yan, S. Feng, Z. Lei, D. Yi and S.Z. Li. "Water filling: Unsupervised people counting via vertical kinect sensor". In Int'l Conf. on Advanced Video and Signal-Based Surveillance (AVSS), IEEE, 2012.

[4] G. Zhao, H. Liu, L. Yu, B. Wang and F. Sun. "Depth-AssistedFaceDetectionand Association for People Counting". In Pattern Recognition, Springer, 2012.

[5] G.J. Brostow and R. Cipolla. "Unsupervised bayesian detection of independent motion in crowds". In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, 2006.

[6] V. Rabaud and S. Belongie. "Counting crowded moving objects". In Proc. of Conf. on Computer Vision and Pattern Recognition, IEEE, 2006.

[7] P. KadewTraKuPong and R. Bowden. "An improved adaptive background mixture model for real-time tracking with shadow detection." In Proc. of European Workshop on Advanced Video-Based Surveillance Systems, 2001

[8] H. Celik, A. Hanjalic and E.A. Hendriks. "Towards a robust solution to people counting". In Proc. of Int'l Conf. of Image Processing, IEEE, 2006.

[9] D. Ryan, S. Denman, C. Fookes and S. Sridharan. "Scene invariant multi camera crowd counting". Pattern Recognition Letters vol. 44, pp. 98–112, 2014.

[10] A.B. Chan, Z.-S.J. Liang and N. Vasconcelos. "Privacy preserving crowd monitoring: Counting people without people models or tracking". In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2008.

[11] L. Atzori, A. Iera, and G. Morabito, The internet of things: A survey, Computer networks, vol. 54, no. 15, pp. 2787–2805, 2010.

[12] Mqtt protocol specification. [Online]. Available: http://docs. oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html

[13] K.Y. Yam, W.C. Siu, N.F. Law and C.K. Chan. "Effective bi-directional people flow counting for real time surveillance system". In ICCE Proceedings, vol. 11, pp. 863-864, 2011.

[14] C. Akasiadis, E. Spyrou, G. Pierris, D. Sgouropoulos, G. Siantikos, A. Mavrommatis, C. Vrakopoulos and T. Giannakopoulos "Exploiting Future Internet Technologies: The Smart Room Case" In Proc. of International Conference on PErvasive Technologies Related to Assistive Environments (PETRA), 2015.

[15] C.C. Loy, K. Chen, S. Gong and T. Xiang. "Crowd counting and profiling: Methodology and evaluation". Modeling, Simulation and Visual Analysis of Crowds, pp. 347-382, Springer, 2013.

C.PLAGARISM REPORT

PAPER-converted

by Paper-converted Paper-converted

Submission date: 22-Mar-2021 08:53AM (UTC-0700)
Submission ID: 1539435862
File name: PAPER-converted.pdf (138.61K)
Word count: 3930
Character count: 21374

Deep learning based dense object counting with density map

POTTI SAI YATEESH [1] , PRADEEP S [2] , ASHOK KUMAR K[3]

CSE,

CSE,

: Assessing the quantity of people in a public spot gives valuable data to video-based reconnaissance and checking applications. On account of sideways camera arrangement, tallying is either accomplished by distinguishing people or by measurably setting up relations between estimations of basic picture highlights to the quantity of individuals. Ongoing individuals tallying from video records is a fundamental structure block for some applications in shrewd urban areas. In the current System, a head indicator can be utilized to appraise the spatially fluctuating head size, which is the key component utilized in our mind tallying methodology. We influence the best in class convolutional neural organization for the meager head identification in thick group. After sub-partitioning the picture into rectangular patches, we first utilize a SURF highlight based SVM double classifier to name each fix as group/not-swarm and kill all not-swarm patches. In the current framework, task as a rule experiences numerous issues, similar to the absence of continuous handling of the recorded recordings or the event of mistakes because of superfluous individuals being tallied. The proposed framework defeats the above issues with a novel continuous individuals tallying approach named YOLO-PC (YOLO based People Counting). In the proposed framework, after the extraordinary pre-treatment, versatile division and highlight extraction for the human checking information, the element vector is utilized as the contributions of the prepared YOLO to characterize and measurements of the absolute number individuals.

