



# The intelligent decision model for determine the best path of transportation on smart city using random forest algorithm and bayesian optimization (RF-BO)

Ahmad Fali Oklilas<sup>1,\*</sup>, Milda Kamilia<sup>2</sup>, Abdurahman<sup>3</sup>, Bitra Parga Zen<sup>4</sup>, Ari Widodo<sup>5</sup>

<sup>1,2,3</sup>Departement of Computer System, Sriwijaya University

<sup>4</sup>Department of Informatics, Institut Teknologi Telkom Purwokerto

<sup>5</sup>Department of Aquaculture, National Taiwan Ocean University

<sup>1,2,3</sup>Jl. Raya Palembang - Prabumulih, KM. 32, Ogan Ilir 30662, Indonesia

<sup>4</sup>Jl. D. I. Panjaitan, No. 128, Purwokerto 53147, Indonesia

<sup>5</sup>No. 2, Beining Rd., Keelung City 202301, Taiwan (R.O.C.)

\*Corresponding email: [fali@ilkom.unsri.ac.id](mailto:fali@ilkom.unsri.ac.id)

Received 4 September 2023, Revised 3 November 2023, Accepted 22 November 2023

**Abstract** — This study investigated various approaches and algorithms in the context of object detection and best path determination for managing vehicular traffic in an urban environment, particularly in Palembang City. This research was a step towards the development of the smart city concept. In the object detection analysis, we applied the YOLOv3 method on video footage to identify vehicles, resulting in mAP accuracy rates between 72.7 % to 79.3 % for both motorcycle and car categories. The total detection accuracy of the model reached 76.0 %. Next, we adopted the Random Forest algorithm to classify traffic conditions into three classes: smooth, moderate, and congested. After optimizing the algorithm with Bayesian optimization, the model accuracy increased from 89 % to 92 %, while the classification accuracy increased from 91.6 % to 92.3 %. When the model fits the training data too well, leading to a decline in performance on test data, overfitting could be prevented by carefully optimizing the parameters using Bayesian optimization. The A\* algorithm would then process the classification results from Random Forest optimization by Bayesian optimization using the Heuristic Search method, considering road conditions and travel distance. This would reveal that path 5, which runs from SMK PGRI 1 Palembang to Bom Baru Jl Perintis Kemerdekaan Arah Charitas (STMIK MBC), was most frequently selected nine times out of twelve trials. This approach was chosen because it had the lowest travel distance compared to other options and tended to have "smooth" traffic levels. Choosing the optimal path also considered the road width factor, where wider roads could reduce traffic density and the risk of congestion.

**Keywords** – A\* heuristic search, best path determination, bayesian optimization, random forest algorithm, YOLOv3

Copyright ©2023 JURNAL INFOTEL  
All rights reserved.

## I. INTRODUCTION

Traffic congestion is one of the most common problems faced in daily routines, hampering mobility and often causing delays in reaching destinations. The ever-increasing population growth rate significantly exacerbates the situation, leading to higher congestion on roads and more motorists. In the face of congestion, drivers often look for alternative routes with the shortest distance to travel to waste as little time as possible. However, the lack of information about the state or condition of the road on the route to be passed can also hinder the search for alternative routes that can be used as the best route to get to the destination if

the main route has obstacles [1].

The use of AI technology continues to grow rapidly, especially in the field of computer vision. Computer vision refers to technology that is capable of making intelligent decisions based on analyzing objects in the real world through visual imagery. It serves to replicate human visual capabilities so that the system can "see" objects around it and convert that information into meaningful decisions. Intelligent Transportation System (ITS) is one of the implementations of advanced technology in the fields of electronics, computers, and telecommunications used to improve the overall efficiency and effectiveness of transportation [2].

