Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 156 (2019) 327–337

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Young Scientist Conference on Computational Science

People Detection and Finding Attractive Areas by the use of Movement Detection Analysis and Deep Learning Approach

Ebrahim Najafi Kajabad [a,*], Sergey V. Ivanov [a]

ITMO University, Saint-Petersburg, Russia
naser.najafi1988@gmail.com,
sergei.v.ivanov@gmail.com.

Abstract

This paper presents a technical approach related to the video computer analysis, to detect people and control the behaviour of people. Control the behaviour of people in public place can be a benefit for understanding the share of overall traffic your area is attracting. Find out what, encourage customers to buy products are most crucial for big companies to increase the sales rate and to improve the quality of customer service. We use surveillance cameras, which located in the museum. We offered two methods, the first method for detecting people in a closed space and second method finding density areas which people more spend time to visit. The YOLO model makes predictions with a single network evaluation. Systems like R-CNN and Faster R-CNN, on the other hand, make multiple assessments for a single image, making YOLO extremely fast, running in real-time with a capable GPU. For detect people used YOLOv3 algorithm which is published by [18] and shows that it has high accuracy to identify people, also we compared the proposed method with other detectors, HOG, SSD and YOLO-tiny which shows the proposed algorithm has better performance in this point. And for finding density areas, We utilized a background subtraction with Gaussian Mixture algorithms and heatmap colour technique to analysis each frame and figure out, where are the density areas which shows people like to spend more time to visit. The experimental results have shown that the accuracy and the performance of both algorithms are quite good.

Keywords: Visual Surveillance, People Detection (YOLO), Background Subtraction, Density Maps (heatmap color)

1. Introduction

Nowadays, intelligent video surveillance systems play a crucial role in controlling and analyzing the behaviour of

[*] Corresponding author. Tel.: +7-951-662-8986
E-mail address: naser.najafi1988@gmail.com

1877-0509 © 2019 The Authors. Published by Elsevier Ltd.

10.1016/j.procs.2019.08.209

human in public areas. Since the last decade, people detection has become one of the most popular topics in different areas. Many researchers have focused on this area, and various approaches have been proposed to detect people. These approaches can divide into two groups according to the view representation type: namely deep learning algorithms, and computer vision techniques. Identifying and tracking people are the fundamental requirements when analyzing customer behaviour and translate to much higher business value not only when it comes to security or authentication, but also consists of an analysis of store traffic, hot zones, and decision making, among other things [1]. The customers might change their shopping behaviours and expand new needs for products when life changes happen. However, it is hard for a retailer to conduct targeted direct marketing based on life changes. Analysis of the psychology of customer behaviour or visitors in, for example, a store or a museum, can be more beneficial to increasing the sales rate and improving the quality of customer service. Unfortunately, however, customer detection in crowded spaces, whether public or private, is the most challenging task to realize, mostly because of the people occlusions that occur when many people are there, making it hard for a surveillance system to detect people. Many algorithms can be used for people detection, but most of them, such as Faster R-CNN or R-CNN, suffer from low speed.

Furthermore, finding the areas that are more attractive to customers or visitors - hot zones - are significant to companies due to them being able to signify what type of products or areas are more appealing to customers. There are various methods of finding the hot zones in a place such as installing cameras in storefronts, objects for counting people or using other kinds of sensors to detect people in specific situations. These kind of solutions, however, are very costly and could also be challenging to control and maintenance. In this paper, we propose using convolutional neural network algorithms to detect people in a closed public space (a museum), which enables people detection; and, as a way to highlight hot zones, moving object and colourmap for collecting statistical data about customer behaviour.

This research has two objectives: the first is people detection based on a deep learning algorithm, and the second is distinguishing and finding the areas in which people spend more time. We suggested the YOLOv3 algorithm for people detection. A background subtraction algorithm and heatmap colour proposed for determining the interest areas.

2. Related Works

In this section, we discuss published research related to people detection and high-density areas. Singh et al. [2], developed an algorithm for real-time object detection. They used a background subtraction algorithm, optical flow method, and the median filter to remove noise during tracking and detecting the object. This method can be successful in case of a static camera. In [3], Faster- R-CNN was proposed for identifying and tracking multiple people. As for object association, two methods utilized. The first method was a simple Euclidean distance, while the second was a Siamese neural network. Although it works faster and with higher performance and accuracy than the original R-CNN version, Faster- R-CNN still depends on an external proposal algorithm. [4] proposed R-CNN, which improves mean average precision by more than 30%, for object detection. It works based on a combination of feature region proposals with Convolutional Neural network (CNN). After receiving an input image, the algorithm extracts approximately 2000 bottom-up region proposals and then computes features for each plan by using CNN, after which a linear support vector machine used for classification. Region-based convolutional network method obtains extraordinary object detection accuracy by using deep ConvNet to classify object proposals. It, however, has notable drawbacks such as long training period, training in multistage pipelines, and, finally, the speed of object detection is deficient. In [5], described an object detection method by using a preprocessing method, such as a background subtraction algorithm, followed by the genetic algorithm for the segmentation plan.