Keywords: People Counting , YOLO-PC, CNN

1 INTRODUCTION

Dependably assessing the quantity of individuals present in a public spot (for example road, shopping center, tram station) over the long run might be of basic significance for both wellbeing and monetary reasons. For example, a specific number of people in a given setting may mirror an abnormal and possibly risky circumstance. Then again, observing the quantity of individuals in a territory of a shopping center will give important data not exclusively to streamlining the working hours of the shops yet additionally for assessing the appeal of the shopping region. With the approach of savvy cameras and the expanding opportunities for robotized reconnaissance and checking, the mechanization of the human tallying task additionally turns out to be mechanically doable. In this work we modify the current article location framework YOLO by proposing the purported YOLO-PC(People Counting dependent on YOLO). YOLO-PC broadens the first YOLO framework utilizing a profound learning way to deal with accomplish a higher exactness in individuals checking. Contrasted with other existi[7] article location frameworks, for example, , and , YOLO been picked as base technique in our YOLO-PC, as a result of its low calculation overhead and its capacity to identify objects continuously. Regarding Performance continuously, YOLO-

PC retrains a profound convolutional neural organization to distinguish individuals at in excess of 40 fps (outlines each second) with the help of a GPU. As to individuals tallying measure, the picture is partitioned into a 9*9 networks and limits are noted. This prompts more recognized frameworks and a more noteworthy normal certainty esteem. Besides the YOLO-PC picks an alternate limit territory as indicated by the genuine application situation to include individuals in a contrapuntal way which further improves the tallying exactness. YOLO-PC can likewise overlook the immaterial people who might be in announcements or other unimportant territories Experimental outcomes show that YOLO-PC can tally individuals rapidly with a higher exactness at the passageway or ways out of spots like lifts, arenas, shopping centers and so on

The initial step comprises in setting the identification edge and changing the camera. The location results underneath the limit, which is generally set from 0.2 to 0.4, won't be tallied. For straightforwardness, we utilize the default estimation of 0.2 in this work. In the real scene, the camera ought to be acclimated to the proper tallness and point. In the subsequent advance, we distinguish individuals through re-preparing a convolutional neural organization. YOLO partitions the picture into a 7*7 framework and for every network cell predicts two jumping boxes just as the certainty esteem for those crates. We accept that this division isn't adequate and we point that our calculation will be more effective in distinguishing individuals to accomplish higher tallying exactness. At the end of the day, YOLO-PC works better with more recognized boxes and higher certainty esteems. To this end, YOLO-PC utilizes 9*9 lattice and 3 bouncing boxes. We set up three arrangements of examinations of 4 minutes video each utilizing various edges (i.e., 0.2, 0.3 and 0.4). The acquired outcomes are promising since YOLO-PC distinguishes more boxes and accomplishes higher certainty esteems for those containers contrasted with YOLO. All the more explicitly, YOLO-PC distinguishes in excess of 10% of the containers, the certainty normal worth is in excess of 50% higher when the limit is 0.2. The third step comprises in recognizing and checking individuals dependent on a suitable limit determination. YOLO-PC chooses at least one lattice cells as the territory limit from 243(9*9*3) cells and picks an alternate limit as per the real circumstance. In the event that individuals turn left by some place, the limit of the left region of the video ought to be chosen, the estimation of the limit is around 113 and individuals through that limit will be tallied. Essentially, if individuals turn directly by some place, the estimation of the limit is around 129.

