NUANSA INFORMATIKA
Volume 20 Nomor 1 January 2026

p-ISSN :1858-3911 **,** ***e-ISSN : 2614-5405***
https://journal.fkom.uniku.ac.id/ilkom

High Performance Yolov4 with Different Optimizers and Backbones to Determine Distance

Irma Amelia Dewi[1]*, Mikael Prapaskalis G[2], Muhammad Ichwan[3]
[1,2,3]Department of Informatics, Institut Teknologi Nasional Bandung, Indonesia

E-mail: *[1]**irma_amelia@itenas.ac.id**,[2]paskalis97@gmail.com,[3]ichwan@itenas.ac.id

Abstract

Monitoring the distance between individuals in public spaces remains relevant for crowd management, operational safety, and risk mitigation in public service. Manual inspections are difficult to carry out consistently and in real time, thus requiring an automated machine vision-based system. This research proposes a safe distance monitoring pipeline that detects humans, calculates the proximity between individuals. Human detection is performed with YOLOv4 and the CSPResNeXt50 backbone, then the center of the bounding box is used as a position representation. The distance between centroids is calculated using Euclidean Distance and calibrated based on camera parameters, thus that it can be mapped to distance units. To obtain a stable training configuration, two optimizers (Adam and SGD) are compared at two learning rates (1E-2 and 1E-3). Test results show that Adam with a learning rate of 1E-3 provides the best performance, achieving 96% object detection accuracy. The distance calculation module achieves 90% accuracy in determining the actual and predicted distances in optimal camera installation scenarios, with an average processing speed of 60 fps. These findings demonstrate a fast and practical approach for camera-based public space safety. Additionally, the backbone comparison shows that CSPResNeXt50 is slightly superior to DarkNet50, making it suitable for lightweight implementation on edge devices in real-time.

Keywords*—object detection, Yolov4, CSPResNeXt50, deep learning*

Submission: November 25, 2025 Accepted: December 23, 2025 Published: January 20, 2026

DOI Article: https://doi.org/ 10.25134/ilkom.v20i1.533

1. INTRODUCTION

The density and proximity of individuals in public spaces (stations/terminals, shopping centers, public service areas, building entrances, campus corridors, and industrial areas) are important indicators for crowd management, operational safety, and compliance with Standard Operating Procedures (e.g., queue management, capacity restrictions, and prevention of mass gatherings) [1], [2], [3]. Manual monitoring by officers is not easy to carry out consistently due to limited visibility, fatigue, and the need for quick responses, so an automated system is needed that can work in real-time and be easily integrated with existing surveillance cameras.

Rapid developments in computer vision and artificial intelligence for crowd analytics, such as people detection, people counting/crowd density, and estimation of proximity between individuals from CCTV video. Crowd analytics research itself remains active and relevant to the context of safety and urban space management, including the need for efficient computing, particularly when directed towards edge/near-edge implementation [4]. The core component of such systems typically starts with human detection. The single-stage detector family is popular because of its favorable trade-off between speed and accuracy for real-time applications, and continues to improve in terms of architecture and training strategies, such as YOLOX (anchor-free, decoupled head) [5], YOLOv7

p-ISSN :1858-3911 , *e-ISSN : 2614-5405*
https://journal.fkom.uniku.ac.id/ilkom

(trainable bag-of-freebies) [6], and RT-DETR [7], which leads to real-time end-to-end detection. This trend shows that the choice of architecture and training configuration (optimizer, learning rate, backbone) greatly determines the performance of the system in real-time scenarios.

On the other hand, estimating the actual distance from a single camera (monocular) is a major challenge due to the perspective effect, where the distance in pixels is not linear to the actual distance. Therefore, many approaches utilize calibration/homography (to map the floor area), inverse perspective mapping, or pose/depth-based strategies to reduce distance error in various CCTV camera conditions [8]. Other common challenges include occlusion, viewing angle variations, and differences in camera mounting heights, which could shift estimation accuracy. Based on these requirements, this study proposes a proximity monitoring pipeline between individuals based on human detection and simple yet fast distance calculation.