In [6] described a new efficient algorithm for Movement Estimation and object tracking in video scenes using Optical flow and Gabor feature based contour model. To obtain foreground pixel, Gaussian Mixture Model algorithm based on background subtraction used. Also, Adaboost classifier used for classified extracted features; the problem is that this algorithm works just based on the moving object. [7] proposed a deep learning (CNN) algorithm for human detection, to achieve higher accuracy and less computation for detecting, they used a combination of deep learning and classical machine learning methods. This algorithm, unfortunately, has the same problem low speed. In [8], a YOLOv2, which is state of the art on standard detection tasks, was proposed. A database such as Pascal Voc or COCO, used for training which able to detect over 9000 object categories. It uses a hierarchical view of object classification that allows the combination of distinct datasets and focuses mainly on improving recall and localization while maintaining classification accuracy. Although this algorithm achieves such high object detection by performing joint training for both object detection and classification, the accuracy of the 156 class version of COCO is weak - just 16%

mean Average Precision. [9] described incorporating local crowd density to human detection by motion heat map method, local feature tracking and non-max suppression for removing overlap noise and to dealing with false positive detection dynamically, with learning correction filters exploiting the aspect ratio and perceived height of discoveries proposed. [10] described a method for object detection using a single deep neural network called single shot detector. A vital point of this algorithm is the use of multi-scale convolutional bounding box outputs attached to multiple feature maps at the top of the network. This algorithm is faster than the current real-time YOLO. However, SSD uses upper layers for detection, and therefore, it cannot detect small objects. However, own proposed algorithm which uses the Feature Pyramid Network (FPN) for detecting small objects. FPN is a feature extractor designed with feature pyramid concept to improve accuracy and speed. While the performance and the efficiency of R-CNN family are high, these methods suffer from low rate, obtaining only 5 FPS on a GPU. YOLO algorithms combat this problem very well, but they still lack some of the essential elements that now stapled in most of the state-of-the-art algorithms. They usually have no residual blocks, no skip connections and no upsampling. However, YOLOv3 includes all of these.

In [11], the authors explained a solution to detecting a pedestrian with low-resolution mobile cameras by using background subtraction to extract the foreground, and the sparse-sensing strategy suggested to extract foreground silhouettes and classify them in real-time. Additionally, a cascade of short binary strings is presented to model the appearance of pedestrians. In [12], the researchers focused on static crowds - groups of people who stayed in the same place for a long time. They designated the mean of one class as support vector machine to classified patches as the potential crowd, which patch level extracted by texture features. SVM was used to classify the pieces from the crowded class from classes belonging to the background. As a means of filtering out misclassified patches and removing noise, a motion feature accepted. [13] described a method to people density maps which are related to the video computer analysis based on people movement in sales rooms. A background subtraction algorithm, optical flow methods, and heatmap colours proposed to track these features.

3. YOLO Algorithm

More recent approaches, such as R-CNN, use region proposal methods first to generate potential bounding boxes in an image and then run the classifier on these proposed boxes. After classification, post-processing is used to refine the bounding boxes, remove overlap detection and re-score the boxes based on another object in the scene. These complex pipelines, however, are not fast and are computationally expensive due to each component requiring separate training [14]. However, You Only Look Once (YOLO) at an image is used to predicting what objects are present and where they located. Therefore, this algorithm could be faster than Fast R-CNN and R-CNN in which only 5 FPS on a GPU can obtain [22][17]. It reforms object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. It is extraordinarily fast because it detects each frame as a regression problem and does not need a complex pipeline. Indeed, we need to run our neural network on a new image at test time to predict detections. YOLO and SSD help increase the speed of deep learning based object detectors using a one stage detector strategy. This algorithm treats object detection as a regression problem, taking a given input image while simultaneously learning bounding box coordinates and corresponding class label probabilities. Although single stage detectors tend to be less accurate than two-stage detectors, they are significantly faster because of that the first stage generates proposals, and the second stage is a classifier [15].

Fig. 1. Displays the pipeline structure of YOLO that makes use of whole topmost feature map to predict both confidences for multiple categories and bounding boxes. Yolo divides the input image into an $S * S$ grid and each of them is responsible for predicting one object in that grid cell without considering the number of bounding boxes. Each grid cell predicts B boundary boxes and each box has one box confidence score. Officially, confidence scores determined as $Probability(Object) * IOU_{pred}^{truth}$, which indicates how likely objects exist ($Probability(Object) \geq 0$) and shows the confidence of probability ($The\ IOU_{pred}^{truth}$). At the same time, regardless of the number of boxes, C conditional class probability ($Probability(Class_i \mid Object)$) should also be predicted in each grid cell. It should be considering that only the contribution of the grid cell containing an object calculated. The below formula shows class-specific confidence scores for each box.

$$\begin{aligned} Pr\ (Object)\ * IOU_{pred}^{truth} * Pr(Class_i \mid Object) \\ = Pr(Class_i\ ) * IOU_{pred}^{truth} \end{aligned} \tag{1}$$

Where the existing probability of class-specific object in the box and the fitness between the predicted box and the purpose are both taken into consideration [16], YOLO has several advantages when compared to conventional methods of object detection like R-CNN. Firstly it applies a single CNN for both classification and localization of the object and secondly YOLO can process images very fast, about 40-90 FPS, which is quite fast and allows real-time video processing, in a few milliseconds. It is 1000 times faster than R-CNN and 100 times faster than fast R-CNN [17].

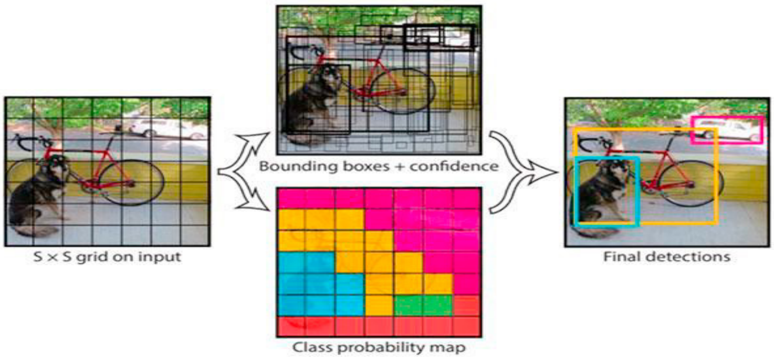

Fig. 1. YOLO Object Detector Structure [15].