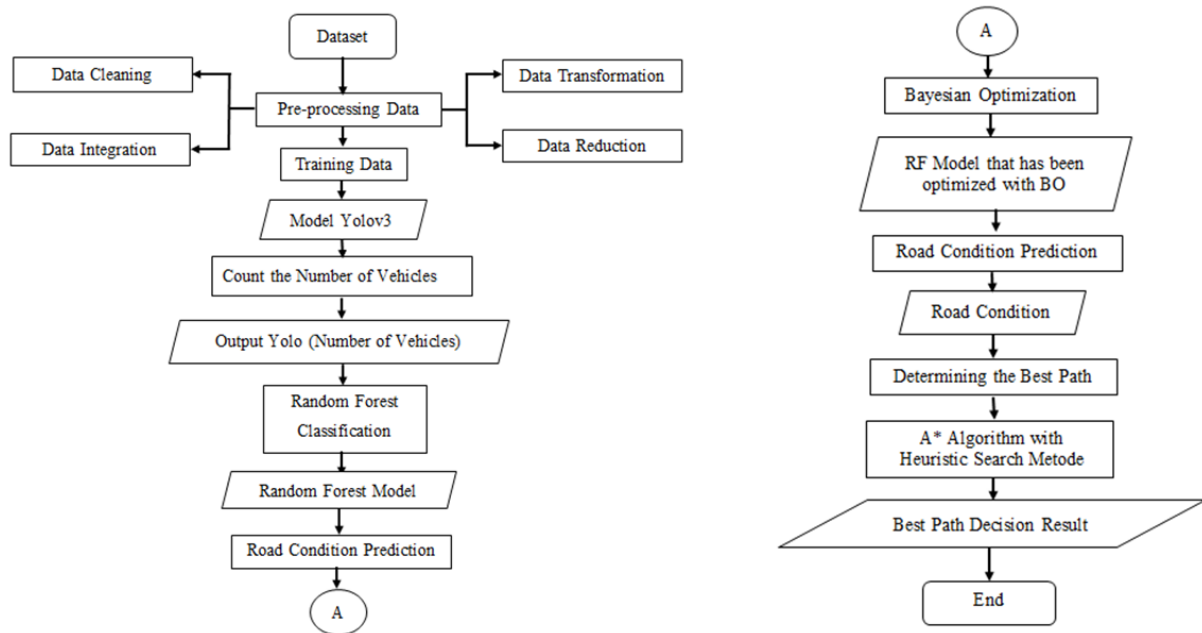


Fig. 1. Research stage.

Therefore, the focus of this research is the application of Intelligent Transportation within the scope of smart cities.

A smart City is a city concept that is intelligent in managing resources to provide better services to the community, both through efficient management and anticipation of unexpected situations [3], [4]. The application of CCTV is becoming common in many cities in Indonesia, including Palembang, as evidence of traffic violations as well as a traffic flow monitoring tool [5], [6].

The following is the formulation of this research problem: How can we best use classification using Random Forests optimized by Bayesian optimization to determine the optimal path, considering the data from YOLOv3 used for object detection on the path. Moreover, how can we find the optimal path using the A\* Heuristic Search algorithm while accounting for the roads' ever-changing conditions. For drivers to reach their destination swiftly and safely, the main problem that needs to be solved is how to avoid any obstacles or traffic on the main route.

In this context, traffic CCTV footage can be further utilized by using the YOLO (You Only Look Once) version 3 application to detect objects in real-time, including counting the number of vehicles with a good level of accuracy [7], [8]. Random Forest algorithm, as an effective machine learning method, can be used to classify road conditions based on key parameters, such as number of vehicles and road width [9]. Optimization using Bayesian optimization can improve the performance of Random Forest to provide more accurate predictions [10]. The results of this model can then be used to determine the traffic condition, whether it is smooth, congested, or jammed, which in

turn contributes to the selection of the best lane.

This research also applies the A\* (A Star) algorithm to find the best path based on road condition parameters and distance traveled. The A\* algorithm can provide the best path-finding solution by considering the estimated closest distance to the destination with high time efficiency [11], [12].