In the event that individuals go straight by some place, the estimation of the limit is around 121. YOLO-PC can be more precise in recognizing the progression of individuals as some garbage obstruction, like individuals in the boards and disconnected backs, can be overlooked as a result of the limit choice. The checking data of the past advance is currently handled in the fourth step. In the chose limit zone, the cases number collects and continually refreshes, we allude to this number as S. The estimation of S at a second t in a video addresses the quantity of identified individuals at that point, which is exact. The estimation of S in a timeframe addresses the complete number of identified individuals. Since it requires some investment for individuals to move in the limit region, the estimation of S is rehashed, in other words, a similar individual has been distinguished commonly. As per the examinations, each individual has been distinguished around multiple times when going through the chose limit zone, so the anticipated number is S/18 at the default edge. In the fifth and last advance, we yield the constant individuals tallying data. YOLO-PC can straightforwardly show the realtime individuals including data in the video pictures, including the current number, FPS, certainty esteem and so forth YOLO-PC can likewise save continuous data and proceed to refresh,

and afterward yield them through certain interfaces.

handled examination endeavors, utilizing a few heterogeneous methodologies. In any case, an investigation in important writing uncovers that each approach is on a basic level persuaded by the uncommon states of the current issue. Numerous strategies utilize face/individual discovery calculations, while others join mass/object location and following plans. The issue is normally alluded to as "individuals tallying" or "swarm checking" when it is applied to huge, open zones. Regular applications center chiefly around reconnaissance or traffic. The intrigued peruser may discover an investigation zeroing in on checking. To assess person on foot stream checking, ***Hsieh et al. [1]*** introduced a framework that utilizes Kinect. Their methodology depended on morphological handling and extraction of associated parts to extricate districts of interest. They guaranteed that their framework delivered continuous outcomes, having wonderful precision. ***Ryan et al. [2]*** proposed a methodology that pre-owned neighborhood includes and worked on independent closer view mass sections. Thusly, they got a complete group gauge amount gathering asserted methodology adaptable concealed group little preparing informational index. ***[3]*** introduced framework utilized abused profundity data methods for eliminating impact variety. sa in every case nearer different pieces , individuals tallying equivalents locate reasonable nearby least locales. Accordingly, they built up a solo water filling strategy, ready to discover these locales powerful . ***[4]*** utilized an ordinary face location calculation, upgraded by profundity data procured by a Kinect sensor. At that point, they followed individuals and checked directions. Among their perceptions we ought to stress the affectability of face recognition to changing lighting conditions. ***Brostow et al[5]*** followed straightforward picture includes and assembled them into bunches which addressed autonomously moving elements with a probabilistic methodology. ***Rabaud et al [6]*** proposed an exceptionally parallelized adaptation of the notable at first handled bunch highlight directions. at that point, a strategy for spatially and transiently molding was applied on the last mentioned. This portrayal was at last taken care of to an ordinary article descriptor. ***Celik et al. [8]*** explored a few methodologies for point of view contortion adjustment strategy depended forefront viewpoint rectification certainty guided middle channel tallies. ***Denman . [9]*** group tallying calculation, objective work different aligned . Highlights perspectives standardized locales covering redressed. additionally explored a few highlights, for example, a few relapse like organizations, closest and so forth They accomplished best outcomes by joining every accessible component. ***Chan et al. [10],*** pecting to ensure the security of guineas pigs, -venture calculation, group fragmented parts movement, utilizing combination surfaces movement . At that point a bunch of straightforward all encompassing highlights was extricated from each fragmented district. The correspondences among highlights and the quantity of individuals per portion were gotten the hang of utilizing Gaussian Process relapse. Thusly, they didn't utilize neither article acknowledgment nor following. At last he proposed an ongoing tallying calculation. They

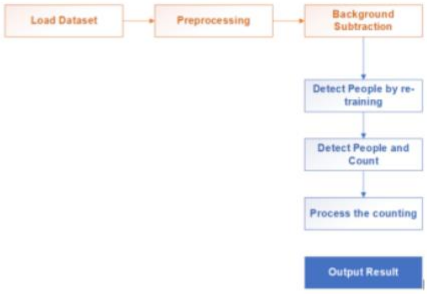
utilized an observation camera with ordinary mounting. Their procedure joined component coordinating item identification characterized recognition checked individuals