The YOLOv3 algorithm utilized logistic regression to predict an objectness score for each bounding box. It should be 1 if the bounding box prior overlaps a ground truth object by more than any other bounding box prior. However, if the bounding box prior is not the best but does overlap a ground truth object by more than some threshold, we ignore the prediction. In this algorithm, binary cross entropy loss used for the class predictions. It could be useful for more complex domains like the Open Image Dataset, due to there being many overlapping labels (i.e. Woman and Person) in the dataset. YOLOv3 predicts boxes on 3 different scales. This algorithm extract features from those scales using a similar concept to feature pyramid networks. The last of this predicts a 3-D tensor encoding bounding box, objectness and class predictions. In the COCO dataset, each scale predicts 3 boxes, so the tensor is N*N*[3*(4+1+80)] for the 4 bounding box offsets, 1 objectness prediction, and 80 class predictions. It still uses k-means clustering to determine bounding box priors. It is a good detector because it is fast and more accurate. YOLOv3 is not as high on COCO, with an average AP between 0.5 and 0.95 IOU metric. But, it is very good at the old detection metric of 0.5 IOU. In terms of COCOs weird average mean AP metric, it is on par with the SSD variants but is 3x faster [18]. YOLOv3 In total uses 9 anchor boxes which are generated by k-means clustering, three for each scale. It then arranges the anchors in descending order of a dimension, assigning the three most prominent anchors for the first scale, the next three for the second scale, and the last three for the third.

4. Background subtraction Algorithm

A background subtraction algorithm is one of the most important algorithms for detecting moving objects from a static camera. It works based on the current frame and the previous frame. In other words, it will identify the object via a comparison between a current frame and the background frame.

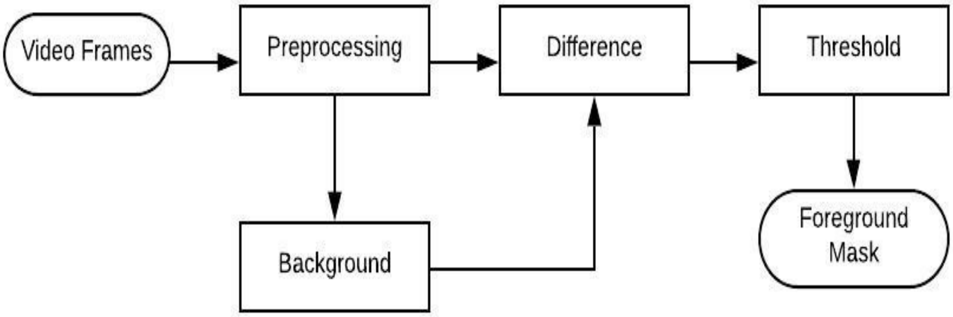

Fig. 2. Diagram of Background Subtraction Algorithm.

Fig.2 shows the simple structure of background subtraction, including four significant steps, namely preprocessing, background modelling, foreground detection, and data validation. Background modelling divided into two groups, which include: (i) non-recursive, and (ii) recursive techniques [19]. The recursive procedure includes frame differencing, median filter, linear predictive filter, Kalman filter and Mixture of Gaussian (MOG). In this model for background estimation, the background will be updated based on each input frame recursively. But the non-recursive technique for background estimation uses the sliding windows method. It stores a buffer of previous video frames and then predicts the background image based on a value of each pixel in the buffer. The equation to explain the way of separating a foreground object from the background can be as follows:

$$|I(x, y, t) - B(x, y, t)| > Th, \tag{2}$$

where *I* is the image at time *t*, and *B* is the background of an image at time *t*. This *Th* threshold to the completely different, to get a mask when subtracted the estimated background from an input frame [20].

4.1 Gaussian Mixture Model(GMM)

GMM is a background estimation technique. This algorithm uses a combination of K Gaussian components to simulate the background image. At time t, the probability of current pixel value Xt can be writ as:

$$P(x_t) = \sum_{i=1}^{k} w_{i,t} \times \eta(X_{t,} \mu_{i,t,}, \Sigma t), \tag{3}$$

where k is the ith Gaussian which usually is an integer value between *3* and *5*; $w_{i,t}$ is an estimate of the weight of the *ith* Gaussian distribution at time *t*; X_t is a current pixel value vector; $\mu_{i,t}$ is the mean value of the *ith* Gaussian distribution at time *t;* and *Σ,t* represents covariance matrix, And the sum of the K weight value is 1 [13].

$$\sum_{i=1}^{k} w_{i,t} = 1 \tag{4}$$

For each pixel, model parameters tuned. The Gaussians ordered by the value of *w/*|*Σ*| and the first distribution of Gaussians selected as a background model.

$$B = argMin_b \{\sum_{i=1}^{b} w_{i,t} > T\}, \tag{5}$$

where *B* is a component responsible for the background and *T* represents the threshold [21].

5. Proposed methods

This section presents two methods for people detection and finding hot zones place in a museum.