Therefore, the research was conducted to obtain a system that is able to detect the distance between humans that is considered according to the rules of physical distancing using Object Detection. Object detection is a technology that is able to detect an object of a particular class in an image or video. Object detection is capable of detecting human faces, detecting objects, detecting pedestrians (humans), vehicles, and so on [9]. The system uses You Only Look Once (YOLO).

YOLO is the best real-time object detection algorithm because YOLO uses a single convolutional neural network that simultaneously predicts several bounding boxes and classes of the entire image in a single detection or in one run [10]. Although newer YOLO variants such as YOLOv7 are reported to improve accuracy in the real-time range, this study chose YOLOv4 [11] because (i) the focus of the contribution is on the design of a CCTV-based proximity monitoring system and distance estimation evaluation, not on optimizing state-of-the-art detectors; (ii) YOLOv4 is designed for speed-accuracy trade-offs and is reported to be capable of achieving real-time performance in certain configurations , and (iii) the stability and maturity of the YOLOv4 implementation/deployment toolchain supports experiment reproducibility and simpler real-time integration compared to end-to-end export pipelines in newer models [11], [12].

Based on these requirements, this study proposes a pipeline for monitoring proximity between individuals that relies on human detection and simple yet fast distance calculations. The system uses YOLOv4 with a CSPResNeXt50 backbone. This backbone was chosen based on the Cross Stage Partial Network (CSPNet) concept, which can improve CNN learning capabilities by combining CSP with several architectures, such as ResNet, ResNeXt, and DenseNet. Among these combinations, ResNeXt reportedly provides better results, giving rise to the CSPResNeXt variant, which can reduce computation time by about 20% while maintaining or even improving detection accuracy [13], [14].

In addition, this study compares two optimizers (Adam and SGD) and two learning rate values to obtain the most stable training configuration. The distance between individuals is calculated from the representative position (centroid bounding box) using Euclidean Distance, then tested in several scenarios of camera height and angle to determine the most optimal installation conditions.

In summary, the emphasized contributions are: (1) the design of a lightweight, real-time CCTV-based proximity monitoring system; (2) analysis of the influence of optimizers and learning rates on detection performance; and (3) evaluation of distance calculation accuracy under varying camera installation conditions to support practical implementation in the field.

2. RESEARCH METHOD

The proposed algorithm performs preprocessing on the image. The image will be resized according to the insert method.

p-ISSN :1858-3911 , e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

Next, the process will continue on data augmentation.

The feature extraction process is carried out using CSPResNeXt50. In CSPResNeXt50 the convulsion filter will be divided into two parts in order to reduce computation time and reduce memory usage, for example, when convulsing with filter 128, the convulsion filter will be divided into two so that it becomes convulsive filter 64 with a size of 3x3, when it has passed the convulsion process it will enters the transition layer where the output of the convulsion is combined with the half of the input that has been divided previously to prevent duplication. This process is followed by ReLU activation and Maxpooling. If there is a low confidence value in the bounding boxes, then the bonding boxes are removed using non-max suppression. The flow process can be seen in Figure 1.