## II. RESEARCH METHOD

In the context of ever-increasing population growth, especially in urban areas, the surge in the number of private vehicles resulting in increasingly congested traffic has become a serious challenge. This phenomenon prompts the need for an effective approach to designing the best path, focusing on key factors such as varying road conditions and distances faced by road users [3], [13]. To accomplish the research goal of identifying the optimal path, the methodology in this study uses heuristic search algorithms in conjunction with machine learning algorithms. Two primary techniques are combined in this study: the A\* Heuristic Search algorithm is used to find the optimal path and Random Forest optimization is used to classify data using Bayesian optimization.

As can be seen in Fig. 1, it begins with the collection of datasets and then proceeds with the data preprocessing stage, which consists of four processes to be carried out. The first is data cleaning, which is the process of filtering unnecessary data sets such as deleting missing or incomplete data. The second is data integration, in this process vehicle images are released, namely motorcycle images with label 0 and car images with label 1. The third stage of data transformation is the process of combining data in a dataset. Finally, in the fourth stage of data reduction,

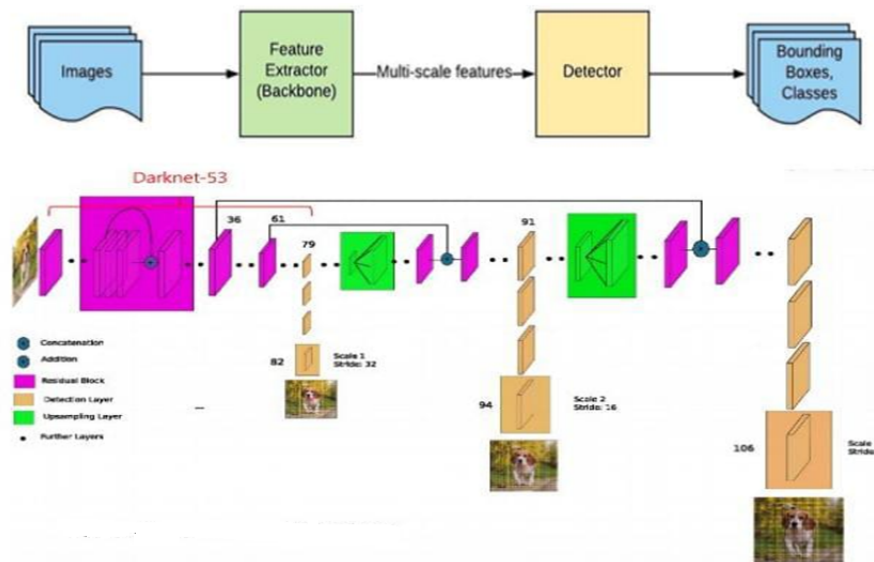


Fig. 2. YOLOv3 network architecture.

namely the division of training and testing data into 80:20. After the pre-processing stage is complete, the next step is to carry out the data training process. In this process, the program will be trained to recognize objects according to the dataset and labels that have been given using the YOLOv3 model. This process involves a feed-forward and backward propagation process where the model will make initial predictions by comparing the predicted results with the actual labels and making adjustments to the weights. This research summarizes several main steps to formulate an effective solution. It starts by performing vehicle detection through the YOLOv3 model, followed by road condition classification using the Random Forest algorithm. This approach is then refined through the application of Bayesian optimization to obtain the optimal configuration [14]. Next, an important step in this research is to determine the best path based on the identified critical parameters: road condition and distance. Through the combination of these steps, it is expected to make a meaningful contribution to overcoming traffic congestion constraints and improving travel efficiency in the midst of increasingly dynamic urban development [15].