5.1. Object detection (YOLOv3)

The following diagram, Fig.3., shows the process of object detection based on the proposed algorithm. In this implementation, we used the pre-training model of YOLOv3 which trained on COCO dataset. It includes 80 classes. In this process, after receiving a sequence of frames from the camera but before implementing the YOLO algorithm, pre-processing needs to be applied to resize the image for importing to the network. For training, we changed the size of the input image to 416*416. After resizing a picture, we used mean subtraction to help combat illumination changes in the input images in our dataset, where the RGB mean has been calculated from a dataset of images and subtracted from the original image. Mean subtraction is used as a technique to aid our Convolutional Neural Network. Before we even begin training our deep neural network, we first compute the average pixel intensity across all images in training set for each of the Red, Green, and Blue channels. The scaling factor is used to scale the input image space into a particular range between 0 and 1. After that, the result of pre-processing passed to YOLO object detection. Our model consists of 75 hidden layers. After processing, each object represented by a bounding box.

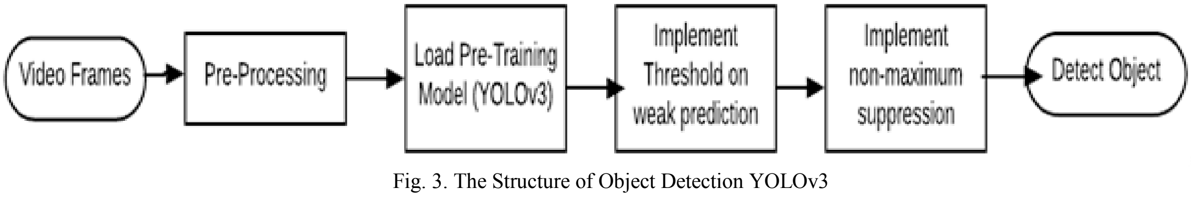
Video Frames
Pre-Processing
Load Pre-Training Model (YOLOv3)
Implement Threshold on weak prediction
Implement non-maximum suppression
Detect Object

Fig. 3. The Structure of Object Detection YOLOv3

When object detection completed, weak predictions are filtered out by using the detected probability, which is higher than the minimum expectation of the Intersection Over Union (IOU) threshold = 0.5. It means that detection with an IOU greater than .5 is a real positive. If we use two anchor boxes, then for each N*N grid cells, we will get two predicted bounding boxes. Low probability predictions need to exclude. Finally, generating final predictions depends on the number of classes. For detecting an object, non-maximum suppression (NMS) must run independently. NMS also ensures that we do not have any redundant or extraneous bounding boxes and keep only the most positive ones. The overlap threshold usually is between 0.3 and 0.5.

5.2. Finding Density Areas

Fig. 4 represents the second part of our proposed algorithm. We first grab the first frame of the video and implement pre-processing, which included, resize the image to 400 * 400 and convert RGB colour to the grey image. In the end, the array of the matrix is converted to zero (black) to have an empty array. We then add this empty array to a binary image to show changes in the motion of an object in a sequence of frames. Fig. 5. shows pre-processing of the first frame. After completing the first frame, the next series of video frames to be pre-processed needs to import. All video frames were resized, converted to grey images, and passed on to the background subtraction algorithm. It utilized to extract foreground and background mask image. A critical part of this algorithm is that it selects the appropriate number of Gaussian distribution for each pixel. We used Threshold = 2000 on the squared Mahalanobis distance between the pixel and the model to decide whether a pixel described well by the background model.

Furthermore, we used to detect shadows as True. It will help us to discover shadows and mark them when people stand in a particular place. Subsequently, the background subtraction result is sent to the next step - applying binary thresholding, which only keeps pixels above Threshold =255. It means that If pixel value is higher than a threshold value, it is assigned one value [maybe white], else it is assigned another value [maybe black] and setting the result to maximum value = 0.6, which represents the value to be given if the pixel value is more than (sometimes less than) the threshold value. If you want the motion to be picked up more, increase the amount of maximum value. The human understanding does not make for noticing subtle changes in grayscale images. However, human eyes are more sensitive to observing differences between colours, for this reason, you might need to recolour your grayscale photos to get a better perceptive about them.

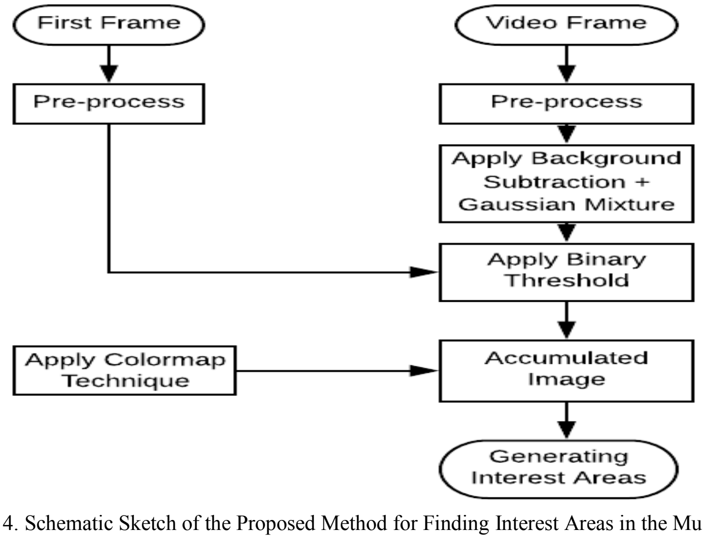

Fig. 4. Schematic Sketch of the Proposed Method for Finding Interest Areas in the Museum.

For providing a visual representation of data need to the visualization process. Mapping of numbers to colours is one of the most substantial and critical aspects of this process. This technique allows us to false colour an image or object by different numerical information and to mark regions correlating with various substances with different colours. The select of colour map is essential to let the viewer understand the reverse mapping back to scalar values easily. Generally, a colourmap is a mapping technique from 0-255 values to 256 colours.

