International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 9, No.6 (Nov-2020)

http://dx.doi.org/10.12785/ijcds/090605

Video-Based Automated Pedestrians Counting Algorithms for Smart Cities

Willson Meli[1], Fred Lacy[1] and Yasser Ismail[1]

[1]Electrical Engineering Department, College of Sciences and Engineering, Southern University and A&M College, Baton Rouge, LA, USA

Received 6 Jun.2020, Revised 16 Jul. 2020, Accepted 29 Jul. 2020, Published 1 Nov. 2020

Abstract: Smart cities possess several technologies in collecting pedestrian activity data, which may be used to manage city planning. A growing body of research exists on video processing based pedestrian counting methods, due to the development of new computer vision techniques. This research reviews different, vision-based methods for counting pedestrians and applies a specific counting method which is formed by a combination of You Only Look Once Version 3 (YOLOv3) and Simple Online Real-time Tracking (SORT) with a deep association metric. The results suggest that although clustering, as well as the direction and intensity of pedestrian traffic, achieves a minimal effect on the count, occlusion constitutes the main source of errors. Adequate training may serve to increase accuracy.

Keywords: Smart Cities, YOLO Algorithm, Pedestrians Counting Algorithm, Simple Online Real-time Tracking

1. INTRODUCTION

A pedestrian count determines the volume and direction of the pedestrian traffic in the Central Business District (CBD) at a specific time and place. In that sense, a pedestrian count assesses the need and efficacy of numerous pedestrian planning initiatives at particular locations – in this instance, the City Centre. The count represents a convenient and inexpensive way to provide vital objective data.[1]. A pedestrian delivers an unquestionable impact on the structure of the CBD since the pedestrian traffic energizes the activity of the city's traffic-generating centers. Since the study analyzes how pedestrians circulate and gather in the downtown area, the research can consider in-depth how to utilize and organize the CBD [1].

The numerous benefits of pedestrian count may be summarized into three essential aspects of CBD pedestrian planning. First, a pedestrian count may improve a pedestrian's pleasure. Although the enjoyment and comfort of walking are primarily influenced by the overall downtown area in which the walker moves, unique features of the pedestrian circulation system may help ameliorate the sensations of an individual traveling by foot [1]. Artwork, trees, shrubs, and other street elements create sources of visual beauty that the pedestrian can appreciate. Kinds of facilities, inclusive of heated sidewalks, waiting stations, and canopies sidewalks can enhance the pedestrian's comfort in the experience. In this instance, the pedestrian count may be used to establish priorities for a situated improvement for pedestrian pleasure. The sites may be established regarding where these chosen stops would most benefit pedestrians [1].

Second, analyzing a pedestrian count tends to contribute to a better movement of the pedestrian. Multiple factors are impacting the choices of pedestrians to circulate directly and freely inside the CBD. One important factor is the extent of sidewalk congestion. The extent of sidewalk congestion is determined by two data: the pedestrian count and the sidewalk width. Sidewalk capacity could also become maximum at traffic intersections, where pedestrian numbers often become greatly increased when traffic lights halt the pedestrian flow. Another factor that can influence pedestrian motion is the presence of conflicts between pedestrians and automobiles [1]. Conflicts between pedestrians and vehicles mostly occur at downtown intersections, usually attenuated by control technologies and traffic signals. From this standpoint, there are multiple ways to use the findings of a pedestrian count. The knowledge identifies the necessity for the installation and operation of traffic signals. Add to traffic signals a plethora of physical aids such as the construction of a) underpasses and

E-mail: willoubien@gmail.com, fred_lacy@subr.edu, yasser_ismail@subr.edu

overpasses, b) refuge islands, c) barriers, and d) painted crosswalks [1]. A final factor in determining how well walkers move in the CBD is the availability of pedestrian routes and the land use design. Land utilization and the availability of pedestrian paths often restrict the distances pedestrians must walk and the chosen route inside the downtown area. The use of land in the CBD may be compared to a system that works to generate walker traffic, which is separated by variable distances [1]. Lengthy walks between pedestrian traffic generators are sometimes caused by the ineffective distribution of land use. If inconveniently located, unloading or final points for diverse means of transportation could compel pedestrians to take longer than average walks to maintain destinations inside the CBD.

Third, a pedestrian count keeps pedestrians safe. A conflict with car traffic affects a pedestrian's safety. Such conflict may not only cause a decreased mobility for the pedestrian, but such an issue can also be responsible for hazardous situations where the pedestrian may be injured. Consequently, steps are must be taken to lessen the conflict, and thus improving the mobility of the pedestrian in order to also would protect his/her health and safety [1]. This triad may be utilized in different strategies to guarantee the safety of the pedestrian: a) to assess the number of jaywalkers at street intersections or anywhere else as a fraction of the total pedestrian count; b) to extrapolate the results of a thorough study based on the pedestrian's observation of traffic signals, and; c) to find relationships between pedestrian numbers along adjacent sidewalks and car accidents [1].

Besides, the pedestrian count has several other applications not directly linked to pedestrian planning itself. These applications include the delimitation of not only the CBD's boundaries, but also central traffic district borders [2], the selection of public locations, such as libraries and retail store locations [3], and the analysis of the increase and decrease in CBD land worth [4].

2. LITERATURE REVIEW

The accuracy of pedestrian count methods determines how thoroughly the exposure for safety analysis may be measured, how well infrastructure developments and safety programs will be prioritized, and how efficiently the advantages of pedestrian projects, the models of pedestrian quantities, and the variations in pedestrian activity may be assessed. However, automobile counts in most communities are still much more popular than pedestrian counts. Moreover, current informational, non-standardized, pedestrian count methodologies allow for no estimation of weekly, monthly, or annual numbers [5].

To record manual counts, one may use clickers or collection data sheets in the field. Another way manual counts can be performed is by using video technology. The video technology process allows more precise, deliberate monitoring, as the video can be replayed, slowed down, or sped up [6]. Although using datasheets and clickers is the least accurate manual count methodology, the process is less costly than analyzing video data, since the application does not require specific equipment [6]. Automated count methods are usually less accurate than manual counts. However, inaccuracies may be caused by human error [7]. Video-based manual count accuracy depends on both the awareness and the degree of motivation of the observer. To improve the accuracy of such counting, one can either lower the number of details being recorded by the observer [6], or one can avoid continuous counts over long periods so that data collectors do not experience fatigue [7].

In the case of automated pedestrians counting, several technologies have been developed in recent years. Automobiles are easier to count than pedestrians since their path is much more constrained [7]. Consequently, it is crucial to understand the particular kind or kinds of pedestrian movements that must be enumerated before selecting a suitable automated counter. Size, legal restrictions, installation costs, data storage, accuracy, location, and maintenance costs are other essential considerations [8]. Common Options for automated counting devices include radio beams and passive and active infrared devices [6]. Less common options include Laser scanners, pressure or acoustic pads, and thermal cameras. Alternatively, some automated measurements may be used to quantify pedestrian traffic. These measurements can be captured via Wi-Fi technology, Bluetooth, or traffic signals recording pedestrian pushbutton use [9].