#### A. YOLO (You Only Look Once) version 3

The feature extractor in the YOLOv3 architecture uses darknet-53 as the backbone network. Darknet-53 consists of various convolution layers and residual blocks, which together help in generating more robust and complex features from the input image. In the context of object detection, YOLOv3 utilizes a Convolutional Neural Network (CNN) architecture to predict the class and location of objects in the image [7], [16]. The applications of YOLOv3 have extended to various fields, including security systems with object detection, traffic analysis, industrial monitoring, human identification in computer vision systems, and various other purposes. YOLOv3's high speed together with

its good object detection capabilities have made it one of the popular architectures in the development of real-time object detection systems. Its ability to detect objects of different classes simultaneously is also one of its main advantages. YOLOv3 models are trained to recognize several predefined object classes, including but not limited to humans, cars, cats, and so on [8].

YOLOv3 is a computer vision model used to detect objects in images in real time. Its architecture consists of two main components: Darknet-53 as the main network and the detection head to generate predictions. As Fig. 2 shows, in the process, there are operations such as concatenation and addition that help combine information from previous layers with newer ones so that the model can better understand the spatial context. The residual box, or residual block, is used to improve the performance and stability of the model training.

Three precise scales are provided to YOLOv3 by down-sampling the dimensions of the input image by 32, 16, and 8 pixels, respectively. The 82nd layer is responsible for the initial detection. The network samples the image for the first 81 layers, which means that there are 32 steps in the 81 layers. The resulting feature map will measure 13 by 13 if the original image has dimensions of 416 by 416. Here, a  $1 \times 1$  detection kernel is used to make one detection, providing a  $13 \times 13 \times 255$  detection feature map. The feature map from layer 79 is then sampled by  $2 \times$  to a dimension of  $26 \times 26$  after passing through several convolutional layers. The feature map from layer 61 and this one are then deeply combined. Subsequently, the feature map is merged once more, incorporating numerous  $1 \times 1$  convolutional layers to amalgamate the characteristics from the preceding layer (61). The 94 layer is then used for a second detection, producing a detection feature map with dimensions of  $26 \times 26 \times 255$ . The process is repeated with the 91-layer feature map going through

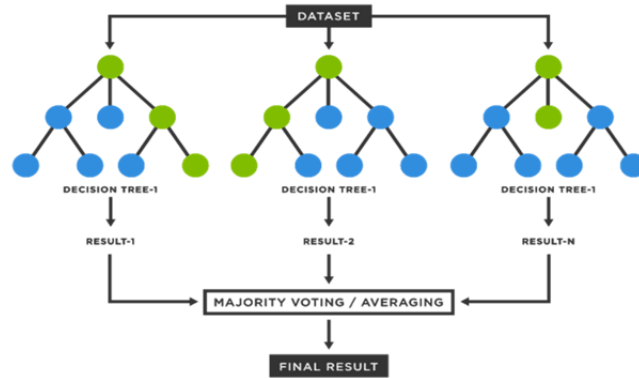


Fig. 3. Random forest architecture.

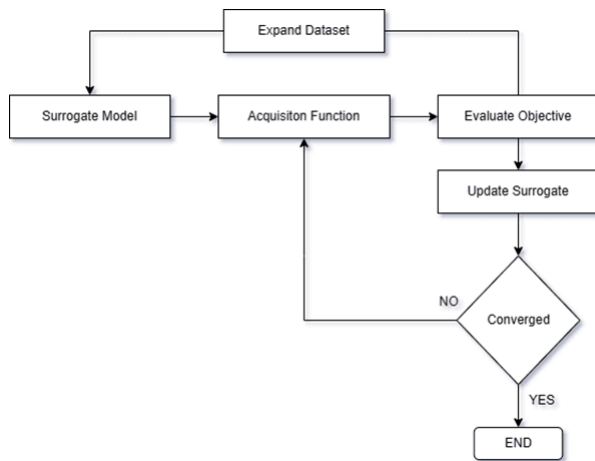


Fig. 4. Bayesian optimization architecture.

several convolutional layers and then being thoroughly merged with the 36-layer feature map. The information from the previous layer (36) is merged once more using numerous  $1 \times 1$  convolutional layers. We make the final of 3 in the 106 layer, resulting in a feature map of size  $52 \times 52 \times 255$ .