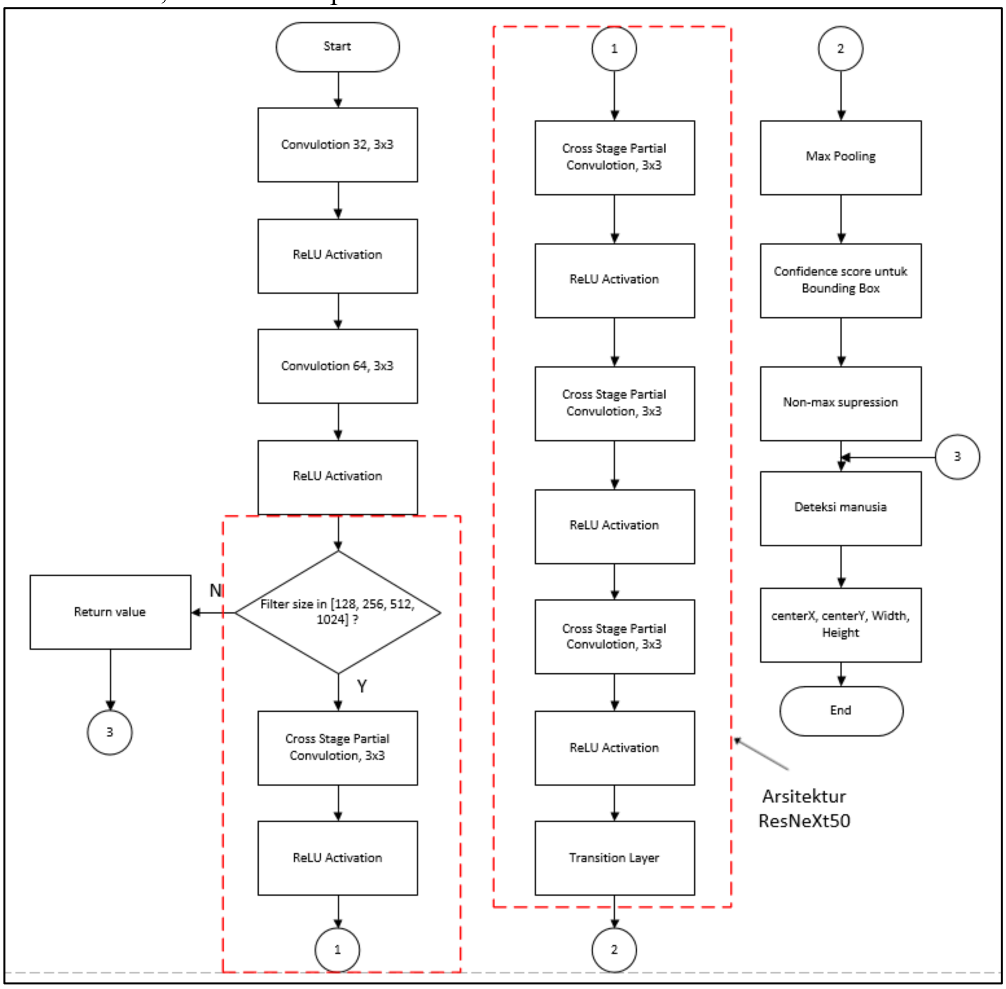

Figure 1. CSPResNeXt50

After feature extraction is performed on the input image, the extracted features will then be collected for detection and classification. This process occurs at the neck. Neck is the next process of object detection where the neck is in charge of taking and combining the features that have been extracted during the feature extraction process on the backbone which is then carried out in the object detection process. In Yolov4 using PANet in the process of merging the extraction features.

After successfully detecting an object in the form of a human, the next step is to determine the centroid point of the detected object. To determine the centroid of the detected object, you can use Equation (1).

$$p = \left(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}\right) \qquad (1)$$

p-ISSN :1858-3911 , e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

The centroid point will be obtained from the results of the determination using equation (1) which is based on processing xmin, xmax, ymin, ymax to produce x and y values which determine the value of the centroid point as shown in Figure 2.