Existing approaches for pedestrian counting algorithms can generally be grouped into three categories: 1) clustering-based methods, which track the number of features of target objects; 2) regression-based methods, which use features of detection regions to learn a regression function and subsequently utilize that function for counting; and 3) detection-based methods, which aim at extracting the foreground, localizing the target, tracking objects, and finding trajectories [10].

Using clustering-based methods, numerous object features – such as points or people elements – are tracked; clustering feature paths allow people counting. For example, Brostow et al. [14] suggested a technique that initially tracks basic image features and then applies probability to group these features into clusters, depending on both the trajectory consistency throughout the image space and the space-time closeness. Antonini et al.

The primary goal of regression-based methods is to interpolate the variations into a scene, indicating the passing of pedestrians so that a regression function may be learned. For instance, Benabbas et al. [15] suggested a technique where the research collects image slices and then evaluates the optical flow. The study discussed features by linear regression and blob detector extraction, and then obtained pedestrian data relative to orientation, position, and velocity.

The techniques that fall under the detection-based category incorporate the same ordered processing pipeline described as follows. First, the video stream is processed for foreground extraction, then the target objects travel through detection and tracking [16]. Furthermore, the tracked paths are classified to determine the count of the target objects. There are four distinct groups among detection-based methods: 1) Depth video methods, 2) RGB video methods, 3) hybrid methods, and 4) deep learning framework methods [16].

The paper is organized as follows, in section 2, a literature review is elaborated. The problem formulation and motivation of this work is discussed in detail in section 3. The YOLOv3 algorithm is discussed in section 4. Simulation and implementation are discussed in section 5. A conclusion will be drawn in section 6.

3. PROBLEM FORMULATION AND MOTIVATION OF THIS WORK

This research has chosen to apply a deep learning object counter to solve the problem of pedestrian counting, as this method recently displayed considerable success in object classification and detection assignments [16]. In the past, Liu et al. [17] suggested a people counting model based on a basic, convolutional, neural network (CNN) and Spatio-Temporal Context modeling. Likewise, Wei et al. created a framework relying on supervised learning [18]. This framework, by associating a super-pixel multi-appearance with multi-motion characteristics, extracted Spatio-temporal features, then mixed with the VGG-16 model features. However, for the deep learning object counter of this research, the study will use YOLOv3 for detection, and SORT with deep learning association metric for tracking.

YOLO stands for You Only Look Once and is the very first tentative application in designing a fast, real-time object detector. YOLOv3 is the second and latest upgrade on the original YOLO version, which is suitable for detecting small objects (2). YOLOv3 was selected as this study's detector because the application is more precise than SSD; further, both YOLOv3 and its predecessor YOLOv2 are similar in performance with DSSD, as seen in Table 1. YOLOv3 is also faster than two-stage Faster R-CNN variants, using ResNet, FPN, G-RMI, and TDM [19]; SORT stands for Simple Online and Real-time and centers on frame-to-frame prediction and association. SORT, equipped with a deep learning association metric, also is called Deep SORT, indicating an improvement on the original SORT algorithm; Deep SORT allows object tracking over more prolonged periods of occlusion [20]. Deep SORT, as the study's research tracker, presents a simple and well-suited baseline and runs in real-time.

Moreover, Table 2 shows that the Deep SORT framework achieves better performance concerning the accuracy, compared not only with the original SORT but also with other online trackers such as POI and EAMITT [20].

TABLE 1. AP RESULTS FOR SOME TYPICAL DEEP LEARNING DETECTORS ON THE COCO DATASET [19]

	Backbone	**AP**
Two-Stage Methods		
Faster R-CNN+++	ResNet-101-C4	34.9
Faster R-CNN w FPN	ResNet-101-FPN	36.2
Faster R-CNN by G-RMI	Inception-ResNet-v2	34.7
Faster R-CNN w TDM	Inception-ResNet-v2-TDM	36.8
One-Stage Methods		
YOLOv2	DarkNet-19	21.6
SSD513	ResNet-101-SSD	31.2
DSSD513	ResNet-101-DSSD	33.2
RetinaNet	ResNet-101-FPN	39.1
RetinaNet	ResNeXt-101-FPN	40.8
YOLOv3	Darknet-53	33

TABLE 2. PERFORMANCE OF DIFFERENT OBJECT TRACKERS ON MOT[19]

Name of Tracker	**Type**	**MOTA**
KDNT	Batch	68.2
LMP	Batch	71
MCMOT_HDM	Batch	62.4
MOMTwSDP16	Batch	62.2
EAMTT	Online	52.5
POI	Online	66.1
SORT	Online	59.8
DEEP SORT	Online	61.4

4. COUNTING ALGORITHM

The counting algorithm that this study will be using combines YOLOv3 and the SORT with a Deep cosine metric. Following the video inputs, YOLO first performs pedestrian detection; the detected pedestrians then are tracked by Deep SORT for counting. Figure 1 presents the overall block diagram of the counting algorithm.

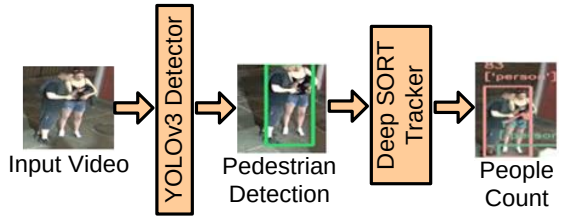

Figure 1. Overall Counting Algorithm.

When an image is processed by YOLO, all the objects present in the image are localized and classified at once. YOLO divides the input image into several grids, and each grid is in charge of detecting an object. The overall YOLOv3 algorithm is described in Figure 2.

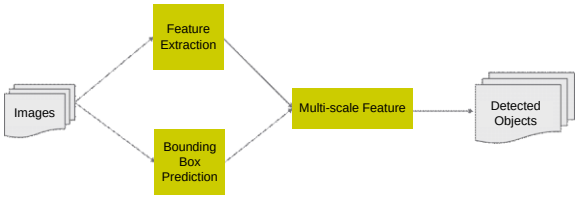

Figure 2. Overall YOLOv3 Algorithm.

When YOLOv3 receives an input image, the image travels through two simultaneous processes. One step is the feature extraction step, which identifies the class of the object present in the image. The other step is the bounding box prediction, which locates the position of the object in the image. Both the feature extraction and bounding box prediction processes are repeated three times, each time using different image processing parameters. Through this process known as multiscale prediction, the research obtains and compares different detections and then selects the best object detection.