In OpenCV, to generate a colour map need to create an 8-bit colour image of size 256*1 to store the 256 colour values. The result of binarization combined with the previously captured first frame. When moving an object during each frame, changes will show on the scene To visualization the Colormap method load after accumulate image to generate interest areas. The depth map initially has colour values at each point from 0 to 255, where 0 shows black colour, and 255 shows white colour. When "repainting" for each volume of grey colour is taken corresponding to the identified line of colours. We use the COLORMAP_JET palette for colouring. When applied, a value of 0 corresponds to colour (dark blue), at 255, colour (red). Fig.6. illustrates the results of combining the first frame and binarization frame and loading colourmap.

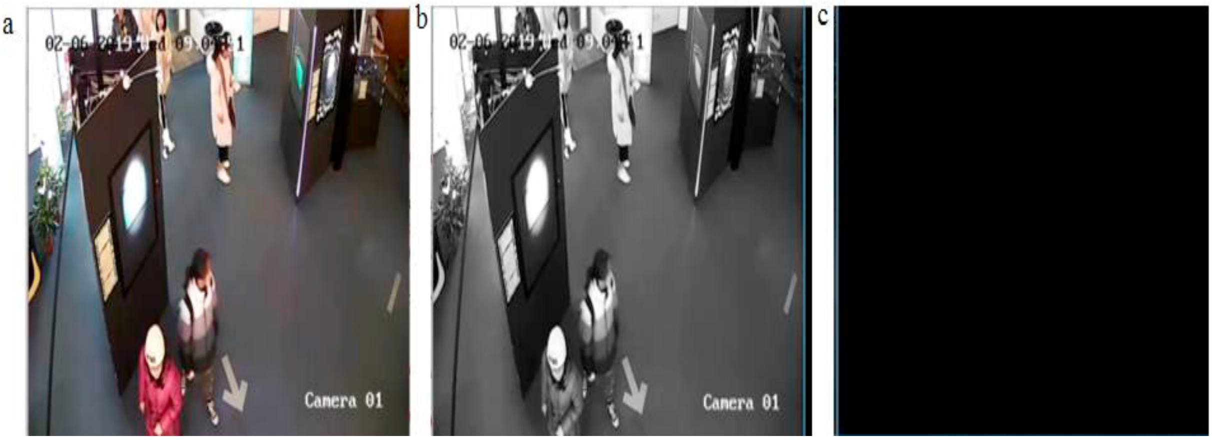

Fig. 5. Represents the Original Image (a), Gray Image (b) and Convert to Zero Value(c).

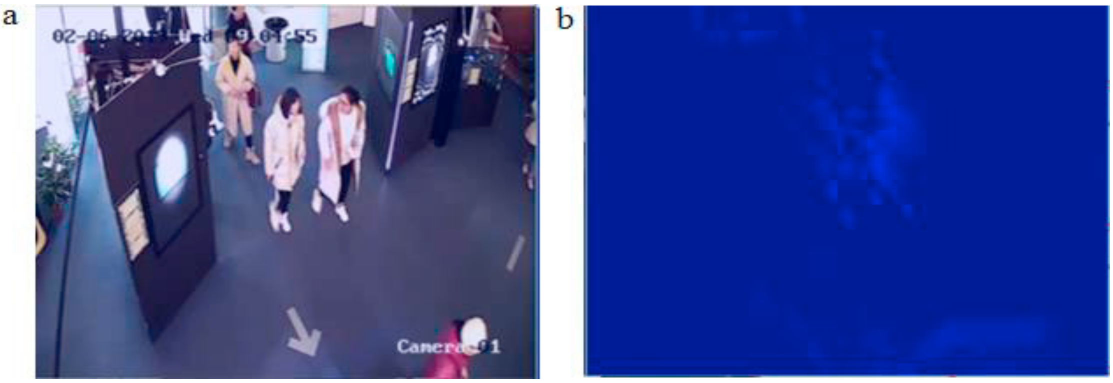

Fig. 6. Input Frame(a), Accumulate image plus Colormap (b).

6. Experimental Results

To implement people detection and to find interest areas, the programming language Python 3.5 with (OpenCV 3.4v) library on an Intel Core i5, 2.7 GHz PC with a 6 GB memory utilized. As a reminder, the structure of implementation explained in continue. In the first part, for object detection, a YOLOv3 algorithm model, which was already trained by COCO dataset that is composed of 300,000 adequately segmented images, was used. Each image has an average of 7 object instances from a total of 80 categories such as humans, cats, dogs, traffic lights, etc. Our model includes 75 hidden convolutional layers and uses a linear activation function for the final layer, and all other layers use leaky rectified linear activation. The total frames in the video were 87077, and because our processing system works based on CPU and had more hidden layers, the speed for processing was low. After receiving a frame from the camera and applying it to preprocess, and loading the algorithm, the preprocessing result fed to the network for detecting an object. Using non-max suppression, 0.3 and only retains the bounding box that has the maximum probability of object detection associated with it. Fig.7 shows the performance of the YOLOv3 algorithm, which was able to detect most of the people, although they are closer to each other. In this investigation, we compared our own proposed algorithm with different types of algorithms such as HOG, SSD, and YOLOv3-Tiny. The YOLOv3-tiny model includes 13 convolutional hidden layers and processing for every single frame was quite fast - less than 1 second[0.75 ms], but it lost people during detection and could not able to detect all the people.