EXISTING SYSTEM

The current framework works best with shading (RGB) pictures containing thick group for example in excess of 500 heads in a picture. The current framework includes four significant segments. 1. The primary part is a CNN-based head indicator that gives a scanty area of heads and their sizes in the pictures. 2. The subsequent segment is a component classifier the picture is initial separated into equivalent size rectangular patches, which are arranged as group or not group by a Support Vector Machine SVM classifier on speeded up hearty highlights SURF highlights. 3. The third part is a relapse module that assesses the head mean each group fix dependent on its spatial facilitates and assessed head sizes. While notcrowd fixes clearly have zero checks, it is conceivable that the head finder may neglect to recognize any head in a portion of the group patches. 4. This is settled by the fourth segment in which the means these group patches are assessed by the spatially reliant weighted normal of the tallies from the adjoining eight patches. The last advance is to whole all the individual fix appraisals to get a complete mean the whole picture. The current framework doesn't accept that the group fills the whole picture. Since we follow a fix approach, a portion of the patches may not contain any group. It is imperative to distinguish such fixes to dodge over assessment. To address this, we present a parallel group/not-swarm classifier. The current framework just delivers inadequate discoveries, which prompts numerous patches having no heads identified in them at all while SVM may arrange them as group fix. In the event that there are no heads identified in a fix by CNN, the assessed head size would be zero.

IV PROPOSED SYSTEM

The proposed framework sets the location edge and changing the camera. The discovery results underneath the edge, which is generally set from 0.2 to 0.4, won't be tallied. For straightforwardness, we utilize the default estimation of 0.2 in this work. In the genuine scene, the camera ought to be changed in accordance with the fitting tallness and point.



framework distinguishes individuals through re-preparing a convolutional neural organization. YOLO isolates the picture into a 7*7 network and for every framework cell predicts two jumping boxes just as the certainty esteem for those cases. We accept that this division isn't adequate and we point that our calculation will be more effective in distinguishing individuals to accomplish higher tallying precision. The proposed framework is a powerful foundation deduction module is first considered to portion moving items from each caught video outline. To defeat light varieties, a powerful edge esteem related with recognizing areas of interest from the separated picture is iteratively determined by the conveyances of foundation and closer view pixels in each casing. Subsequent to acquiring the closer view areas, four states including new, leaving, blended and split are allocated to the recognized moving articles as per their appearances in the current casing. Specifically, targets recognized as conditions of consolidation and split further pass through in reverse following for calming

the impediment impacts by exploring the centroid distances among objects in the past edge. At last, focuses in four states are labeled to yield the aftereffects of individuals following and tallying.

V MODULES DESCRIPTION

5.1 PREPROCESSING

Numerical activities: By thinking about the picture number juggling tasks, use of one of the standard numerical or intelligent tasks to at least two pictures is the choice. The administrators are applied in a stepby-step way. That is, the estimation of the yield pixel relies just upon the estimation of the info pixel. Consequently, the size of the picture should be the equivalent. The significant bit of leeway of utilizing numerical activities is that, it is exceptionally quick and easy to execute. The same numerical activities, legitimate tasks are oftentimes used to consolidate at least two double pictures. In the situation of advanced pictures, the consistent administrator is typically applied in somewhat insightful way Convert shading pictures to grayscale to lessen calculation intricacy: in specific issues you'll see it valuable to lose pointless data from your pictures to diminish space or computational intricacy.

For instance, changing your shaded pictures over to grayscale pictures. This is on the grounds that in numerous items, shading isn't important to perceive and decipher a picture. Grayscale can be adequate for perceiving certain articles. Since shading pictures contain more data than highly contrasting pictures, they can add pointless intricacy and occupy more room in memory (Remember how shading pictures are addressed in three channels, which implies that changing it over to grayscale lessens the quantity of pixels that should be handled). One significant requirement that exists in some AI calculations, like CNN, is the need to resize the pictures in your dataset to a bound together measurement. This infers that our pictures should be preprocessed and scaled to have indistinguishable widths and statures before took care of to the learning calculation.

Another regular pre-handling strategy includes expanding the current dataset with bothered variants of the current pictures. Scaling, revolutions and other relative changes are run of the mill. This is done to amplify your dataset and uncover the neural organization to a wide assortment of varieties of your pictures. This makes it almost certain that your model perceives objects when they show up in any structure and shape.

5.2 EDGE DETECTION

Edges are critical nearby changes of force in an image.Edges commonly happen on the limit between two distinct areas in an image.The clear edge in the picture is the vertical line between the dark paper and the white paper. To our eyes, there is a very abrupt change dark white . Yet, at -by-pixel level, is the progress actually that unexpected? On the off chance that we focus in on the edge all the more intently, as in this picture, we can see that the edge between the highly contrasting zones of the picture is anything but an obvious line.

Our edge discovery technique in this workshop is Canny edge recognition, made by John Canny in 1986. This technique joining sorts recognition. skimageskimage. plays out the accompanying advances: haze (portrayed boundary, eliminate commotion . (So in the event that we are doing edge recognition through this capacity, we ought not play out our own obscuring step.) identification discovery processes subsidiary territories

subordinate, , yet eliminated. -most extreme delivered techniques. twofold expected incidental clamor killed. On off chance light contender . In event inclination beneath . In event slope possibility . Last identification edges is performed utilizing hysteresis. Here, frail up-and-comer pixels are inspected, and in the event that they are associated with solid up-and-comer pixels, they are viewed as edge pixels; the excess, non-associated feeble up-and-comers are killed.

5.3 PREDICTION

A natural property of articles on the planet is that they just exist as significant elements over specific scopes of scale. A straightforward model is the idea of a part of a tree, which bodes well just at a scale from, say, a couple of centimeters to probably a couple of meters. It is pointless to examine the tree idea at the nanometer or the kilometer level. At those scales it is more applicable to discuss the atoms that structure the leaves of the tree, or the woodland where the tree develops. Likewise, it is simply important to discuss a cover over a specific scope of coarse scales. At better scales it is more fitting to think about the individual drops, which thusly comprise of water particles, which comprise of iotas, which comprise of protons and electrons and so

an element location identify portray images.Applications incorporate article acknowledgment, mechanical planning and displaying, , video , singular . Also, start checking cycle.

VI CONCLUSION

In the proposed work introduced a pressed ongoing individuals tallying approach named YOLO-PC. YOLO-PC improves the first convolutional construction of YOLO, and uses the smoothed out fire layer to supersede the 3 x 3 convolutional layer and the loaded model with less boundaries is procured through planning by more divisions of cells, YOLO-PC achieves extra bouncing boxes and higher acknowledgment sureness. Gotten together with a limit choice method, individuals excluding goes to be progressively proper with higher area, just as checking accuracy. The proposed framework is a YOLO based constant individuals checking approach utilizing limit determination. YOLO-PC beats YOLO as it re-trains YOLO organization, which empowers it to distinguish more boxes and accomplish higher normal certainty esteem. The limit determination in YOLOPC makes the checking more focused on and its outcome precise and quick. Taking everything into account, this strategy is exceptionally compelling and it is additionally ready to perceive immaterial individuals and overlook them in the tallying cycle. YOLO-PC has a wide scope of uses as it can help the advancement of numerous parts of the keen urban communities.

REFERENCES

[1] C.T. Hsieh,H.C. Wang, Y.K. Wu, L.C. Chang and T.K. Kuo. “A Kinectbased people-flow counting system” In Proc. of Int. Symp. on Intelligent Signal Processing and Communications Systems (ISPACS), IEEE, 2012.