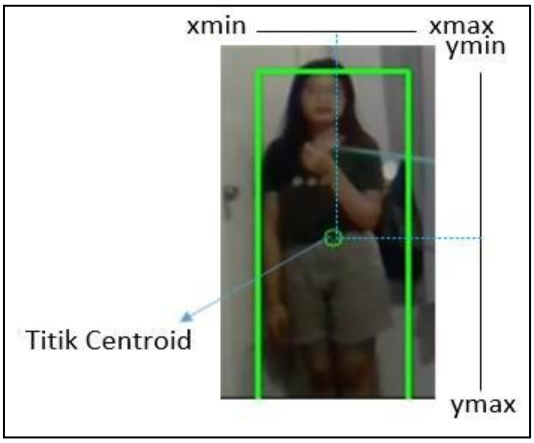

Figure 2. The Centroid Point

After obtaining the centroid point value of each object that has been detected, the process of calculating the distance between objects detected will be carried out using the Euclidean distance. Euclidean distance is the distance between two points in Euclidean space. Euclidean Distance is often used to calculate the distance between two different points. In this study, the distance between 2 objects was measured using Euclidean Distance and based on the object's centroid point that had been detected. After obtaining the centroid value on the detection object as in the above process, a process is carried out to calculate the distance between objects based on the predetermined centroid point [15]. This method is used to calculate the distance between objects detected by Equation (2).

$$d(p,q) = \sqrt{(p1 - q1)^2 + (p2 - q2)^2} \quad (2)$$

2.1 Distance Calibration

The calibration method for converting pixel distances to physical distances is critical because pixel measurements from camera images do not directly correspond to real-world distances due to perspective projection, camera mounting height, and viewing angle [16].

The calibration process involves steps starting from reference object placement, which two reference markers are placed at a known physical distance (Dref = 200 cm) on the ground plane within the camera's field of view. Pixel distance measurement for capturing an image and measures the pixel distance (dref_pixel) between the two reference markers using the same centroid-based approach as human detection.

After that, the pixel to centimeter scale factor (α) is calculated as: α = Dref / dref_pixel (cm/pixel). Converting the physical distance for each detected human pair to the pixel distance Dpixel (from Equation 2), the physical distance is: Dphysical = α × Dpixel (cm).

3. RESULT AND DISCUSSION

This section describes the results of testing the physical distancing system after going through the preprocessing process, extracting features using CSPResNeXt50, determining the centroid point and calculating the distance using Euclidean distance. The results of system testing can be seen in Figure 3.

Figure 3. The Result of Object Detection and Define a Centroid Point between Two Objects

The performance evaluation of the method is applied by measuring the optimizer and learning rate, i.e. Adam and SGD with learning rate 1E-2 and 1E-3. The following are the measurements results of Yolov4 method using different optimizer dan learning rate. Table 1 shows the effect of using different optimizer dan learning rate

p-ISSN :1858-3911 , *e-ISSN : 2614-5405*
https://journal.fkom.uniku.ac.id/ilkom

that can impact the result of accuracy. The Adam optimizer hyperparameter and the learning rate of 1e-3 significantly influence the production of a more optimal model.

Based on the results of all the training processes that have been carried out, the Yolov4 model and the ResNeXt50 architecture use the Adam optimizer with a learning rate of 1E-3 and use 150 epochs of the highest accuracy with an accuracy quality of 96%. Furthermore, Table 2 shows the performance of the Yolov4 method using DarkNet50 architecture. The parameter is the same as when the Yolov4 method using ResNeXt50 architecture. The parameter influences the performance values and the better the performance value will be obtained. Model training was also carried out using the Yolov4 model with DarkNet50 architecture using the Adam optimizer with Learning rate 1E-3, epoch 100, and batch size 64. Training using the DarkNet architecture resulted in an accuracy of 94%.

Table 1. Measurement of Yolov4 Method with ResNeXt50

Backbone	Optimizer	Learning Rate	*Epoch*	Acc
ResNeXt50	Adam	1E-2	100	94.70%
ResNeXt50	Adam	1E-3	100	95.10%
ResNeXt50	Adam	1E-3	150	96%
ResNeXt50	SGD	1E-2	100	80.55%
ResNeXt50	SGD	1E-3	100	88.15%

Table 2. Measurement of Yolov4 Method with DarkNet50

Backbone	Optimizer	Learning Rate	*Epoch*	Acc
ResNeXt50	Adam	1E-3	100	95.10%
ResNeXt50	Adam	1E-3	150	96%
DarkNet50	Adam	1E-3	100	94%
ResNeXt50	SGD	1E-3	100	88%

Based on the results of the model training that has been carried out using two architectures, namely ResNeXt50 and DarkNet50. Where the ResNeXt50 architecture gives an accuracy result of 95.10% using the Adam optimizer, learning rate 1E-3, epoch 100, and batch size 64 and the ResNeXt50 architecture gives 94% accuracy results using the Adam optimizer, learning rate 1E-3, epoch 100, and batch size 64. ResNeXt50 architecture gives 1% better or higher results than DarkNet50 architecture. A comparison of the two architectures used can be seen in Table 2.

From the results of the tests carried out by the Yolov4 model using the ResNeXt50 architecture, it gives better results and when training on the Yolov4 model with the ResNeXt50 architecture by adding 50 epochs it gives an accuracy of 96% which has an increase of 1% when using the optimizer and learning rate. which is equal to epoch 100.

Testing the Yolov4 model with the ResNeXt50 architecture is carried out by testing the model that has been trained by performing object detection. Where the Yolov4 model with ResNeXt50 architecture is distinguished based on the optimizer used when training the optimizer, namely Adam and SGD. The optimizer used in model testing is the optimizer that produces the highest accuracy value.

Table 3. The Evaluation of Yolov4 Model

Backbone	Optimizer	Learning Rate	Precision	Recall	F-Score
ResNeXt50	Adam	1E-3	96%	94%	95%
ResNext50	SGD	1E-3	89%	87%	87%

The results of the model testing carried out show that the Yolov4 model with the ResNeXt50 backbone, in Table 3 using the Adam optimizer with Learning Rate 1E-3 gives better results when compared to using the SGD optimizer and is the best weight or best fit used in the detection object of the Yolov4 model. Based on the system performance testing that has been done to find the highest accuracy based on the Euclidean distance used where the test was carried out 20 times. Model testing is done using the Yolov4 model and the CSPResNeXt50 backbone with the Adam optimizer giving good results, this is done by using a camera height of 2 meters with a camera angle of 90^{o}.

Table 4. Euclidean Distance Accuracy

Method	Distance Object to Camera	Distance Between Objects	Acc	FPS

Euclidean	300 cm	200 cm	90%	60
Euclidean	200 cm	200 cm	80%	60

From the test results as shown in the Table 4, it shows that The study has 90% accuracy in determining the actual and predicted distances at a camera distance of 300 cm and 80% at a distance of 200 cm based on 20 tests with the camera at a height of 2 meters and a 90° angle. The optimal distance when detecting using the Euclidean Distance Method is at a distance of 300 cm with a higher accuracy of 10% compared to the distance of the object to the camera is 200 cm. The detection results also provide more optimal results when object detection is carried out at the most optimal distance between the object and the camera where the optimal distance between the object and the camera is 3 meters.

4. CONCLUSSION

The implementation of the system using the YOLOv4 model with the ResNeXt50 is able to detect objects in the form of humans and is able to compare the distance between human objects and the distance between physical distancing rules and determine whether the detected object violates the physical distancing rules or not. The distance calculation is done using Euclidean Distance which gives optimal results with an accuracy of 90% concordance between the actual distance and the distance predicted by the system and gets an average detection speed of 60 fps.

The YOLOv4 Model using the ResNeXt50 backbone using 2 optimizers with 2 different learning rates, it shows different results even though using the same batch size and epoch where the Adam optimizer provides 95.10% accuracy and the SGD optimizer provides 88.15% accuracy. Where the Adam optimizer shows better and more stable results than the results given by the SGD optimizer. Where the YOLOv4 model using Adam's ResNeXt50 optimizer backbone using a learning rate of 1E-3 with a batch size of 64 with 150 epochs shows better results with an accuracy of 96%. From the comparison results between the two architectures, namely ResNeXt50 and DarkNet50, it is found that the ResNeXt50 architecture gives 1% better results with 95.10% accuracy than the DarkNet50 architecture which gives 94% results using the Adam optimier, epoch 100, and batch size 64.