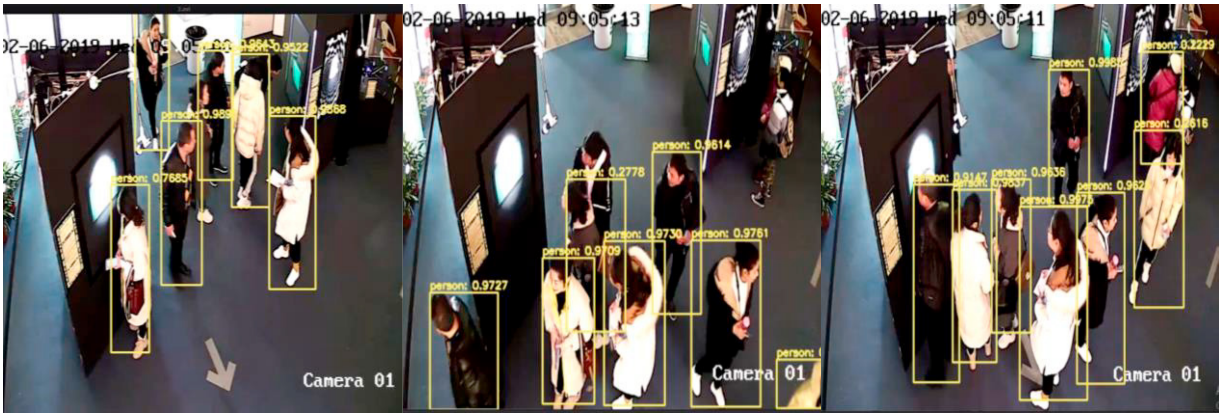

Fig.7. The Results of People Detection Based on YOLOv3 Algorithm.

The SSD-Mobilenet is another CNN pre-training model which is trained by COCO dataset and includes 31 hidden layers. The speed of this model is quite fast, and it can work in real time, this framework processed 6 -7 frames per second based on the CPU of my system. However, the accuracy is lower than YOLOv3, and it misses some people during detection when they are close together. Also, we used the Histogram Oriented Gradient plus Support Vecore Machines to detect people. It used sliding windows technique to extract features from an image. At each window extract, HOG descriptors apply the classifier technique. However, Histogram Oriented Gradient detects people on different scales and also lost people during detection. Fig.8 demonstrates the performance of other algorithms.

The second part represents the results of background subtraction based on a Gaussian Mixture and heatmap colour algorithms which were used to show what types of unique objects or places are more interesting for visitors in the museum. The proposed algorithm works based on a static camera. We tested our algorithm on the video, which was recorded by the museum camera. The algorithm does not focus on the number of people that pass during sequence frames from the scene. It does, however, examine the dissimilarities in people density in various sections of the analyzed areas. While processing the entire video sequence, the high-interest area is generated based on the people who are standing before each unique object.

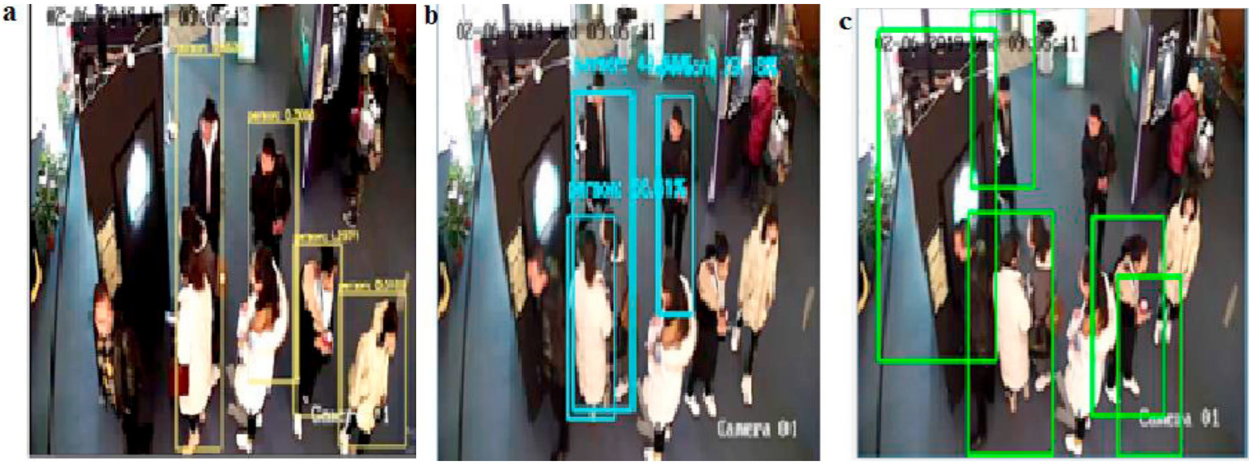

Fig. 8. Detect People by Different Algorithms: (a) Represent results of YOLO-tiny,(b) SSD, and (c) HOG Algorithm.

The results of density maps computed based on different intervals times, 10, 20 and 30 minutes, respectively where the red areas represent the hot zones that people were more likely to visit, by observing changes in the people distribution in the room during the time. As mentioned before, in section 5.2, a threshold of 2000 was used to calculate the distance between the pixel and the model to decide whether the background model well describes a pixel. It also helps us to reduce the noise from the scene when people are merely moving. Fig. 10. shows the result of 2 minutes processing when the threshold value is low, about 80, and the scene is noisy - which is not useful to us.