A. *Prediction of Bounding Boxes Prediction and Cost Function Calculation*

The YOLOv3 system utilizes dimension clusters as anchor boxes to perform prediction of a number n of bounding boxes. The YOLO network forecasts the four coordinates of every bounding box t_x, t_y, t_w, t_h. The study obtains the following predictions *if* the bounding box prior has height and width p_h, p_w , with the cell showing an offset from the image's top-left corner by $(c_x,\ c_y)$ [19]:

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y \quad (1)$$
$$b_w = p_w e^{t_w}$$
$$b_h = p_h e^{t_h}$$

Figure 3 presents the bounding boxes with dimension priors and a location prediction, with the blue box as the bounding box prior and predicted bounding box. According to Joseph Redmon et al. [19], "If the ground truth for some coordinate prediction is $\hat{t} *$ our gradient is the ground truth value (computed from the ground truth box) minus our prediction: $\hat{t} * -t.$" Inverting the equations above may aid in calculating the value of the ground truth.

YOLOv3 applies a new means to compute the cost function, compared with the first YOLO algorithm. YOLOv3 predicts an objectness score of every bounding box on the base of logistic regression [19]. If a ground truth object is overlapped by a bounding box prior by more than any bounding box prior , then the objectness score should be 1. Following, the study disregards the prediction if the bounding box prior is not ideal, yet overlaps a ground truth object by a value above a particular threshold. In this case, a threshold of 0.5 is used. The YOLO system allocates a unique bounding box before every ground truth object unlike. No loss for class predictions or coordinate will happen should a bounding box prior not be allocated [19].

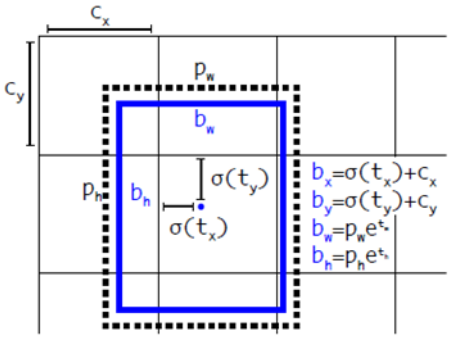

Figure 3. Bounding boxes with dimension priors and location prediction [19].

B. *Prediction of Classes*

Multilabel classifications were used by each box for the prediction of classes that the bounding box might contain. Such classification modeled the data better when dealing with complicated domains such as the Open Images data set [19], where many labels like woman and person overlapped. The original YOLO version normalized confidence scores into probabilities that add to one. This study accomplished normalization with the use of a softmax function. In YOLOv3 however, utilizing individualistic logistic classifiers rather than a softmax

tends to increase performance. Class predictions required the use of cross-entropy loss during training [19].

C. Multi-scale prediction

To increase the accuracy of features detection in an image, YOLO performed a multi-scale prediction. YOLOv3 then achieved a box prediction at three distinctive scales. From those scales, YOLOv3 extracted features that relate an inspired concept, uniquely drawn from feature pyramid networks [19]. The study added multiple convolution layers to the base feature extractor.

Figure 4 aids in visualizing the multiscale prediction performed by YOLOv3.

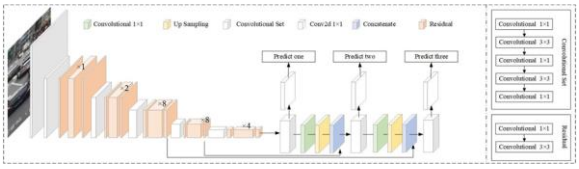

Figure 4. Structure detail of YOLOv3 using three scale predictions. [20].

The final convolutional layer predicted a 3-d tensor that determined class predictions, bounding box, and objectness. Referencing Redmon et al. [19], the study drew the prediction from "3 boxes at each scale, therefore the tensor was N ×N×[3*(4+1+80)] for the 4 bounding box offsets, 1 objectness prediction, and 80 class predictions."

Subsequently, the research took a feature map from two layers prior, so it could be up-sampled by a factor of 2. Concatenation was then used to fuse the up-sampled features with another feature map from an earlier stage of the network [19]. Using this method, the study obtained significant semantic data from both – with finer-grained data coming from the previous feature and the up-sampled features. By adding additional convolutional layers for a feature map combination, a similar tensor was predicted [19].

For the last scale, the research applied the same configuration for box prediction. As a result, the 3rd scale predictions displayed gain from the early fine grains features of the network, as well as the previous computation.

To determine the bounding box priors, the study applied k-means clustering. The study selected nine clusters and scales arbitrarily, and then segmented the clusters across the scales in an even manner [19]. More specifically, the nine clusters on the COCO dataset are (373 x 326), (156 x 198), (116 x 90), (59 x 119), (62 x 45), (30 x 61), (33 x 23), (16 x 30), (10 x 13).

D. Feature extractor

As its name suggests, a feature extractor uses convolutional operations to identify features from an image. In YOLOv2, the first upgrade on YOLO, the feature extractor used is Darknet-19. However, YOLOv3 replaces Darknet-19 with a new 53-layer Darknet-53. Similar to the residual network present in Resnet, Darknet-53 mainly contains 1 x 1 and 3 x 3 filters with skip connections. Darknet-53 achieved 2x faster with the same classification accuracy as the ResNet-152, although with less BFLOP (Billion Floating-Point Operations) [19]. Table 3 describes the structure of Darknet-53. In this table, the first line presents the characteristics of the first convolutional layer inside Darknet-53. This layer is made of 32 filters of size 3 x 3. Using skip connections, the residual block allowed one to jump from one layer to another to improve training. The "x" in front of the box represents the thickness dimension of a layer. Also, the global avg pool computes the average value of all values across the entire matrix to reduce computer calculations. Finally, fully connected layers are used for the actual image classification.

TABLE 3. DARKNET-53 STRUCTURE [19]

	Type	Filters	Size	Output
	Convolutional	32	3 x 3	256 x 256
	Convolutional	64	3 x 3 / 2	128 x 128
1x	Convolutional	32	1 x 1	
	Convolutional	62	3 x 3	
	Residual			128 x 128
	Convolutional	128	3 x 3 / 2	64 x 64
2x	Convolutional	64	1 x 1	
	Convolutional	128	3 x 3	
	Residual			64 x 64
	Convolutional	256	3 x 3 / 2	32 x 32
8x	Convolutional	128	1 x 1	
	Convolutional	256	3 x 3	
	Residual			32 x 32
	Convolutional	512	3 x 3 / 2	16 x 16
8x	Convolutional	32	1 x 1	
	Convolutional	62	3 x 3	
	Residual			16 x 16
	Convolutional	1024	3 x 3 / 2	8 x 8
4x	Convolutional	256	1 x 1	
	Convolutional	512	3 x 3	
	Residual			8 x 8

E. Training

In YOLOv3, the loss function to be trained is similar to the loss function in the original YOLO [21], except cross-entropy error terms replaced the three last squared error terms. The training was accomplished on full

images of the COCO dataset [22]. No hard, negative mining was used in the process. The training applied a multiscale process with different standard techniques, including batch normalization and data augmentation. Both training and testing required the use of the Darknet network [19].