[2] D. Ryan, S. Denman, C. Fookes and S. Sridharan. “Crowd counting using multiple

local features". In Proc. of Digital Image Computing: Techniques and Applications (DICTA), IEEE, 2009.

[3] X. Zhang, J. Yan, S. Feng, Z. Lei, D. Yi and S.Z. Li. "Water filling: Unsupervised people counting via vertical kinect sensor". In Int'l Conf. on Advanced Video and Signal-Based Surveillance (AVSS), IEEE, 2012.

[4] G. Zhao, H. Liu, L. Yu, B. Wang and F. Sun. "Depth-Assisted Face Detection and Association for People Counting". In Pattern Recognition, Springer, 2012.

[5] G.J. Brostow and R. Cipolla. "Unsupervised bayesian detection of independent motion in crowds". In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, 2006.

[6] V. Rabaud and S. Belongie. "Counting crowded moving objects". In Proc. of Conf. on Computer Vision and Pattern Recognition, IEEE, 2006.

[7] P. KadewTraKuPong and R. Bowden. "An improved adaptive background mixture model for real-time tracking with shadow detection." In Proc. of European Workshop on Advanced Video-Based Surveillance Systems, 2001

[8] H. Celik, A. Hanjalic and E.A. Hendriks. "Towards a robust solution to people counting". In Proc. of Int'l Conf. of Image Processing, IEEE, 2006.

[9] D. Ryan, S. Denman, C. Fookes and S. Sridharan. "Scene invariant multi camera crowd counting". Pattern Recognition Letters vol. 44, pp. 98–112, 2014.

[10] A.B. Chan, Z.-S.J. Liang and N. Vasconcelos. "Privacy preserving crowd monitoring: Counting people without people models or tracking". In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2008.

[11] L. Atzori, A. Iera, and G. Morabito, The internet of things: A survey, Computer networks, vol. 54, no. 15, pp. 2787–2805, 2010.

[12] Mqtt protocol specification. [Online]. Available: http://docs. oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html

[13] K.Y. Yam, W.C. Siu, N.F. Law and C.K. Chan. "Effective bi-directional people flow counting for real time surveillance system". In ICCE Proceedings, vol. 11, pp. 863-864, 2011.

[14] C. Akasiadis, E. Spyrou, G. Pierris, D. Sgouropoulos, G. Siantikos, A. Mavrommatis, C. Vrakopoulos and T. Giannakopoulos "Exploiting Future Internet Technologies: The Smart Room Case" In Proc. of International Conference on PErvasive Technologies Related to Assistive Environments (PETRA), 2015.

[15] C.C. Loy, K. Chen, S. Gong and T. Xiang. "Crowd counting and profiling: Methodology and evaluation". Modeling, Simulation and Visual Analysis of Crowds, pp. 347-382, Springer, 2013.

PAPER-converted

ORIGINALITY REPORT

12%	2%	5%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Harrisburg University of Science and Technology Student Paper	5%
2	Dimitris Sgouropoulos, Evaggelos Spyrou, Giorgos Siantikos, Theodoros Giannakopoulos. "Counting and tracking people in a smart room: An IoT approach", 2015 10th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP), 2015 Publication	4%
3	Submitted to Humble Independent School District Student Paper	1%
4	Submitted to Panjab University Student Paper	<1%
5	Submitted to GLA University Student Paper	<1%
6	www.computer.org Internet Source	<1%

7	Ildar Rakhmatulin. "Detect caterpillar, grasshopper, aphid and simulation program for neutralizing them by laser", Research Square, 2021 Publication	<1%
8	P. Saikumar, P. Bharadwaja, J. Jabez. "Android and Bluetooth Low Energy Device Based Safety System", 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 2019 Publication	<1%
9	f4k.dieei.unict.it Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		