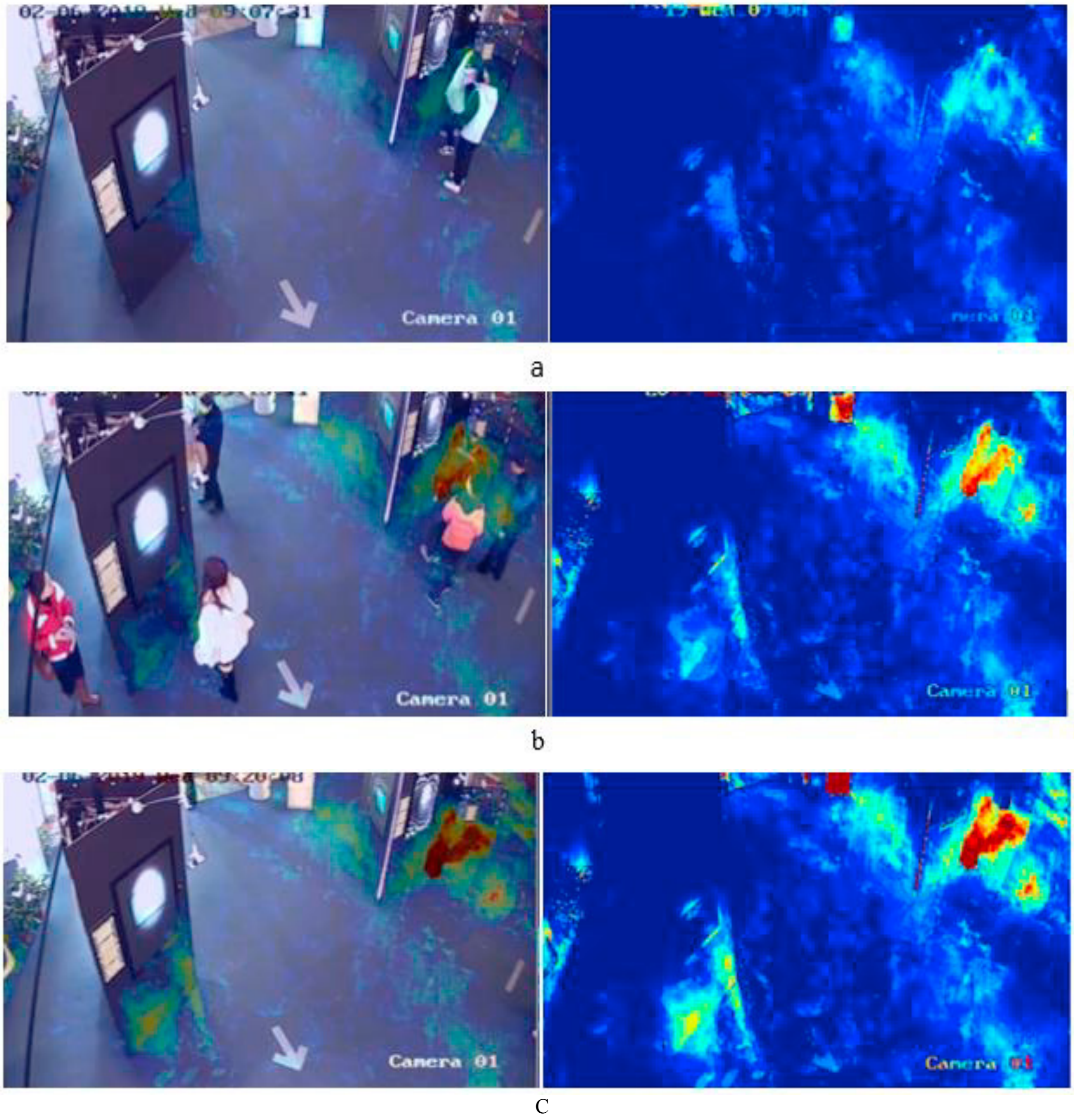

Fig. 9. The Results of Density Color Maps in Various Times. (a) after 10 minutes,(b) after 20 minutes and (c) after 30 minutes.

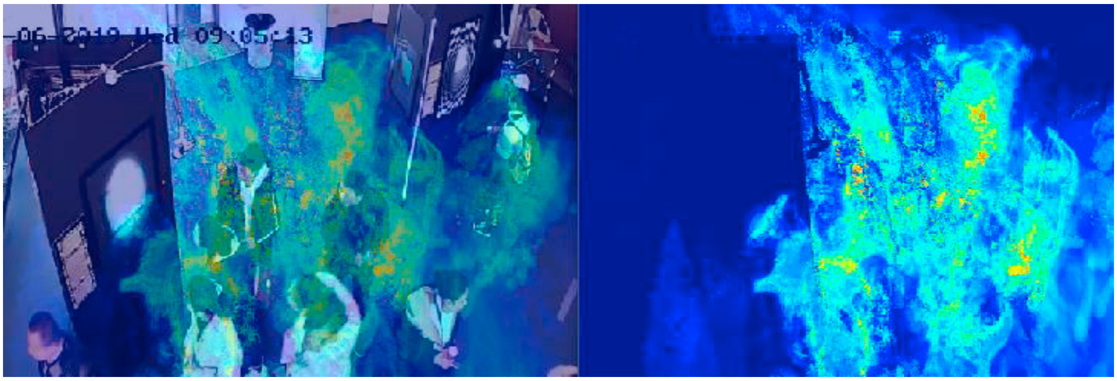

Fig. 10. Represent result with low threshold = 80 with more noise.

7. Discussion and Conclusions

Object detection is a core difficulty in the computer vision system. This study tried to present two algorithms for people detection and finding hot zones in an indoor place. The YOLOv3 algorithm was proposed to detect people in density place. We conclude that YOLOv3 algorithm better detects people in closed space, although it is not as fast as the SSD algorithm, it has more accurate results and is also faster than Faster-R-CNN and YOLOv2. YOLOv3 works well, but it suffers from some restrictions. First of all, If there are two anchor boxes but three objects in the same grid cell, it does not support them correctly, which ultimately leads to missing objects. Another drawback is that If there are two objects in the same grid cell and have the same anchor box shape. It can not handle well.