F. Deep SORT

Once YOLOv3 detected the pedestrians, these were tracked and counted using Deep SORT. The overall Deep SORT algorithm is described in

Figure 5. When the Deep SORT is used, a deep appearance descriptor first must be trained. The descriptor initially extracted deep features for a re-identification task [23], with the use of convolutional layers (convs) and wide residual blocks (wrbs). These features were L2 and batch normalized to prevent overfitting and improve training. Once the object's position was determined via YOLOv3 detection, a Kalman filter estimated the state of the object using its position and velocity; this allowed the creation of tracks. After the new YOLOv3 bounding boxes were tracked by the Kalman filter, the next issue was to associate new predictions with new detections. The appearance descriptor produced normalized, deep features to calculate the min cosine distance between tracks and detection [23]. The Mahalanobis distance and the cosine distance obtained fusing dissimilarities for matching by utilizing a combination of the motional activities to enable better tracking. The study solved the association problem and applied a matching cascade to handle the limitations of the Kalman filter or the association metrics.

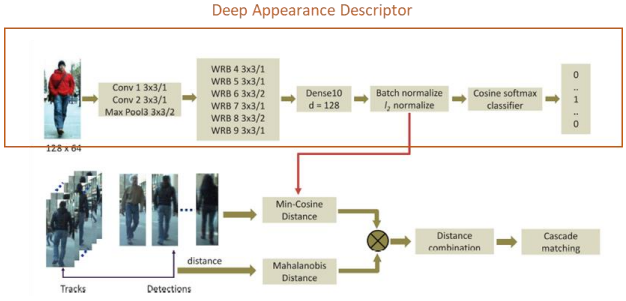

Figure 5. Deep Sort framework [23].

G. State estimation and creation of tracks using the Kalman filter

The YOLOv3 detection produced some noise to be filtered out. A Kalman filter via a state estimation handled both the noise filtering and the creation of tracks. Kalman filtering and track handling were performed similarly to the initial SORT algorithm [24]. The assumed tracking condition was that no ego-motion data was obtainable, and the camera, therefore, was not calibrated. Although this condition challenged the filtering framework, the most recent object tracking benchmarks considered this issue in their setup [24]. That being said, according to [18], "the tracking scenario is defined in the eight-dimensional state space $(u, v, \gamma, h. \dot{u}, \dot{v}, \dot{\gamma}, \dot{h})$ that contains the bounding box center position (u, v), aspect ratio γ, height h, and their respective velocities in image coordinates." As a result, the research chose to use a linear observation model, relying on a standard Kalman filter as well as the bounding coordinates $(u, v, \gamma, h. \dot{u}, \dot{v}, \dot{\gamma}, \dot{h})$.

For every track k, the study counted the number of frames since the previous successful association measurement. The counter's increment occurred at the same time as the Kalman filter prediction: When a track association occurred, the counter reset to 0. Should a track become more than the predefined maximum age A_{max}, it was deleted from the track set, since the deleted track was considered to have left the scene. For every detection that the algorithm could not associate with an old track, initiation of new track hypotheses occurred. During the initial three frames, the algorithm attempted to classify the newly created tracks. A good association measurement was expected at every time step during this period. In the same time interval, the deletion of unsuccessfully associated tracks occurred [18].

H. Solving the assignment problem using the Mahalanobis distance and the cosine distance

Building an assignment problem solved by the Hungarian algorithm is a common technique to resolve the association between newly arrived measurements and predicted Kalman states. Based on this method, the merging of two suitable metrics allows for the integration of both appearance and motion data. The study used the Mahalanobis distance between newly produced measurements and predicted Kalman states for motion information incorporation. The Mahalanobis distance is given by the equation:

$$d^{(1)}(i,j) = \left(d_j - y_i\right)^T S_i^{-1} (d_j - y_i) \qquad (2)$$

Where (y_i, S_i) denotes the projection of the i-th track distribution into the space of measurement and $d_j$denotes the j-th detection of the bounding box. To consider state estimation unreliability, the Mahalanobis distance measures the number of standard deviations that can be calculated when the detection is not close to the mean track location. Also, if the Mahalanobis metric is thresholded at a 95% confidence range calculated from the inverse χ^2 distribution, unlikely associations can be excluded [18].

The following indicator denotes this decision:

$$b_{i,j}^{(1)} = 1\,[d^{(1)}(i,j) \leq t^{(1)}] \qquad (3)$$

In the case that the association between j-th detection and i-th track is admissible, the value above evaluates to

1. The Mahalanobis threshold that corresponds to the four-dimensional measurement is $t^{(1)} = 9.4877$.
When there is a little uncertainty, the Mahalanobis distance is a suitable metric for the association. However, the predicted state distribution of the Kalman filter – regarding the formulation of the image-space problem – provides only a rough approximation of the location. More particularly, a fast-displacement in the image plane could be caused by an unpredictable camera movement. In this case, tracking through occlusions becomes difficult when the process employs the Mahalanobis metric.

To solve this issue, an additional metric is introduced. An appearance descriptor r_j is calculated for every bounding box detection d_j given $\|r_j\| = 1$. For every track k, a gallery $R_k = \left\{r_k^{(l)}\right\}_{k=1}^{L_k}$ of the previous $L_k = 100$ appearance descriptors are conserved. The new metric then calculates the smallest cosine distance between the j-th detection and i-th track in appearance space. The distance is described as follows:

$$d^{(2)}(i,j) = ming\left\{1 - r_j^T r_k^{(i)} \mid r_k^{(1)} \epsilon R_i\right\} \quad (4)$$

To indicate whether an association is admissible based on this metric, another binary variable is first introduced [18]:

$$b_{i,j}^{(2)} = 1\,[d^{(2)}(i,j) \le t^{(2)}] \quad (5)$$

Then on a separate training set, an appropriate threshold is found. Practically, computing bounding box appearance descriptors require the use of a pre-trained CNN [18]. The network architecture of the CNN will be explained later.

Since the Mahalanobis metric and the cosine metric serve various parts of the assignment problem, these metrics are complementary to each other in the association. On the one hand, the Mahalanobis provides particularly useful short-term data about possible object locations. On the other hand, the cosine distance determines appearance data is crucial for identities recovery after long-term occlusions where the motion becomes less distinctive. Utilizing a weighted sum, the merging of both metrics creates the association problem. The weighted sum is given by:

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1-\lambda) d^{(2)}(i,j) \quad (6)$$

If the association is within the gating region of both metrics, the association is called admissible[18]:

$$b_{i,j} = \prod_{m=1}^{2} b_{i,j}^{(m)} \quad (7)$$

Using the hyperparameter λ, it becomes possible to manage the effect of every metric on the merged association cost. If the camera is moving significantly, the setting λ = 0 is a feasible choice. This setting allows the use of only appearance data in the association cost term. Nevertheless, due to the fact that the Kalman filter locates objects, the Mahalanobis gate is still utilized.

I. Matching Cascade

After the study used the Mahalanobis distance and the cosine distance to associate new predictions with new predictions, a matching cascade was introduced to solve for measurement-to-track associations by resolving a series of subproblems. The cascade's input included the maximum age Amax, the set of track *T* indices, and the set of detection *D* indices. The study first computed the matrix of admissible associations, as well as the association cost matrix. Then the research solved the linear assignment issue for tracks of growing age by iteration over the trackage n. Next, the study selected a group of non-associated T_n tracks in the previous n frames which had not been associated, subsequently solving the linear assignment between unmatched detection $\mathcal{L}$ and tracks in T_n. Afterward, the group of unmatched and matched detections was updated. Priority was given to the most recently seen tracks, also called tracks of smaller age. The research ran an IOU (Intersection Over Union) association on the set of unmatched and unconfirmed tracks with the age $n = 1$. This IOU association was similar to the one suggested in the initial SORT algorithm and took into consideration any abrupt appearance modifications. Either robustness improvement against incorrect initialization or limited occlusion with fixed scene geometry caused such modifications. The matching algorithm is summarized as follows [18].

Input: Track Indices $T = \{1,,,,,N\}$, Detection Indices $D = \{1,\dots,M\}$, Maximum age A_{max}
1: Compute cost matrix $C = \left[c_{i,i}\right]$ using Eq 6
2: Compute cost matrix $B = \left[b_{i,i}\right]$ using Eq 7
3: Initialize set of matches $\mathcal{M} \leftarrow \emptyset$
4: Initialize set of unmatched detections $\mathcal{L} \leftarrow D$
5: **for** $n \in \{1,\dots,A_{max}\}$ do
6: Select tracks by age $T_n \leftarrow \{i \in T \mid a_i = n\}$
7: $\left[x_{i,j}\right] \leftarrow$ min cost matching $(C, T_n, \mathcal{L})$
8: $\mathcal{M} \leftarrow \mathcal{M} \cup \left\{(i,j) \mid \sum_i b_{i,i} \cdot x_{i,j} > 0\right\}$
9: $\mathcal{L} \leftarrow \mathcal{L} \setminus \left\{j \mid \sum_i b_{i,i} \cdot x_{i,j} > 0\right\}$
10: **end for**
11: **return** $\mathcal{M}, \mathcal{L}$

J. Deep Appearance Descriptor

For SORT with a deep association metric to be successful, a good outcome required training of an embedding feature offline before a performance of online tracking. The CNN used for this purpose trained on a large-scale person re-id set of data [25]. As the dataset is made of 1,100,000 images of 1,261 pedestrians, there was some confidence that the trained CNN would be appropriate for a deep metric aimed at counting people.

The CNN utilized represents a residual network [26] composed of six residual blocks and two convolutional layers. The computation of the global feature map with 128 of dimension occurred in the tenth layer. Features were projected on the unit hypersphere by a final batch and one normalization. There is a total of approximately 2.8 million parameters in the network. The architecture of the CNN is presented in Table 4.

TABLE 4. OVERVIEW OF NETWORK ARCHITECTURE [18].

Name	Patch Size/Stride	Output Sizes
Conv 1	3 x 3/1	32 x 128 x 64
Conv 2	3 x 3/1	32 x 128 x 64
Max Pool 3	3 x 3/2	32 x 64 x 32
Residual 4	3 x 3/1	32 x 64 x 32
Residual 5	3 x 3/1	32 x 64 x 32
Residual 6	3 x 3/2	64 x 32 x 16
Residual 7	3 x 3/1	64 x 32 x 16
Residual 8	3 x 3/2	128 x 16 x 8
Residual 9	3 x 3/1	128 x 16 x 8
Dense 10		128
Batch and L2 Normalization		128

5. SIMULATION AND IMPLEMENTATION

Using traffic cameras installed at busy street intersections in downtown New Orleans and near the Baton Rouge Community College (BRCC) in Baton Rouge, the Louisiana Transport Research Center was able to obtain hundreds of hours' worth of videos containing pedestrian activity that were considered for this study. An analysis of these videos showed that they differed according to 5 main characteristics including the location and time of recording, the direction and level of pedestrian traffic, the degree of clustering, and the degree of occlusion. Consequently, the study focused on determining how these characteristics may affect the accuracy of the counting algorithm. Table 5 summarizes all information about the videos.

For the experiment, the study implemented the YOLOv3 + SORT with a deep metric counter in python with the aid of the open-source Github repository, as created by Bobby Chen [26]. The study kept the original pre-trained parameters for both the YOLO detector (training on COCO dataset), as well as the cosine metric in SORT (training MARS data set). The original code was slightly modified to meet the project requirements; the study took care in selecting the "people" class only for detection. For cost limitation reasons, the study chose to process only 2 mins of footage among all videos available for each of the 4 types of videos. However the counting algorithm processed the videos at a reduced FPS compared with the original FPS of each video. Since the counter processed the videos at varying speed, the researcher added a timestamp to each video for accurate data collection. The study recorded both manual and automatic counts every 10 s for each of the four videos. Figure 1 shows a snapshot of a video before and after processing.

A. Video A

Figure 6 and

Figure 7 show snapshots of video A, before and after being processed by the automated pedestrian counter. Table 6 summarizes all the manual and automated counts obtained for video A and presents the relative errors between the manual and automated counts.

Referring to Figure 8, Figure 9, and Table 6, the results for video A show that 92 % of the time, the frames have a relative error of more than 50%, with 58% of the frames having a relative error of more than 100%. This means that the counting algorithm fails to correctly count pedestrians most of the time. After re-watching video A after processing, the study noted that the time frames with the highest relative error values (20-30, 50-60, 60-70) were the time frames where there was the most occlusion.

Figure 6. Video A before processing.

Figure 7. Video A after processing.

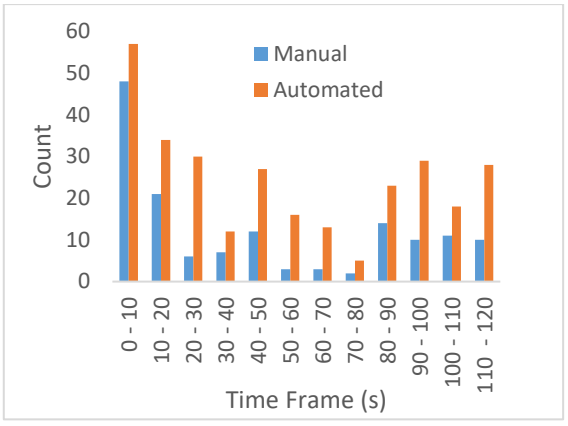

Figure 8. Manual vs Automated Counts Comparison for Video A.

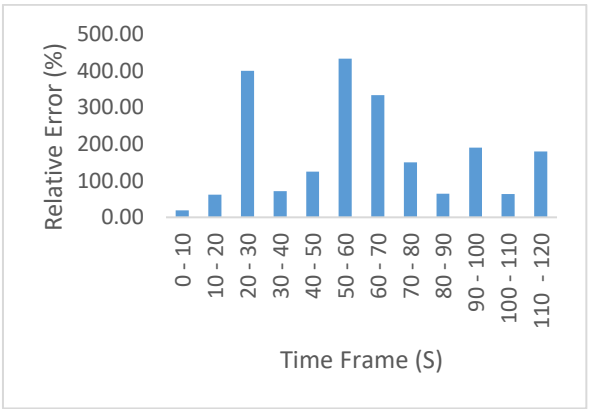

Figure 9. Relative error comparison for video A.

TABLE 5. VIDEO DATA DESCRIPTION.

Video #	Time	Bicyclists or Pedestrians	The direction of pedestrian Traffic	Pedestrian/Bicyclist Traffic	Presence of Clustered pedestrians	Amount of Occlusion
Video A	Morning	Pedestrians	All directions	High	High	High
Video B	Afternoon	Pedestrians	All directions	Low	medium	High
Video C	Night	Pedestrians	All directions	medium	High	Low
Video D	Morning	Cyclists	One direction	medium	medium	Medium

TABLE 6. AUTOMATED AND MANUAL COUNTS FOR VIDEO A AND RELATIVE ERROR

Time Frame (s)	Manual Count	Automated Count	Relative Error (%)
0 - 10	48	57	18.75
10 - 20	21	34	61.90
20 - 30	6	30	400.00
30 - 40	7	12	71.43
40 - 50	12	27	125.00
50 - 60	3	16	433.33
60 - 70	3	13	333.33
70 - 80	2	5	150.00
80 - 90	14	23	64.29
90 - 100	10	29	190.00
100 - 110	11	18	63.64
110 - 120	10	28	180.00

B. Video B

Figure 10 and

Figure 11 show snapshots of video B before and after being processed by the automated pedestrian counter. Table 7 summarizes all the manual and automated counts obtained for video B and presents the relative errors between the manual and automated counts.

The results for video B show that 58 % of the time frames have a relative error of more than 50%. This percentage shows that for video B, the algorithm is

somewhat accurate 50% of the time. After re-watching video B after processing, the study notes once more that the time frames with the highest relative error values (10-20, 50-60) are the ones where there was the most occlusion. However, although occlusion exists in both videos A and B, the algorithm appears to count video B

more accurately than video A. This could be due to less pedestrian traffic and less clustering in video B.

Figure 11 compares the manual and automated counts for video B, while

Figure 13 compares the relative errors on each time frame.

Figure 10. Video B before processing.

Figure 11. Video B after processing.

TABLE 7. AUTOMATED AND MANUAL COUNTS FOR VIDEO B AND RELATIVE ERROR.

Time Frame (s)	Manual Count	Automated Count	Relative Error (%)
0 - 10	16	19	18.75
10 - 20	3	10	233.33
20 - 30	6	15	150.00
30 - 40	8	6	25.00
40 - 50	15	32	113.33
50 - 60	7	16	128.57
60 - 70	13	17	30.77
70 - 80	2	25	1150.00
80 - 90	8	9	12.50
90 - 100	5	15	200.00
100 - 110	4	8	100.00
110 - 120	8	5	37.50

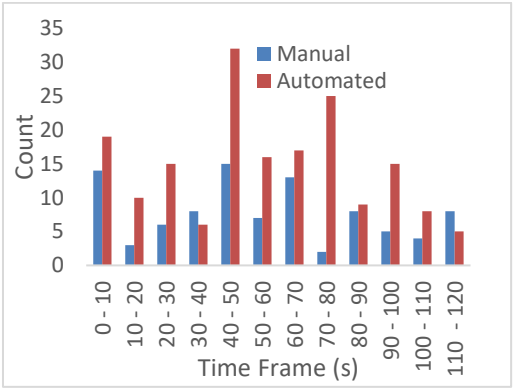

Figure 12. Manual vs Automated counts comparison For Video B.

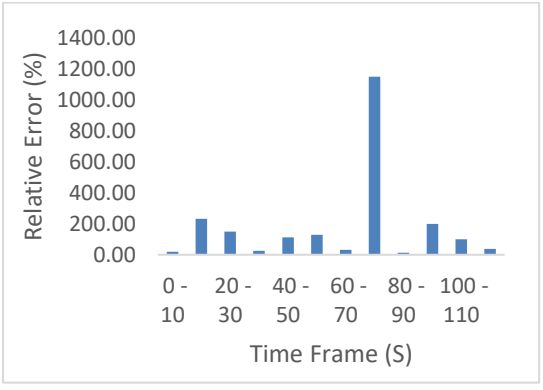

Figure 13. Comparison error for video B.

C. *Video C*

Figure 14 and

Figure 15 show snapshots of video C before and after being processed by the automated pedestrian counter. Table 8 summarizes all the manual and automated counts obtained for video C and presents relative errors between the manual and automated counts.

The results for video C show that 50 % of the time frames show a relative error of less than 30%, with 30% of the time frames having a relative error of 0%. This shows that the counting algorithm is far more accurate for video C than it is for videos A and B. This result is rational, as video C presents much less clustering and also shows low occlusion and pedestrian traffic.

Figure 16 compares the manual and automated counts for video B, while

Figure 17 compares the relative errors on each time frame.

Figure 14. Video C before processing.

Figure 15. Video C after processing.

TABLE 8. AUTOMATED AND MANUAL COUNTS FOR VIDEO C AND RELATIVE ERROR.

Time Frame (s)	Manual Count	Auto Count	Relative Error (%)
0 - 10	8	2	66.67
10 - 20	0	0	0.00
20 - 30	2	1	33.33
30 - 40	0	3	300.00
40 - 50	2	2	0.00
50 - 60	0	0	0.00
60 - 70	0	0	0.00
70 - 80	5	7	33.33
80 - 90	0	4	400.00
90 - 100	9	8	10.00
100 - 110	6	10	57.14
110 - 120	2	5	100.00

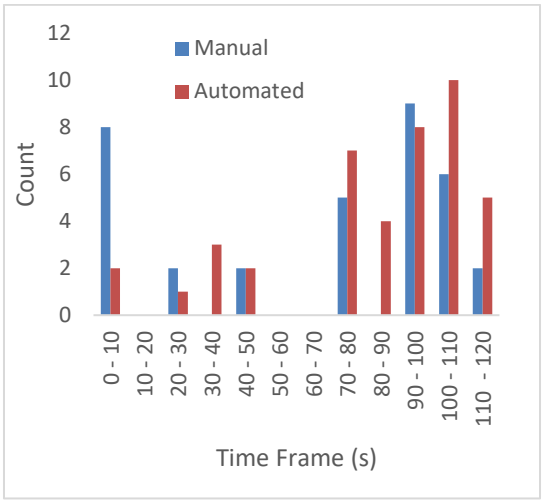

Figure 16. Manual vs Automated counts comparison For Video C.

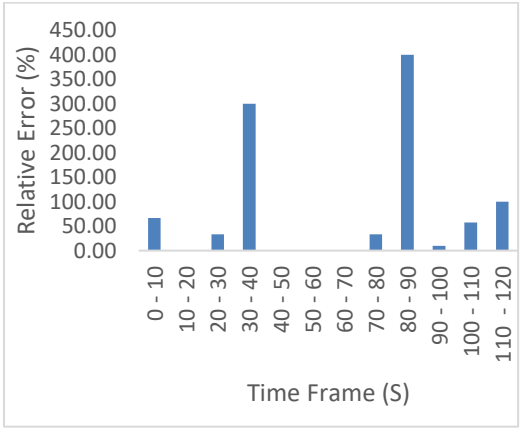

Figure 17. Relative error comparison for video C.

D. Video D

Figure 18 and

Figure 19 show snapshots of video D before and after being processed by the automated pedestrian counter. Table 9 summarizes all the manual and automated counts obtained for video D and presents the relative errors between the manual and automated counts.

The results for video D show that the counting algorithm counts both pedestrians and bicyclists should the right conditions be present. This is logical since the detector is initially trained on people in general, rather than specifically on pedestrians in the video. Of the time frames, 58 % have a relative error of less than 30%, with 25% of the time frames having a relative error of 0%. This shows the counting algorithm for video D to be even more accurate than for video C. This result could be because the traffic in video D travels only in one direction.

Figure 20 compares the manual and automated counts for video D, while

Figure 21 compares the relative errors on each time frame.

Figure 18.Video D before processing.

Figure 19. Video D after processing.

TABLE 9. AUTOMATED AND MANUAL COUNTS FOR VIDEO D AND RELATIVE ERROR.

Time Frame (s)	Manual Count	Automated Count	Relative error (%)
0 – 10	4	4	0.00
10 - 20	28	23	17.86
20 - 30	34	41	20.59
30 - 40	15	19	26.67
40 - 50	14	17	21.43
50 - 60	7	12	71.43
60 - 70	6	6	0.00
70 - 80	1	5	400.00
80 - 90	5	11	120.00
90 - 100	5	5	0.00
100 - 110	7	11	57.14
110 - 120	1	8	700.00

Figure 20. Manual vs Automated counts comparison for Video D.

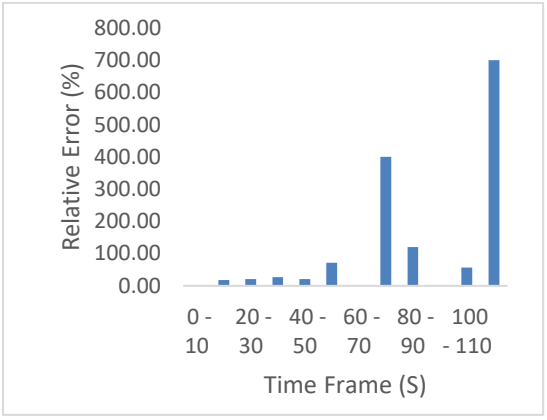

Figure 21. Relative errors comparison for video D

6. CONCLUSION AND FUTURE WORK

In this study, the pedestrian count is found to be important to three major criteria of CBD pedestrian planning; mobility, safety, and pleasure. The study discussed manual pedestrians counting and considering the different types of automated pedestrians counting, based on computer vision inclusive of clustering-based methods, region-based methods, and detection-based methods. For implementation, the study committed to working with a real-time, detection-based counter that applied YOLOv3 and Deep SORT. The study selected the counter due to its accuracy and simplicity in comparison with other benchmarks. The results obtained after testing the counter revealed that occlusion was the main source of error, while other factors, such as intensity and direction of pedestrian traffic, had less impact.

The relative errors for videos A and B showed to be very high. Since there were only two videos with multiple occlusions present, this study concludes that the occurring occlusions were solely responsible for the errors, regardless of the direction and amount of traffic. Due to occlusion, an object may be tracked more than once. Since some level of occlusion occurs in all videos, the study presents that occlusion constitutes an extant main. To solve the problem associated with occlusion, the Deep SORT tracker must be properly re-trained.

ACKNOWLEDGMENT

The authors would like to thank Electrical Engineering at Southern University and A&M College, Baton Rouge, USA for the great support provided in full to finalize this work. The authors also thank the Louisiana Transportation Research Center (LTRC) for providing real videos that were used to evaluate the YOLOv3 algorithm in this paper.

REFERENCES

[1] American Planning Association, [Online]. Available: www.planning.org/pas/reports/report199.htm. [Accessed: December 15, 2019].

[2] Jack P. Gibbs (ed.), Urban Research Methods (Princeton, N.J.: D. Van Nostrand Co., Inc, 1962), p. 197.

[3] For an exhaustive discussion of this use of the pedestrian count see Vincent J. Hubin, "Pedestrian Traffic Counts," The Appraisal Journal (July, 1953), pp. 397–415.

[4] Paul Wendt, The Dynamics of Central Land Values—San Francisco and Oakland, 1950 to 1960 (Berkeley, Calif.: Institute of Business and Economic Research, University of California, 1961), p. 31.

[5] Schneider, Robert J. "Methodology for Counting Pedestrians at Intersections." [Online] Available: Https://Safetrec.berkeley.edu/Sites/Default/Files/Publications/methodology_for_counting.Pdf. [Accessed: March 17, 2020].

[6] Diogenes, M. C., R. Greene-Roesel, L. S. Arnold, and D. R. Ragland. pedestrians counting Methods at Intersections: A Comparative Study. In Transportation Research Record: Journal of the Transportation Research Board, No. 2002, Transportation Research Board of the National Academies, Washington, D.C., 2007, pp. 26–30.

[7] Greene-Roesel, R., M. C. Diogenes, D. R. Ragland, and L. A. Lindau. Effectiveness of a Commercially Available Automated Pedestrian Counting Device in Urban Environments: Comparison with Manual Counts. Traffic Safety Center, University of California, Berkeley, 2008.

[8] Turner, S., D. Middleton, R. Longmire, M. Brewer, and R. Eurek. Testing and Evaluation of Pedestrian Sensors. Texas Transportation Institute, Texas A&M University, Sept. 2007.

[9] Exploring Pedestrian Count Procedures. (2016, May), [Online] Available: https://www.fhwa.dot.gov/policyinformation/travel_monitoring/pubs/hpl16026/hpl16026.pdf. [Accessed: February 17, 2020].

[13] kurilkin, Alexey V. "A Comparison of Methods to Detect People Flow Using Video Processing." [Online] Available: Https://Www.sciencedirect.com/Science/Article/Pii/S1877050916326837. [Accessed: March 17, 2020].

[14] G. J. Brostow and R. Cipolla, "Unsupervised bayesian detection of independent motion in crowds," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 2006, pp. 594–601.

[15] Y. Benabbas, N. Ihaddadene, T. Yahiaoui, T. Urruty, and C. Djeraba, "Spatio-temporal optical flow analysis for people counting," in Proceedings - IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2010, 2010, pp. 212–217.

[16] Aakhtar, Naveed. "Benchmark Data and Method for Real-Time People Counting in Cluttered Scenes Using Depth Sensors." [Online] Available: Https://Arxiv.org/Pdf/1804.04339.Pdf, [Accessed: April 8, 2020].

[17] G. Liu, Z. Yin, Y. Jia, and Y. Xie, "Passenger flow estimation based on convolutional neural network in public transportation system," Knowledge-Based Systems, vol. 123, pp. 102–115, 2017.

[18] X. Wei, J. Du, M. Liang, and L. Ye, "Boosting Deep Attribute Learning via Support Vector Regression for Fast Moving Crowd Counting," Pattern Recognition Letters, vol. 47, pp. 178–193, 2017. [Online] Available: http://linkinghub.elsevier.com/retrieve/pii/S0167865517304415. [Accessed: April 8, 2020].

[19] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement, [Online] Available: https://pjreddie.com/media/files/papers/YOLOv3.pdf. [Accessed: September 20, 2019].

[20] Wojke, N., Bewley, A., & Paulus, D., "Simple Online and Realtime Tracking with a Deep Association Metric", [Online] Available: https://arxiv.org/pdf/1703.07402.pdf. [Accessed: January 5, 2020].

[19] I. Krasin, et al., "A public dataset for large-scale multi-label and multi-class image classification", [Online] Available: https://github.com/openimages, [Accessed: January 5, 2019].

[20] Mao, Q.-C., "Mini-YOLOv3: Real-Time Object Detector for Embedded Applications", [Online] Available: https://www.researchgate.net/publication/335865923_Mini-YOLOv3_Real-Time_Object_Detector_for_Embedded_Applications. [Accessed: March 20, 2020].

[21] Redmon, Joseph. "You Only Look Once: Unified, Real-Time Object Detection." Retrieved December 15, 2019 from Https://Arxiv.org/Pdf/1506.02640.Pdf, 9 May 2016.

[22] Common Objects In Context. [Online] Available: http://cocodataset.org/#download. [Accessed: March 20, 2020].

[23] Chen, S., & Xu, Y. (2019, January). Deep Learning for Multiple Object Tracking: A Survey. [Online] Available: https://www.researchgate.net/publication/330220368_Deep_Learning_for_Multiple_Object_Tracking_A_Survey. [Accessed: April 26, 2020].

[24] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and ´ K. Schindler, "Mot16: A benchmark for multi-object tracking," arXiv preprint arXiv:1603.00831, 2016.

[25] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, "MARS: A video benchmark for large-scale person re-identification," in ECCV, 2016. BMVC

[26] Chen, B. (n.d.). Object detection and tracking. [Online] Available: https://github.com/yehengchen/Object-Detection-and-Tracking. [Accessed: April 24, 2020].

Mr. Willson Junior Meli Ngong received his B.Sc. degree in Biomedical Engineering from Louisiana Tech University – Ruston, LA, in 2017. As an undergraduate student, he was involved with the brain dynamics lab of the Center for Biomedical Engineering and Rehabilitation Science - Ruston, LA. His work with the lab consisted of processing intercranial EEG recordings in order to localize epileptogenic focus. In 2018 he joined Southern University and A&M to pursue a master of engineering with a concentration in materials science and engineering and a focus in electronic materials and processing. As a master student, his research interest was video processing-based automated pedestrian counting systems.

Dr. Fred Lacy received the B.S.E.E. degree and Ph.D. degree in electrical engineering from Howard University, Washington, DC, in 1987 and 1993, respectively, and the M.S.E. degree from Johns Hopkins University, Baltimore, MD, in 1989. He was a Postdoctoral Fellow in the Bioengineering Department, University of California, San Diego, for four years, where he performed research in the area of biosensors. He was with the US Food and Drug Administration, where he performed medical device reviews. In 2002, he joined the Electrical Engineering Department, Southern University and A&M College, Baton Rouge, LA, where he is engaged in research and teaches courses in solid-state electronics, electrical and electronic circuits, and electronics-based sensors.

Dr. Yasser Ismail received his BS. degree in Electronics & Communications Engineering from Mansoura University - Egypt, in 1999. He received his MS in Electrical Communications from Mansoura University - Egypt, in 2002. Dr. Ismail received his MSc and PhD degrees in Computer Engineering from University of Louisiana at Lafayette - USA in 2007 and 2010 and subsequently joined Umm Al-Qura University - Kingdom of Saudi Arabia as an assistant professor. In Fall 2012, he joined University of Bahrain - Kingdom of Bahrain as a Computer Engineering Assistant Professor. In Fall 2016, Dr. Ismail Joined both Electronics & Communications Engineering Department - Mansoura University - Egypt and Zewail City of Science and Technology - University of Science and Technology - Zewail City - Egypt as an assistant professor. Dr. Ismail is currently working as an assistant professor in the Electrical Engineering Department, Southern University and A&M College - Baton Rouge - Louisiana - USA. His area of expertise is Digital Video Processing Algorithms/Architectures levels, Internet of Things (IoT), VLSI and FPGA Design (Low-Power and High-Speed Performance Embedded Systems), automotive transportation, Robotics, RFID, and Wireless and Digital Communication Systems. He has published two books, two book chapters, and more than 35 articles in related journals and conferences. Dr. Ismail served as a reviewer for several conferences and journals, including IEEE ICIP, IEEE GCCCE, IEEE ICECS, IEEE MWSCAS, IEEE ISCAS, IEEE SIPS, IJCDS, Springer, Elsevier, IEEE Transactions on VLSI, IEEE Transaction on Circuit and System for Video Technology (TCSVT), and IEEE Transactions on Image Processing. He severed in the technical committees of IEEE ISCAS 2007, IEEE ICECS 2013, MobiApps 2016, IEEE Virtual World Forum on Internet of Things (WF-IoT 2020), and IEEE MWSCAS 2018, 2019, and 2020 conferences. He was invited to serve as a lead guest editor for a special issue in mobile information systems – Hindawi publishing corporation September 2016. Dr. Ismail served as a PI and Co-PI for several funded grants from NSF and other international fund agencies. Additionally, Dr. Ismail served as a member of many Editorial Boards 2018-present.