REFERENSI

[1] I. Ahmed, M. Ahmad, J. J. P. C. Rodrigues, G. Jeon, and S. Din, “A deep learning-based social distance monitoring framework for COVID-19,” *Sustain. Cities Soc.*, vol. 65, no. 102571, pp. 1–12, 2021, doi: 10.1016/j.scs.2020.102571. Available: https://doi.org/10.1016/j.scs.2020.102571

[2] L. Gogulamudi, S. Mudigonda, Y. P. Renduchintala, B. Vemula, and G. H. Palnati, “A Social Distance Estimation and Crowd Monitoring System for Surveillance Cameras,” *Res. Adv. Innov. Comput. Commun. Inf. Technol. ICRAIC2IT*, vol. 2796, no. 1, pp. 1–21, 2022, doi: 10.1063/5.0148967

[3] M. Seker, A. Männistö, A. Iosifidis, and J. Raitoharju, “Automatic social distance estimation for photographic studies : Performance evaluation , test benchmark , and algorithm ☆,” *Mach. Learn. with Appl.*, vol. 10, no. October, pp. 1–14, 2022, doi: 10.1016/j.mlwa.2022.100427. Available: https://doi.org/10.1016/j.mlwa.2022.100427

[4] L. Deng, Q. Zhou, S. Wang, J. M. Górriz, and Y. Zhang, “Deep learning in crowd counting: A survey,” *CAAI Trans. Intell. Technol.*, vol. 9, no. 5, pp. 1043–1077, 2024, doi: 10.1049/cit2.12241

[5] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO Series in 2021,” pp. 1–7, 2021, Available: http://arxiv.org/abs/2107.08430

[6] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7,” *Cvpr*, pp. 7464–7475, 2023.

[7] Y. Zhao *et al.*, “DETRs Beat YOLOs

p-ISSN :1858-3911* , *e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

on Real-time Object Detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 16965–16974, 2024, doi: 10.1109/CVPR52733.2024.01605

[8] M. Aghaei, M. Bustreo, Y. Wang, G. Bailo, P. Morerio, and A. Del Bue, "Single image human proxemics estimation for visual social distancing," *Proc. - 2021 IEEE Winter Conf. Appl. Comput. Vision, WACV 2021*, pp. 2784–2794, 2021, doi: 10.1109/WACV48630.2021.00283

[9] V. Sharma and R. N. Mir, "A comprehensive and systematic look up into deep learning based object detection techniques: A review," *Comput. Sci. Rev.*, vol. 38, p. 100301, 2020, doi: 10.1016/j.cosrev.2020.100301

[10] Y. Jamtsho, P. Riyamongkol, and R. Waranusast, "Real-time license plate detection for non-helmeted motorcyclist using YOLO," *ICT Express*, no. xxxx, 2020, doi: 10.1016/j.icte.2020.07.008

[11] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv*, 2020.

[12] D. Wu, S. Lv, M. Jiang, and H. Song, "Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments," *Comput. Electron. Agric.*, vol. 178, no. July, p. 105742, 2020, doi: 10.1016/j.compag.2020.105742

[13] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2020-June, pp. 1571–1580, 2020, doi: 10.1109/CVPRW50498.2020.00203

[14] M. Mahasin and I. A. Dewi, "Comparison of CSPDarkNet53 , CSPResNeXt-50 , and EfficientNet-B0 Backbones on YOLO V4 as Object Detector," *Int. J. Eng. Sci. Inf. Technol.*, vol. 2, no. 3, pp. 64–72, 2022, doi: 10.52088/ijesty.v2i3.291

[15] R. Rizaldi, A. Kurniawati, and C. V. Angkoso, "Implementasi Metode Euclidean Distance untuk Rekomendasi Ukuran Pakaian pada Aplikasi Ruang Ganti Virtual," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 2, p. 129, 2018, doi: 10.25126/jtiik.201852592

[16] M. Nieto and N. Aginako, "Multi-camera BEV video-surveillance system," *Multimed. Tools Appl.*, vol. 82, pp. 34995–35019, 2023, doi: 10.1007/s11042-023-14416-y