The background subtraction with a Gaussian Mixture model plus heat map colour was proposed to find density areas in video frames. This method selects the appropriate number of Gaussian distribution for each pixel. It can provide a more beneficial adaptation for varying scenes during illumination changes. In this part, to understand whether the background model well explains a pixel, we used high value for analyzing the distance between the pixel and the model. Experimental results show that the proposed method can effectively detect the density areas by colour map technique.

The proposed methods can also be beneficial to control the behaviour of the customer in the shopping centre, by detecting people in a crowded situation and also to distinguish what type of stores, brands, and products are more interesting for customers. Thus by analysis, the customer's behaviour, companies manager can draw conclusions that can very fast change the functioning of the sale area. In the future, we plan to analyze and distinguish the number of people who visited a specific area and also to detect people when they are together.

Acknowledgments

This work financially supported by the Ministry of Education and Science of the Russian Federation, Agreement #14.575.21.0165 (26/09/2017). Unique Identification RFMEFI57517X0165.

References

[1] S. Yoshinaga, A. Shimada, and R. I. Taniguchi. (2010) “Real-time people counting using blob descriptor,” *Procedia - Soc. Behav. Sci.*, vol. 2, no. 1, pp. 143–152.

[2] P. Singh, B. B. V. L. Deepak, T. Sethi, M. Dev, and P. Murthy. (2015) “Real-Time Object Detection and Tracking Using Color Feature and Motion,”.

[3] D. Chahyati, M. I. Fanany, and A. M. Arymurthy. (2017) “Tracking People by Detection Using CNN Features,” *Procedia Comput. Sci.*, vol. 124, pp. 167–172.

[4] R. Girshick, J. Donahue1, T. Darrell1, J. Malik. (2014) “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Computer Vision and Pattern Recognition”.* arXiv:1311.2524, pp. 2-9.

[5] C. Coniglio, C. Meurie, O. Lézoray, and M. Berbineau. (2017) “People silhouette extraction from people detection bounding boxes in images,” *Pattern Recognit. Lett.*, vol. 93, pp. 182–191.

[6] S. Kanagamalliga and S. Vasuki. (2018) “Optik Contour-based object tracking in video scenes through optical flow and Gabor features,” *Opt. - Int. J. Light Electron Opt.*, vol. 157, pp. 787–797.

[7] A. Brunetti, D. Buongiorno, G. Francesco, and V. Bevilacqua. (2018) “Neurocomputing Computer vision and deep learning techniques for pedestrian detection and tracking : A survey,” *Neurocomputing*, vol. 300, pp. 17–33.

[8] J. Redmon, A.Farhadi. (2017) “YOLO9000: Better, Faster, Stronger,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* DOI**:** 10.1109/CVPR.2017.690.

[9] H. Fradi, V. Eiselein, J. L. Dugelay, I. Keller, and T. Sikora. (2015) “Spatio-temporal crowd density model in human detection and tracking framework,” *Signal Process. Image Commun.*, vol. 31, pp. 100–111.

[10] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg. (2016) “SSD: Single Shot MultiBox Detector” *Computer Vision and Pattern Recognition*, arXiv1512.02325.

[11] A. Alahi, M. Bierlaire, and P. Vandergheynst. (2014) “Robust real-time pedestrians detection in urban environments with low-resolution cameras,” *Transp. Res. Part C Emerg. Technol.*, vol. 39, pp. 113–128.

[12] M. Manfredi, R. Vezzani, S. Calderara, and R. Cucchiara. (2014) “Detection of static groups and crowds gathered in open spaces by texture classification,” *Pattern Recognit. Lett.*, vol. 44, pp. 39–48.

[13] M. Parzych, A. Chmielewska, T. Marciniak and A. Dabrowski. (2013) “Automatic people density maps generation with the use of movement detection analysis”, DOI. 10.1109/HSI.2013.6577798.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. (2014) "Rich feature hierarchies for accurate object detection and semantic segmentation". *In Computer Vision and Pattern Recognition (CVPR),* 2014 IEEE Conference on, pages 580–587.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. (2016) "You only look once: Unified, real-time object detection,".

[16] Zhong-Qiu Zhao, Peng Zheng and Shou-Tao Xu. (2019) "Object Detection with Deep Learning: A Review," *in IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*, ID: arXiv:1807.05511v2.

[17] Shubham Shinde, Ashwin Kothari, Vikram Gupta. (2018) "YOLO based Human Action Recognition and Localization". *International Conference on Robotics and Smart Manufacturing (RoSMa2018).* DOI 10.1016/j.procs.2018.07.112.Vol. 133, pages *831-838*.

[18] Joseph Redmon and Ali Farhadi. (2018) "YOLOv3: An Incremental Improvement," arXiv:1804.02767v1.

[19] N. G. El-gamal and H. E. Moustafa. (2017) "A New Combination Method for Background Subtraction in Video Sequences,".

[20] A. On. (2017) "A Novel Background Subtraction Algorithm For Person Tracking Based On K-NN," pp. 125–136.

[21] Xuegang Hu and Cheng He. (2016) "Moving Object Detection Algorithm Based on Gaussian Mixture Model and HSV Space," *British Journal of Applied Science & Technology***.** DOI: 10.9734/BJAST/2016/24249.

[22] https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv