Identification and Tracking of Groups of People Using Object Detection and Object Tracking Techniques

Tharuja Sandeepanie, Subha Fernando

***Abstract*— Object detection is one of the most important areas in the fields of Data Science and Computer Vision. In this paper, we present a novel approach to identifying and tracking groups of people, couples, and individuals in videos by using deep learning-based object detection and object tracking techniques along with a proposed grouping algorithm. In this approach, transfer learning is applied on YOLO v3 model for the detection of people in video frames, and Deep SORT is applied for tracking each detected person throughout the video. Results obtained from person detection and person tracking were used by the proposed grouping algorithm to identify and track groups, couples, and individuals who are appearing in input videos. Our proposed grouping algorithm is based on the proximity between each individual and the time duration that proximity is maintained for. It also considers how to identify and track groups, when people are moving within the groups. This approach was evaluated using CCTV videos captured from the restaurant domain and it was able to perform the group detection and tracking tasks successfully with a precision of 0.7083, recall of 0.7906 and F1 score of 0.7471.**

***Keywords*— object detection, object tracking, group detection, group tracking**

I. INTRODUCTION

Object detection is one of the hot research topics among Computer Vision and Data Science research communities. It is a technology, where things like humans, buildings, cars etc., can be detected as objects in images or videos. Object detection can be linked with similar Computer Vision techniques like image recognition, image segmentation, object tracking and image captioning, etc. In such cases, it helps understand and analyze the scenes in images or videos. At present, we are capable of generating applications as solutions to real-world problems by using object detection techniques along with object tracking [1]. In our proposed system, we focus on the problem of identification and tracking of groups of people, couples, and individuals in a given video.

We specifically consider this problem in relation to the restaurant domain. Currently, restaurants process huge amounts of data related to their business operations daily.

Correspondence: K.M.T. Sandeepanie (E-mail: kmtsandeepanie@gmail.com) Received:21-01-2022 Revised:13.01.2021 Accepted: 26-03-2023

K.M.T. Sandeepanie, K.S.D. Fernando, Faculty of Information Technology, University of Moratuwa, Katubadda, Sri Lanka. (kmtsandeepanie@gmail.com, subhaf@uom.lk)

DOI:

Based on this business-related information, organizations need to take important decisions on time [2], [3]. In this case, identifying how people come into the restaurant is one factor, restaurant owners need to consider to assess the business performance and to take strategic decisions. In other words, customers can arrive in restaurants either as a group (family or friends' group) or as a couple or as an individual (alone). Identifying what categories of people arrive and at which time most categories are willing to come will be helpful to change certain arrangements in the restaurant and to provide good customer service by making a flexible environment for them. So, the restaurant owners can obtain a better understanding of customers to take the necessary steps in the future. This information also will be helpful to take decisions regarding the type of promotions, and the time, and group to which it should be offered. In this situation, it is essential to have a system that can collect data and identify whether the customers are arriving as groups, couples, or individuals. This is the problem we address, which required to develop this system. The proposed system will be helpful not only in the restaurant domain but in other different domains too.

In the proposed solution, if a set of people maintains a proximity which is less than a minimum distance for at least a specified minimum number of frames in the video, we identify that set of people as either a "group" or a "couple". And, if a person does not have a proximity with any other one (unpaired) for at least a specified minimum number of frames, that person is considered as an "individual" in that frame. We used CCTV videos obtained from a restaurant as input for our system. The proposed system consists of a retrained YOLO v3 model, Deep SORT framework, and our proposed grouping algorithm to overcome the problem of identification and tracking of groups, couples, and individuals.

The rest of the paper consists of a review of all the related work to our research area under the title "related work", sections on methodology which describes our approach to identifying and tracking groups of people, our implementation, the results we obtained, the discussion which is an interpretation of the results, and then the conclusion of our research work.

II. RELATED WORK

A. Object Detection

Object detection determines both locations of objects via bounding boxes and the classes of those located objects. There are both machine-learning-based approaches and deep learning-based approaches for object detection. But today, most researchers use techniques of deep learning for object

detection because it has the ability to extract the features automatically with the help of algorithms such as Convolution Neural Networks (CNN). Object detection methods based on deep learning mainly include region proposal-based methods and unified pipeline framework-based methods.

Ross Girshick et al. proposed a region proposal-based object detection model which is named Region-based Convolutional Neural Network (R-CNN) [4]. This model uses the selective search algorithm which generates 2000 amount of regions from the input image. These regions are called region proposals. Therefore, without using a large number of regions for classification, this model can work with only 2000 regions. Those generated 2000 candidate region proposals are given as inputs into a CNN model which generates a 4096-dimensional feature vector as the result. Here, this CNN model works as a feature extractor. Finally, the R-CNN model runs a support vector machine (SVM) which takes the extracted features as inputs and classifies the existence of the object within each region. A major drawback of this model is that it takes a severe amount of time to train since it has to run the classifier for 2000 amount of region proposals for just one image. Also, it spends about 47 seconds for each test image so it is difficult to use this model for real-time object detection. Generating bad candidate region proposals can also occur since no learning happens in the selective search algorithm.

The fast R-CNN model [5] was also proposed by Ross Girshick et al. by solving some issues of the R-CNN model and making it faster in detecting objects than R-CNN. Without giving region proposals to the CNN model to extract features as in R-CNN, the fast R-CNN model gives the input image directly to the CNN model and it outputs a convolutional feature map which is then used for identifying the region proposals using the selective search algorithm. Here, a softmax layer is used to determine the class of the region. Since we apply the input image directly into the CNN model only once without giving 2000 amount of region proposals every time into the CNN, fast R-CNN is faster than R-CNN but including region proposals slows down the model performance significantly.

Shaoqing Ren et al. proposed the faster R-CNN object detection model [6] which is derived from R-CNN and fast R-CNN. Here, similarly in the fast R-CNN model, the input image is fed into a CNN model which outputs a convolutional feature map. The selective search algorithm used in R-CNN and fast R-CNN models can lead to producing bad candidate regions, increase the detection time and make the model slow down. Therefore, the faster R-CNN model uses another network to generate the region proposals on the convolutional feature map without using the previous selective search algorithm. Since this model is much faster than R-CNN and fast R-CNN models, the faster R-CNN model can be used for detecting objects in real time as well.

One of the unified pipeline framework-based methods proposed in recent years is the You Only Look Once (YOLO) method [7]. YOLO reframes object detection as a single regression problem. YOLO was implemented as a CNN which looks at the complete image at once and simultaneously predicts bounding boxes and class probabilities directly for the full image in one run of the algorithm. Considering popular state-of-the-art object detection models, the YOLO model performs efficiently with a balanced FPS and mAP score. It is extremely faster and less likely to predict false positives on the background than Region-based CNN (R-CNN) so it can be applied to real-time detection. YOLO is extremely simple and achieves end-to-end target detection without a complex pipeline. However, YOLO struggles to detect small objects which appear in groups and makes more localization errors. Fig 1 shows the YOLO network architecture.

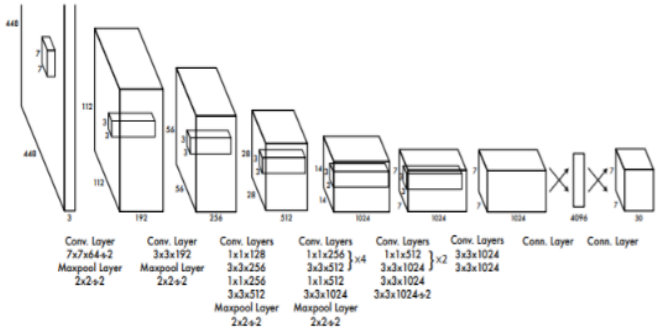

Fig. 1 Network architecture of the YOLO model

A real-time person and car detection system based on modified YOLO v1 has been proposed in [8] with the aim of having both high object detection accuracy and real-time operation. Even with a smaller number of convolutional layers, the model has enabled to improve the detection accuracy of small objects and real-time operation by employing larger grid cells. But it led to a decrease in the speed performance of the model. A lightweight CNN has been introduced for human object detection by Nikouei et al. with an affordable computation workload to an edge device [9]. This has verified that the L-CNN is better for smart surveillance tasks with decent accuracy and reasonable processing speed. Furthermore, it has shown that the L-CNN could handle complex situations where human objects are not completely in the frame.

B. Object Tracking

Simple online and real-time tracking (SORT) [10] is a simple object-tracking approach which performs significant results at high frame rates while achieving good tracking accuracy and precision. It executes Kalman filtering in image space and also uses the Hungarian method for frame-by-frame data association with an association metric which computes bounding box overlap. A major drawback in the SORT framework is that it generates a higher amount of identity switches which leads to a lack of capability to track through occlusions. The reason for this deficiency is that the association metric used by this approach is only precise when there is a low state estimation uncertainty.

This issue was addressed by the Deep SORT (Simple online and real-time tracking) framework which is one of the most popular object-tracking frameworks used today [11]. It uses a more informed metric which integrates appearance and motion data together. This is a deep learning-based approach to tracking custom objects in a

video. Integrating a CNN model improves its robustness against misses and occlusions while maintaining the efficiency and suitability to apply for online scenarios. It keeps track of each object under consideration by mapping unique identifiers. For effective tracking, the Kalman filter and the Hungarian algorithm have been used here [12]. Kalman filter was recursively used for a better association, and it can predict future positions based on the current position. Hungarian algorithm was used for association and id attribution that identifies if an object in the current frame is the same as the one in the previous frame.

C. Group Detection

Most researchers have focused on the task of detecting and tracking a single person but identifying a set of people as a group and tracking a group is not a much-considered research topic. An approach to monitoring social distancing via object detection is proposed in [13]. This model could detect people via the fine-tuned YOLO model, track detected people using Deep SORT techniques, and then it could identify groups of people who are violating social distancing. In this approach, it considered only the proximity of the people in the current frame so that it could detect a set of people who were close in the current frame as a group. According to this approach, if an individual goes close to a particular group only in the current frame, he will also become a member of that group, which will give a detection of a false group. It is difficult to understand what happens if people move within the groups in this system. Also, this approach has not focused on the task of tracking a detected group.

III. METHODOLOGY

Our novel approach keeps tracking the proximity of each pair of people throughout the video and detects a group if this proximity is maintained for at least a specific time duration. It does not identify groups by considering the proximities just only in the current frame as in previous approaches. Also, our approach assigns a group id for each group and via this id, each detected group is tracked throughout the video. The same process is done to identify a person as an individual.

A deep learning-based approach consisted of both object detection and object tracking techniques was used for identifying groups, couples, and individuals. This approach takes videos as the inputs. A YOLO v3 object detection model was fine-tuned to detect only people in the video inputs [14]. This retrained YOLO model generates bounding boxes enclosing each person in the video (object localization). And then these bounding box coordinates and the confidences of each person obtained from the YOLO model are passed to the Deep SORT object tracking framework in order to track each detected person in the video. This gives a unique id for each person and tracks each person throughout the video.

By using these person ids and the bounding box coordinates of each person, we developed a grouping algorithm which identifies groups of people, couples, and individuals in the given video and tracks them throughout the video by assigning a group id. The flow chart of the grouping algorithm is shown in Fig 3. At the end of the video, we can obtain the individuals' person ids, the total number of individuals, group/couple ids, the number of members in that group/couple, and the group type (whether a 'group' or a 'couple') as the final result, and then this result will be saved in the database. The process flow diagram of the overall system is shown in Fig 2.

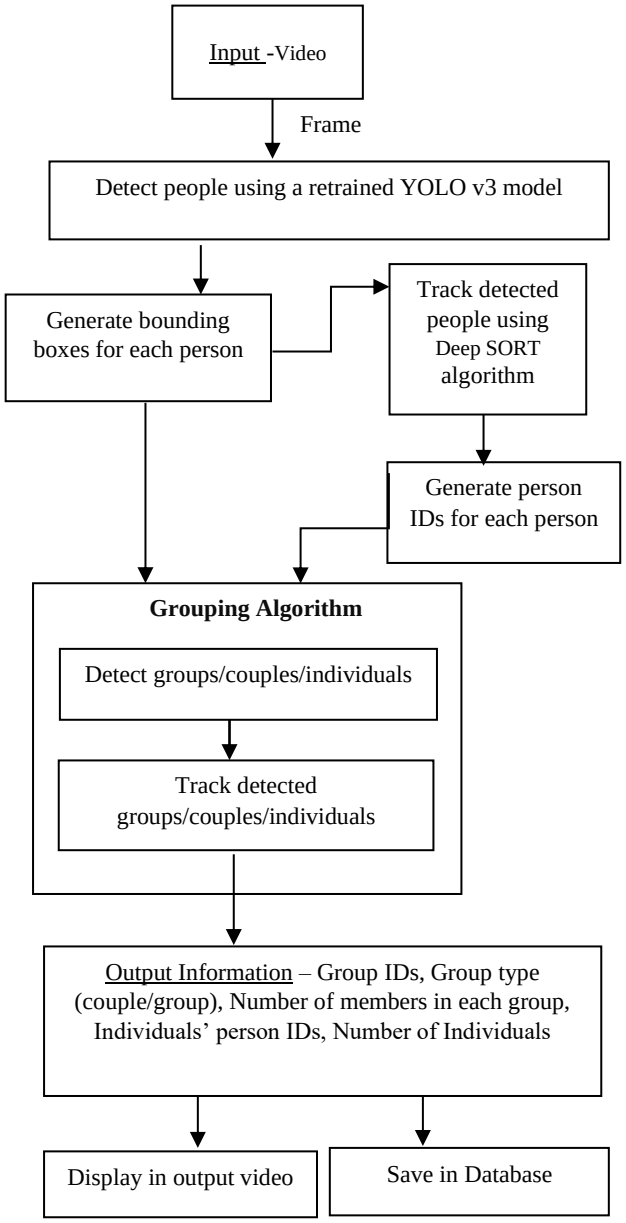

Fig. 2 Proposed system

IV. IMPLEMENTATION

A. Training YOLO for Detection of People

The original darknet YOLO v3 model recognizes 80 different classes in images and videos. To implement this module, first a YOLO v3 object detection model was retrained to detect only a single class (person class) in a given input. To retrain the YOLO v3 model, a PyTorch version of the YOLO v3 model [15] was used since it has weights which are far better than other YOLO v3 considering the mean average precision, and it facilitates model training using PyTorch. We changed the configurations of this YOLO model to detect only a single class. So, the number of classes in all YOLO layers were

changed to 1 and the number of filters in the last convolutional layers (which are before the YOLO layers) were changed to 18 according to equation (1).

Number of filters = (number of classes + 5) * 3 (1)

Without using all the available data to train the model, the dataset should be split into the training set, validation set and testing set for the purpose of avoiding overfitting. Also, the dataset should be divided in a way that a higher amount of data will be used for training the model. According to empirical studies [16], the best results can be achieved when diving the data set as 70-80% of the data for training and 20-30% of the data for both validation and testing. It ensures the model is trained on enough data while having enough data for validation and testing. To make sure the model is being evaluated on unseen data, having at least 10% of data as testing data is generally recommended. So, we used the 70-20-10 ratio, which is a common ratio optimal for small datasets, to split our images dataset; 70% to train the model, 20% to validate the model (tune model parameters) and 10% to test the final performance of the model.

When considering the dataset, first, we obtained 1000 images and annotations per each from the COCO dataset. Among them, 700 images were for training, 200 images were for validation, and 100 images were for testing according to the 70%, 20%, and 10% ratios. The model was retrained in the GPU-enabled Google Co-lab development environment using PyTorch with a batch size of 8 for 200 epochs.

But training only from the COCO dataset was not accurate enough to detect people in the CCTV videos because the top view of the people (heads) appears in CCTV videos and the COCO dataset doesn't have such images sufficiently. Hence, a custom person dataset was created from a collection of CCTV videos with 600 images. The "LabelImg" tool was used to create labels (annotations) for those 600 images. Then the YOLO model was again re-trained by using both our custom person dataset and COCO dataset in the GPU-enabled Google Co-lab development environment using PyTorch with a batch size of 8 for 200 epochs. 1120 images were for training, 320 images were for validation, and 160 images were for testing according to the 70%, 20%, and 10% ratios. And then we applied the Deep SORT object tracking framework along with this retrained YOLO model for the tracking of each detected person.

B. Grouping Algorithm

A "Grouping Algorithm" was implemented which identifies groups, couples, and individuals in the video and tracks them throughout the video by assigning a group id. The flow chart of the grouping algorithm is shown in Fig 3. Here, if a set of people maintains a proximity which is less than a minimum distance threshold for at least a specified minimum number of frames in the video, we define it as a "Group". Similarly, if an individual does not maintain a proximity with any other for at least a specified minimum number of frames, we define him as an "Individual". Before grouping, we need to specify this distance threshold and the minimum time duration (number of frames) for both groups and individuals. These thresholds need to be adjusted after analyzing the relevant domain.

Each individual is associated with three-dimensional feature space (x,y,d), where x, y corresponds to the centroid coordinates of the bounding box and d defines the depth of the individual as observed from the camera. Here, depth means the distance from the observable camera to the detected person, and it is calculated as equation (2) [17]. Here w is the width of the bounding box and h is the height of the bounding box.

$$d = ((2*3.14*180)/(w + h*360)*1000 + 3) \quad (2)$$

The grouping algorithm is updated frame by frame after passing each person's centroid coordinates (x,y), depth (d), and the person id (given by Deep SORT) in each frame with the frame number to the grouping algorithm. In order to track the groups and couples, a Python dictionary is maintained in this algorithm for groups/couples, and saves each pair's following details inside it.

- Frame count - keeps a count of the total number of frames where a pair keeps the proximity.
- Group id - if a pair belongs to a particular group, keeps the id of that group.

In order to track the individuals, another Python dictionary is maintained in this algorithm for individuals, and saves each one's following details inside it.

- Frame count - keeps a count of the total number of frames where the person does not have a proximity with any other one (unpaired).

The flow chart of the grouping algorithm is shown in Fig. 3. This grouping algorithm consists of several functions. In the first function, pairwise L2 norms for each pair of detected people in the given frame are calculated using each one's centroid coordinates and depth (x,y,d) according to equation (3).

$$||D||_2 = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \quad (3)$$

Here for this scenario, n =3 and q, p is the x, y, d three dimensions of two people. This will return all the pairs of people who are in a proximity less than the minimum distance threshold and the rest unpaired people in each frame.

In the next function, the pairs which have intersecting person ids are merged and the union of those pairs forms a set of people who are close. Then all the pairs (including intermediate pairs) for each formed set of people who are close, are generated, and their details are saved inside the group dictionary. When a set of pairs' frame count is higher than the minimum time duration, those pairs are merged to form a group and assigned a group id to all the pairs. There, if any pairs already have a group id, then the group id will be the minimum of it otherwise group id will be a new number. Similarly, unpaired peoples' details are saved in the individuals' dictionary. If a person does not have a proximity with any other one (unpaired) for more than the minimum time duration, that person is considered as an individual in that frame.

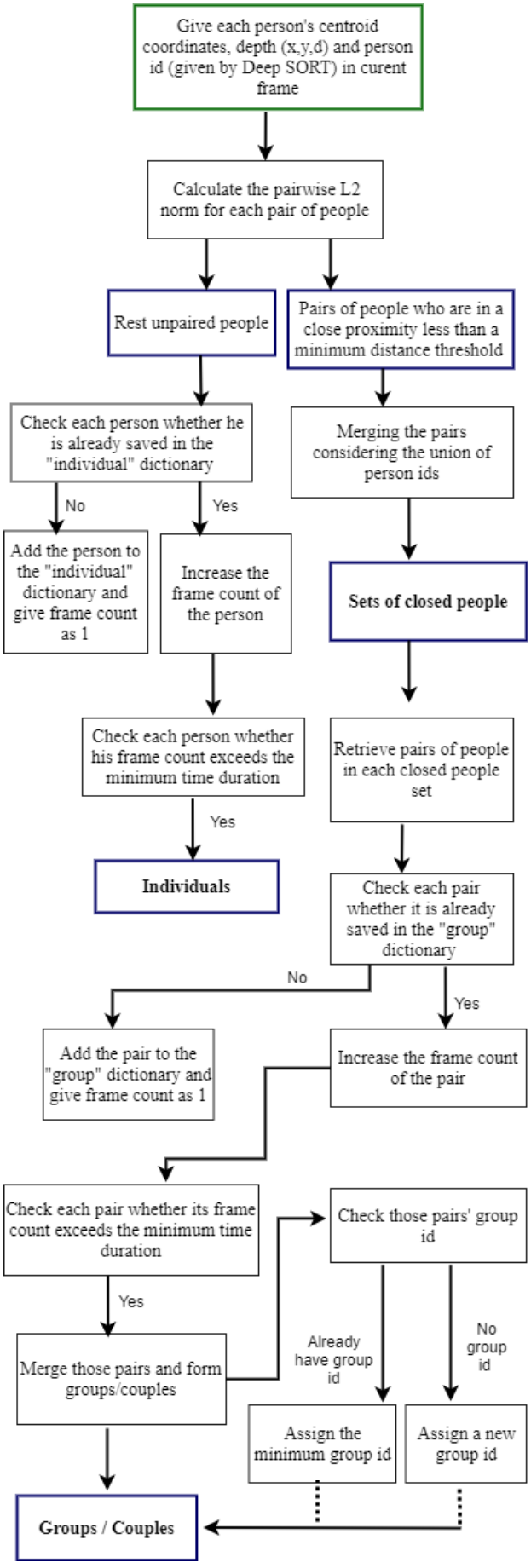

Fig. 3 Flow chart of the grouping algorithm

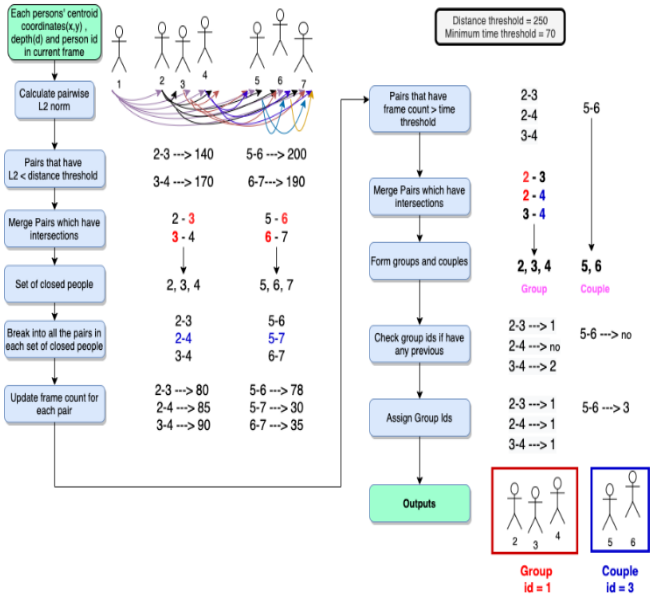

Fig. 4 Example scenario for grouping of people

TABLE I

RESULTS OF THE PROPOSED SYSTEM

Group Id	Number of Members	Group Type
1	2	Couple
2	3	Group
3	5	Group
4	6	Group
5	2	Couple
Individual Ids:	[2, 4, 11, 1] Total number of individuals = 4	

This grouping algorithm returns identified groups/couples and individuals in each frame. Then those group ids and their group type and the individuals' ids are displayed in each frame of the output video. Since the number of people in a group can vary from frame to frame, the final group size of a particular group id will be the size that has the highest number of occurrences throughout the video. Then finally, these group details and individual details will be saved in a cloud database. This final result is received as the example results shown in Table I. Example scenario to explain the process of the grouping of people is depicted in Fig 4. It shows how the model identifies groups of people or couples from the people who are numbered from 1 to 7 using our grouping algorithm.

In this grouping algorithm, once a group is identified, we cannot consider that that set of people will remain as a fixed

group throughout the whole video. Members in a detected group may change because the below scenarios can happen in real situations. When designing the flow of the algorithm, we specifically took into consideration how we group people and assign group ids in the below situations where people can move within the groups.

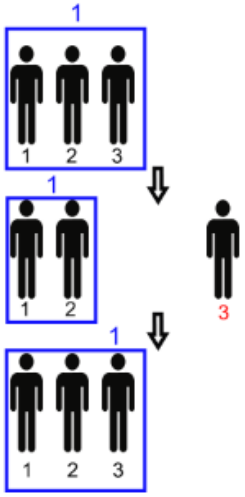

Fig. 5 Scenario-1

- Once a group is defined, later a member in that group will get separated from that group and after a while, he again joins that group. Fig 5 shows this scenario.

Here, at first, people with id 1, 2, and 3 are within a proximity which is less than the distance threshold, for more than the minimum time duration. Hence, they are considered as a group. Once the person with id 3 gets separated, his distance to others is more than the distance threshold and he is not paired with anyone. So, only people with id 1 and id 2 are considered as the group now. Later, the person with id 3 again joins and then he is again within proximity of others. Since, now they are within a proximity which is less than the distance threshold for more than the minimum time duration, once again they can be considered as a group.

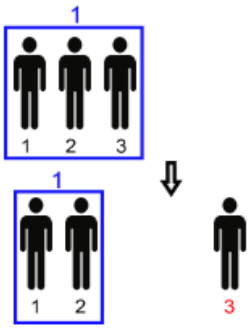

Fig. 6 Scenario-2

- Once a group is defined, later a member in that group will get separated and completely leaves that group. Fig 6 shows this scenario.

Just like in the previous case, at first, people with id 1, 2, and 3 are in a proximity which is less than the distance threshold, for more than the minimum time duration so that they are considered as a group. Once, the person with id 3 leaves that group, his distance to others is more than the distance threshold and he is not paired with anyone after that. Hence, people with id 1 and 2 are considered as the group. If the person with id 3 remains alone without being paired with anyone for more than the time threshold, he can be considered as an individual.

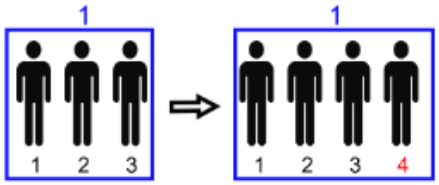

Fig. 7 Scenario-3

- Once a group is defined, later a new person may join that group. This scenario is depicted in Fig 7.

Here, at first, people with id 1, 2, and 3 are within a proximity which is less than the distance threshold, for more than the minimum time duration, so that they are considered as a group. Then, a new person with id 4 joins that group. Now, if his distance to either person 1, 2 or 3 becomes less than the distance threshold, he can be paired with them. Once, he keeps his proximity with others for more than the minimum time duration, he can be added to this group.

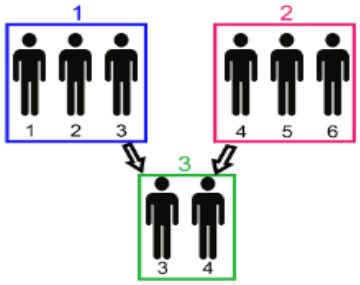

Fig. 8 Scenario-4

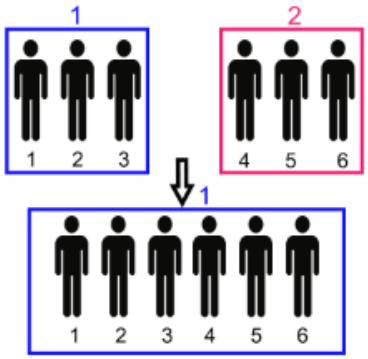

Fig. 9 Scenario-5

- Two people from two different groups can join and form one group. Fig 8 shows this scenario.

Here, the person with id 3 and the person with id 4 leave group id 1 and 2 respectively, and those two keep remaining within the proximity which is less than the distance threshold for more than the minimum time duration. Hence, they can be considered as a couple. Since it is a new pair (person id 3 and id 4), the next group id which is 3 will be assigned to this couple.

- Two different groups may join together as one group. This scenario is shown in Fig 9.

Here, at first, people with id 1, 2, and 3 belong to group-1, and people with id 4, 5, and 6 belong to group-2. When these two groups get closer, their distance to each other becomes less than the distance threshold. Once they keep their proximity for more than the minimum time duration, all of them are considered as one new group. When considering the group ids of the pairs in this new group, some pairs previously had been assigned the group id as 1 while some had been assigned the group id as 2. As they already have group ids, once they become one single group, the minimum value of those group ids will be assigned to this new group. In this case, the minimum group id is 1 and it will be the group id of the new group.

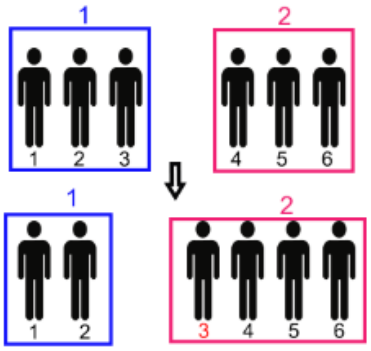

Fig. 10 Scenario-6

- A person in one group later joins another group. This scenario is shown in Fig 10.

Here, at first, people with id 1, 2, and 3 belong to group-1, and people with id 4, 5, and 6 belong to group -2. Then, the person with id 3 leaves group-1 and gets closer to group-2. His distance with person-1 and 2 gets more than the distance threshold so he will no longer belong to group-1. His distance with either person-4 or 5 or 6 becomes less than the distance threshold and if that distance keeps remaining for more than the minimum time duration, person-3 can be added to group-2. Then, the group ids of the new pairs which are 3-4, 3-5, and 3-6 will be assigned as 2.

In our approach, grouping is done considering the closeness of people in relation to distance in the current frame and does not consider the previous status of a person. Also, this keeps saving previous frame count details and group id details of each pair. Hence, all the above scenarios were able to be handled from this grouping algorithm. It identifies groups or couples or individuals correctly and assigns group ids considering all these scenarios.

V. EVALUATION AND RESULTS

The proposed system was evaluated on both person detection performance and group detection performance. Below metrics were used to measure the performance of retrained YOLO model which does person detection.

- Precision - Ratio of true positive (TP) and the total number of predicted positives.
- Recall (TPR) - Ratio of TP and the total of ground truth positives.
- F1 Score - Function of Precision and Recall. This is used when we need to seek a balance between Precision and Recall.
- Mean average precision (mAP)@ IoU=0.5

First, we retrained and tested the YOLO model by only using the COCO dataset person images. Since training only from the COCO dataset was not accurate enough to detect people in the CCTV videos, we retrained the model by using both COCO person images and our custom person dataset created using CCTV images. All training and testing accuracy results of the retrained YOLO v3 model for the detection of people in both scenarios are shown in Table II, Fig 11, and Fig 12.

Table III illustrates a comparison of person detection results between our person detection model and other state-of-the-art object detection models in terms of mean average precision. Our retrained person detection YOLO model has a higher mean average precision than the person-class average precision of the original YOLO model [7] and R-CNN model series which were measured on the PASCAL Voc-2012 test dataset. But when compared to the fine-tuned person detection YOLO model [13] which was tested with the dataset acquired from the open image dataset (OID) repository and frames of surveillance footage, it has a slight increment in mean average precision than our person detection model. However, modified YOLO_7x7, YOLO_9x9 and YOLO_11x11 models in [8] which were tested with the INRIA dataset for the person and car detection possess lower precision values than our retrained person detection YOLO model. Overall, it is observed that our person detection YOLO model which was retrained using both COCO person dataset and our custom person dataset achieved better results with significant accuracy.

Next, we evaluated the accuracy of the system for the detection of groups, couples, and individuals. Here we considered the performance of the overall integrated system which means the functioning of both retrained YOLO object detection model and the Deep SORT object tracking framework along with the grouping algorithm.

TABLE II

TRAINING AND TESTING ACCURACY RESULTS OF OUR RETRAINED YOLO V3 MODEL FOR THE DETECTION OF PEOPLE

	Precision	**Recall**	**mAP@0.5**	**F1 Score**
Using COCO person dataset				
Training	0.557	0.769	0.707	0.646
Testing	0.572	0.633	0.507	0.601
Using both COCO person dataset and our person dataset				
Training	0.739	0.809	0.795	0.773
Testing	0.746	0.818	0.801	0.780

TABLE III

COMPARISON OF PERSON DETECTION RESULTS BETWEEN OUR MODEL AND PREVIOUS OBJECT DETECTION MODELS

Model	mAp@0.5
R-CNN [4]	0.532
R-CNN VGG [4]	0.600
Fast R-CNN [5]	0.720
Faster R-CNN [6]	0.796
Fast R-CNN + YOLO [7]	0.747
Original YOLO model [7]	0.635
Retrained YOLO model [13]	0.846
YOLO_7 x7 [8]	0.423
YOLO_9 x9 [8]	0.438
YOLO_11 x11 [8]	0.441
Our retrained YOLO model	**0.801**

TABLE IV

EVALUATION RESULTS OF THE PROPOSED SYSTEM FOR THE DETECTION OF GROUPS

Precision	Recall	F1 Score
0.7083	0.7906	0.7471

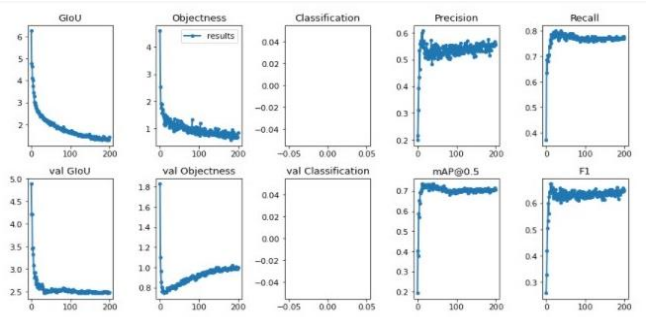

Fig. 11 Evaluation results of the retrained YOLO v3 model for detection of people. (Using COCO person dataset)

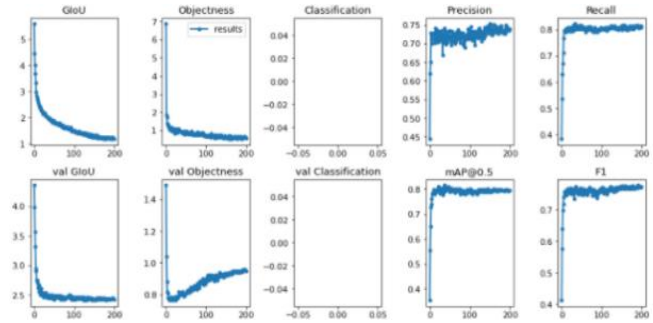

Fig. 12 Evaluation results of the retrained YOLO v3 model for detection of people. (Using both COCO person dataset and our custom person dataset)

That evaluation was done by applying the system for several CCTV video inputs and calculating the performance matrices. Table IV shows the evaluation results of the overall proposed system for the detection of groups, couples, and individuals. As, for the output, the same input video is generated indicating each identified group or couple by using their group id and by a polyline that goes over each member belonging to that group. Fig 13 shows sample output video frames of this proposed system which displays the detected groups, couples, and individuals in given input videos.

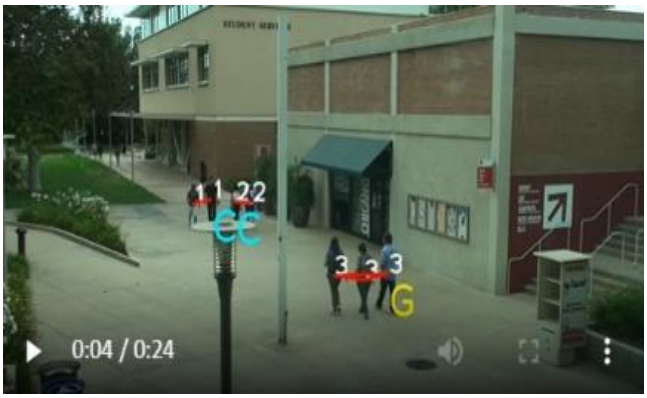

Fig. 13 Sample outputs of the proposed system

VI. DISCUSSION

This approach has successfully performed group identification and group tracking tasks accurately but some limitations are involved. This approach highly depends on the accuracy of the Deep SORT framework because, sometimes, an id that has been assigned to one person may be assigned to another person later, especially when two people interact with each other. Also, if an already tracked person is covered by another object or a person a new id can be assigned to him once he appears. In such cases, that person again needs to wait for a minimum time duration (threshold number of frames) in order to belong to the group. Another limitation is threshold values used in the grouping algorithm may vary according to the different scenarios. So, we need to adjust those thresholds according to the situation or as relevant to the domain by analyzing the behaviour of the people who appear in the videos. Also, this approach is not possible, if the camera is moving (not stable as a CCTV camera) or has different views because tracking of people can't be done continuously in such circumstances.

In real situations, a set of people will not remain as a fixed group because members in a detected group can vary from time to time. Also, when a person is covered by another object that stays in front of him, at those frames, his person id will not return for the grouping algorithm. These problems were also considered and solved by this algorithm by keeping the previous frame count details of each pair and by keeping tracking each detected group. Considering sets of people who are close only in the current frame and not taking into account the states of a person in previous frames is also a major feature in this approach for correctly identifying groups.

The most, specific feature in this system is if a person who is just close to a particular group only for a few frames or in the current frame, will not belong to that group unless he keeps the proximity at least for the defined minimum time duration. This avoids the identification of false groups or adding false members to a real group. So, only true groups will be identified from this algorithm. Here, true groups means groups of real friends, family or couples. If a person does not maintain a proximity with any other one for more than the minimum time duration, that person is considered as an individual. So, apart from identifying groups and couples, this approach also identifies individuals as people who come alone.

VII. CONCLUSION

In this paper, we have presented a novel approach to identifying and tracking groups of people, couples, and individuals in videos by using deep learning-based object detection and object tracking approaches. YOLO v3 object detection model was retrained to detect people in the video inputs. By using Deep SORT framework, each detected person is tracked throughout the video by assigning a unique id. With the aid of bounding boxes data and person ids generated by YOLO and Deep SORT, we have proposed a grouping algorithm which identifies people appearing in the videos as either groups of people or couples or as individuals and tracks each detected group or couple via assigning a group id. The output video indicates each identified group or couple by using their group id and by a polyline that goes over each member belonging to that group. Also, each detected individual is indicated via his person id in the output video. The proposed approach was evaluated using CCTV videos captured from the restaurant domain. The experimental results demonstrate that, this approach has successfully performed group identification and group tracking tasks with a higher accuracy.

ACKNOWLEDGEMENT

Foremost, I would like to express my sincere gratitude to Dr (Mrs) KSD Fernando for providing better guidance and valuable suggestions, and for advising throughout this entire research. The technical assistance and the immense support given to me and especially her knowledge of deep learning helped me a lot to make this research a success. And also I extend my gratitude to CodeGen International (Pvt) Ltd for their valuable support in the completion of this research. Also, I would like to thank the staff at the CurryMuch restaurant for supporting me by setting up and providing CCTV videos. Finally, thanks and appreciation are also extended to all the academic staff of the Faculty of Information Technology, University of Moratuwa for providing technical support and insights throughout this research project.

REFERENCES

[1] A. Bathija and G. Sharma, "Visual Object Detection and Tracking using YOLO and SORT," Int. J. Eng. Res., vol. 8, no. 11, p. 4.

[2] Sandra Durcevic in Business Intelligence, "How Restaurant Analytics Can Make Your Business More Profitable," datapine, May 16, 2019. https://www.datapine.com/blog/benefit-from-your-data-with-restaurant-analytics/

[3] Ryan Andrews, "How Restaurants Are Using Data and Analytics to Increase Profits," Eat App Restaurant Management Software, Jul. 14, 2021. https://restaurant.eatapp.co/blog/restaurant-data-and-analytics-increase-revenue

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," ArXiv13112524 Cs, Oct. 2014, Accessed: Jun. 06, 2020. [Online]. Available: http://arxiv.org/abs/1311.2524

[5] R. Girshick, "Fast R-CNN," ArXiv150408083 Cs, Sep. 2015, Accessed: Jun. 06, 2020. [Online]. Available: http://arxiv.org/abs/1504.08083

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," ArXiv150601497 Cs, Jan. 2016, Accessed: Jun. 06, 2020. [Online]. Available: http://arxiv.org/abs/1506.01497

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," ArXiv150602640 Cs, May 2016, Accessed: Jun. 06, 2020. [Online]. Available: http://arxiv.org/abs/1506.02640

[8] M. H. Putra, Z. M. Yussof, K. C. Lim, and S. I. Salim, "Convolutional Neural Network for Person and Car Detection using YOLO Framework," vol. 10, no. 1, p. 5.

[9] S. Y. Nikouei, Y. Chen, S. Song, R. Xu, B.-Y. Choi, and T. R. Faughnan, "Real-Time Human Detection as an Edge Service Enabled by a Lightweight CNN," ArXiv180500330 Cs, Apr. 2018, Accessed: Jun. 06, 2020. [Online]. Available: http://arxiv.org/abs/1805.00330

[10] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple Online and Realtime Tracking," in 2016 IEEE International Conference on Image Processing (ICIP), Sep. 2016, pp. 3464–3468. doi: 10.1109/ICIP.2016.7533003.

[11] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," ArXiv170307402 Cs, Mar. 2017, Accessed: Sep. 12, 2021. [Online]. Available: http://arxiv.org/abs/1703.07402

[12] J. J. Tithi, S. Aananthakrishnan, and F. Petrini, "Online and Real-time Object Tracking Algorithm with Extremely Small Matrices,"

ArXiv200312091 Cs Stat, Mar. 2021, Accessed: Sep. 12, 2021. [Online]. Available: http://arxiv.org/abs/2003.12091

[13] N. S. Punn, S. K. Sonbhadra, and S. Agarwal, “Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques,” ArXiv200501385 Cs, May 2020, Accessed: Jun. 10, 2020. [Online]. Available: http://arxiv.org/abs/2005.01385

[14] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” ArXiv180402767 Cs, Apr. 2018, Accessed: Sep. 12, 2021. [Online]. Available: http://arxiv.org/abs/1804.02767

[15] Ultralytics, “YOLOv3 in PyTorch,” Dec. 12, 2018. https://github.com/ultralytics/yolov3

[16] A. Gholamy, V. Kreinovich, and O. Kosheleva, “Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation”.

[17] Pias Paul, “Object detection and distance measurement,” Sep. 01, 2019. https://github.com/paul-pias/Object-Detection-and-Distance-Measurement