**B. Random Forest**

Random Forest is an algorithm for classification and regression on large data sets. It works by using a set of structured decision trees, called classifier trees, and each tree generates votes from input unit  $x$  for the most popular class in the classification [17]. Random Forest is used to predict classification and regression situations on large datasets. This approach involves combining results from each decision tree for classification as well as averaging the regression results [10], [18].

The Random Forest machine learning algorithm combines the outputs of multiple decision trees to produce a single result. As the name implies, the forest is formed from many trees obtained through a bagging or bootstrap aggregating process. As shown in Fig. 3 each tree in Random Forest will output a class prediction. Class predictions with the most votes become candidate predictions in the model. The greater the number of trees, the higher the accuracy and prevents the problem of overfitting.

In the usage stage, after obtaining information about the number of vehicles, both motorcycles and cars, the next step is to predict road congestion conditions using the Random Forest algorithm. First of all, the data that has been collected previously, such as the number of motorcycles, the number of cars, the number of lanes, and the distance traveled at a certain time, is used as input for the Random Forest algorithm. This data is used to provide information to the algorithm regarding various factors related to traffic at the time of observation, as shown in (1).

$$(x) = \arg \max \sum (h_i(x) = Y)k; \quad i = 1 \quad (1)$$

where  $H(x)$  represents the prediction generated by the Random Forest model for an input data  $x$ , this model consists of several decision trees ( $h_i$ ), which then each tree votes ( $Y$  value) on the prediction that will be generated. Finally, the  $Y$  value with the most votes ( $\arg \max$ ) is taken as the final prediction of the Random Forest ensemble model [19].

**C. Bayesian Optimization**

Bayesian optimization is a method used to automatically adjust hyperparameters in machine learning algorithms. The focus is to find the hyperparameter configuration that results in the best performance of the model. As can be seen in Fig. 4, a probabilistic model is used in Bayesian optimization to estimate the objective function. As a "surrogate" or "substitute" for the real objective function, this model is used. At each evaluation point, the model usually generates an estimate of the mean and uncertainty (variance), which provides a sense of our level of confidence in the estimate.

The next step in the evaluation site selection process is to use the probabilistic model's information to determine the objective function's evaluation location. The "acquisition function" method is used in Bayesian optimization to identify the next point in time at which the objective function is evaluated. Based on the acquisition function, the objective function is assessed at the chosen point. The probabilistic model is updated using the evaluation's outcome. The Bayesian inference method is used to update the probabilistic

model following receipt of the objective function evaluation result. Stated differently, the objective function properties are better estimated by the model as it "learns" from the new evaluation data. The objective is to effectively search the search space and identify the ideal value of the objective function with the fewest number of evaluations through iterative repetition of this process.

After applying the Random Forest algorithm to perform a prediction analysis of road congestion conditions based on variables such as the number of motorcycles, the number of cars, the number of lanes, and travel distance, the next step is to optimize the Random Forest model using the Bayesian optimization method [20], [21], as shown in (2).

$$f(x) = GP(m(x), k(x, x')) \quad (2)$$

where  $f(x)$  predicted the value of function  $x$  by modeling the estimated value of the objective function  $f(x)$  which was  $GP$  based on previously evaluated data. Then using the function  $m(x)$  calculateed the average of  $f(x)$  at  $x$ . Finally, the function  $k(x, x')$  would describe the extent to which the value at point  $x$  would correlate with the value at point  $x'$ .

This optimization process aimed to strike the right balance between two important aspects, namely avoiding data overfitting and improving the accuracy of the Random Forest model. By using the Bayesian optimization method, we hoped to find the optimal hyperparameter configuration for the model so that the model could provide accurate and consistent predictions of new data [19], [22].

#### D. A\* Algorithm (Heuristic Search)

The A\* algorithm is known as one of the most popular best-first search algorithms. It evaluates each node by combining the values of  $g(n)$  and  $h(n)$  [11]. Heuristic search is finding solutions to problems by exploring various conditions to increase the variety and speed of finding solutions. In the context of the A\* algorithm that uses heuristic methods, pathfinding in Heuristic Search becomes more efficient because it utilizes the estimated weight value or the estimated remaining distance to reach the destination. The decision on the best path results from considering the factors of road congestion conditions and distance to the destination point. This way, the method produces an optimal path that avoids congestion and optimizes the overall journey [12]. The projected total cost of the path through node  $n$  was given by  $F(n)$ . The equation to determine  $F(n)$  is shown in (3).

$$F(n) = G(n) + H(n) \quad (3)$$

where  $H(n)$  is the projected cost of the shortest path from node  $n$  to the destination node, and  $G(n)$  is the cost from the origin node to the current node  $n$ .

### III. RESULT

In this section, we explain the results obtained from the data training process using YOLOV3 in calculating the number of vehicles; then, the predicted results of road conditions using the Random Forest algorithm are optimized using the Bayesian optimization algorithm. Finally, the best path results were obtained using the A\* algorithm with the Heuristic Search method.

#### A. Count the Number of Vehicles

From the model generated using YOLOv3, the results of average precision accuracy of motorcycles were 72.72 % and cars 79.35 %. The results of mean average precision average AP value of motorcycles and cars where in this model MAP produced a value of 76.03 %. The YOLOv3 model testing result is shown in Table 1.

Table 1. YOLOv3 Model Testing Result

Load Model	Yolov3-obj_last.weights	
Motorcycle	AP	72.72 %
	TP	287
	FP	63
Car	AP	79.35 %
	TP	302
	FP	31
FN	251	
FN	13 sec	
FN	0.7	
FN	0.86	
FN	0.77	
FN	64.50 %	
FN	76.03 %	

After obtaining a new model from YOLOV3, namely the weight and cfg files, the next step was to count the number of vehicles in the CCTV video recordings.

Table 2. Average Accuracy of Motorcycle and Car Readings

Time	Accuracy Motor	Accuracy Car
January 2, 2023	91.86 %	95.96 %
January 3, 2023	94.79 %	98.44 %
January 4, 2023	91.68 %	94.51 %
January 5, 2023	94.35 %	95.64 %
Average Accuracy	93.17 %	96.13 %

As seen on Table 2 on the calculation of the number of vehicles, the results of the number of motorcycles and cars in 4 days (January 2, 3, 4 and 5, 2023) at 3 times (morning, afternoon and evening) with an average accuracy of 93.17 % motorcycles and 96.13 % cars.

#### B. Predicted Road Congestion Condition

After collecting vehicle data, the Random Forest algorithm would be used to predict road congestion conditions based on the number of cars and motorcycles, number of lanes, and travel distance with the existing conditions as in Table 3 to Table 7.

The Random Forest algorithm would create a prediction model by combining the results of various separate decision trees, such as the number of motorcycles, cars, lanes, and travel distance, as shown in Fig. 5. To

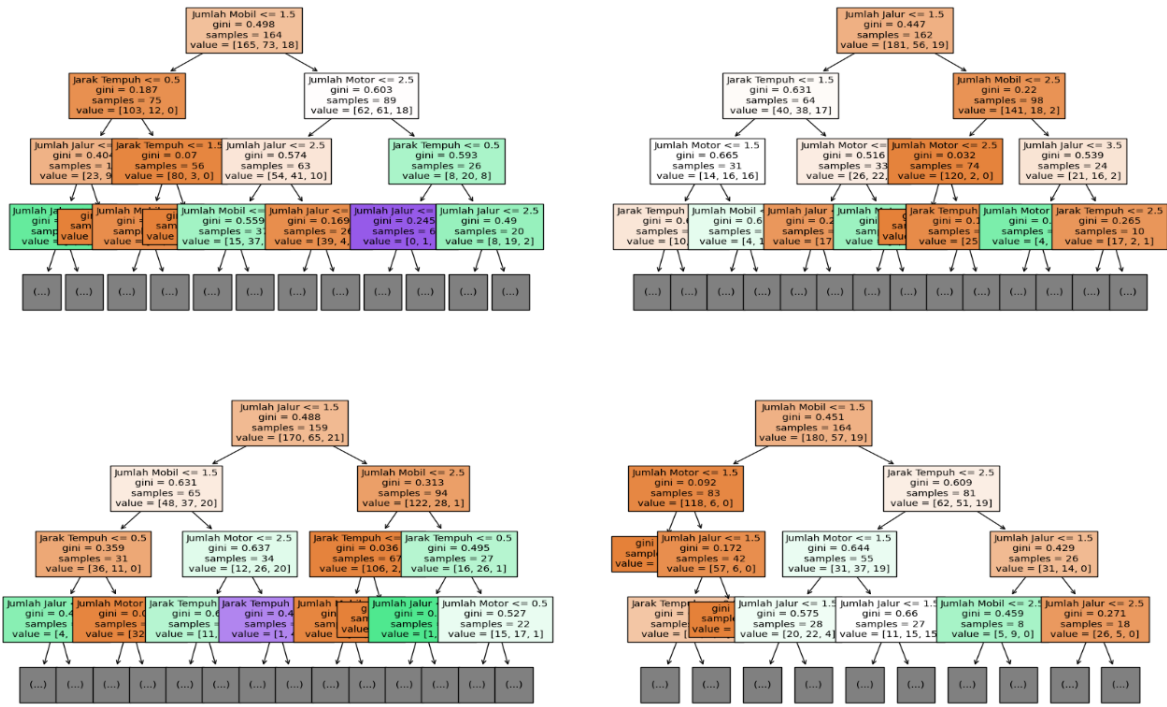


Fig. 5. Decision tree.

Table 3. Motorcycle Vehicle Count Parameter

Number of Motor Vehicles	Quantity	Numeric Value
Little	0 – 15	0
Medium	16 – 30	1
A Lot	31 – 45	2
More	N > 45	3

Table 4. Car Vehicle Count Parameter

Number of Car Vehicles	Quantity	Numeric Value
Little	0 – 10	0
Medium	11 – 20	1
A Lot	21 – 30	2
More	N > 30	3

estimate road conditions, each decision tree in Random Forest took into account various factors. In addition, the prediction of traffic congestion with new data that had never been explored could also be done using the trained Random Forest model.

The model's accuracy on the test data was calculated to assess the model's quality. This was accomplished by contrasting the model's predictions with the actual class labels found in the test data. Utilizing the

Table 5. Number of Lanes parameter

Number of Lanes	Quantity	Numeric Value
Narrow	1	0
Medium	2	1
Wide	3	2
Wider	4	3
Very Wide	N > 4	4

Table 6. Travel Distance Parameter

Travel Distance	Quantity	Numeric Value
Near	0 m – 2,000 m	0
Medium	2,001 m – 4,000 m	1
Far	4,001 m – 6,000 m	2
Farther	N > 6,000 m	3

Table 7. Road Condition Parameter

Road Condition	Numeric Value
Smooth	0
Medium	1
Traffic Jam	2

'accuracy\_score' function was the method utilized to gauge accuracy.

Based on the results obtained from the calculations in Fig. 6, it can be observed that the model's accuracy on the test data is 89 %. This figure showed that the model had a good level of accuracy in predicting road conditions based on the parameters previously described.

Table 8. Random Forest Reading Accuracy Results

No.	Time	Accuracy
1.	Morning	97.91 %
2.	Afternoon	87.50 %
3.	Evening	89.58 %
	Total Accuracy	91.66 %

Table 8, regarding the accuracy of the readings, shows an average accuracy of 91.66 %, and when compared to the model, has an accuracy difference of 2.66 %. This showed that the accuracy obtained in the Random Forest model showed an accuracy that was quite close to the actual accuracy. Overall, the accuracy reading results showed varying levels of accuracy. From a total of 144 vehicle calculation data, 12 density condition results did not comply with the logic rules of road density conditions, and 132 data met the logic rules of road density conditions.

After applying the Random Forest algorithm for predicting road density conditions, the next stage

```
In [18]: # Evaluasi model
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.91	0.98	0.94	42
1	0.81	0.76	0.79	17
2	1.00	0.60	0.75	5
accuracy			0.89	64
macro avg	0.91	0.78	0.83	64
weighted avg	0.89	0.89	0.89	64

Fig. 6. Decision tree.

```
In [23]: # Cetak parameter terbaik yang ditemukan
print("Parameter terbaik:", optimizer.best_params_)

Parameter terbaik: OrderedDict([('bootstrap', False), ('max_depth', 10), ('max_features', 0.9181940453820531), ('min_samples_leaf', 1), ('min_samples_split', 7), ('n_estimators', 194)])

In [24]: # Buat model Random Forest baru dengan parameter terbaik
best_rf_model = RandomForestClassifier(**optimizer.best_params_)

In [25]: # Latih model Random Forest baru dengan parameter terbaik
best_rf_model.fit(X_train, y_train)

Out[25]:
RandomForestClassifier
RandomForestClassifier(bootstrap=False, max_depth=10,
max_features=0.9181940453820531, min_samples_split=7,
```

Fig. 7. Best parameters.

```
In [28]: # Prediksi Label pada data uji menggunakan model yang dioptimasi
y_predict = best_rf_model.predict(X_test)

In [29]: # Evaluasi performa model
accuracy = accuracy_score(y_test, y_predict)
loss = 1 - accuracy

print("Akurasi Model Random Forest yang telah dioptimasi: {:.2f}".format(accuracy))
print("Loss:", loss)

Akurasi Model Random Forest yang telah dioptimasi: 0.92
Loss: 0.078125

In [30]: # Evaluasi model
print(classification_report(y_test, y_predict))
```

	precision	recall	f1-score	support
0	0.93	0.98	0.95	42
1	0.88	0.82	0.85	17
2	1.00	0.80	0.89	5
accuracy			0.92	64
macro avg	0.94	0.87	0.90	64
weighted avg	0.92	0.92	0.92	64

Fig. 8. Accuracy of bayesian optimization.

was to optimize the Random Forest model using the Bayesian optimization method. This optimization aimed to ensure a proper trade-off between data overfitting and the high accuracy of the Random Forest model. The optimization process automatically explored and exploited the hyperparameter search space to find the optimal configuration.

After building the model, the best parameters of the model were obtained, which can be seen in Fig. 7 which are as follows: (bootstrap: False), (max depth: 10), (max features: 0.9181940453820531), (min samples leaf: 1), (min samples split: 7), (n estimator: 194).

From Fig. 8, the model evaluation shows that the accuracy of the optimized model is 92 % with a loss value of 0.078125. Which means that the model has increased accuracy by 3 %.

The average reading accuracy in Table 9 is 92.36

Table 9. Bayesian Optimization Reading Accuracy Results

No.	Time	Accuracy
1.	Morning	97.75 %
2.	Afternoon	91.65 %
3.	Evening	87.50 %
	Total Accuracy	92.36 %

%, which is 0.36 % higher than the model accuracy. The overall accuracy reading findings showed varying levels of accuracy. There were 11 density condition findings out of 144 vehicle count data points that did not follow the logical guidelines for road density conditions. 133 data points conform to the road density standard.

### C. Determination of the Best Path

The A\* algorithm was used to process the data using a heuristic method to determine the best path. The selection of the best path was done after taking into account the road density conditions and the required

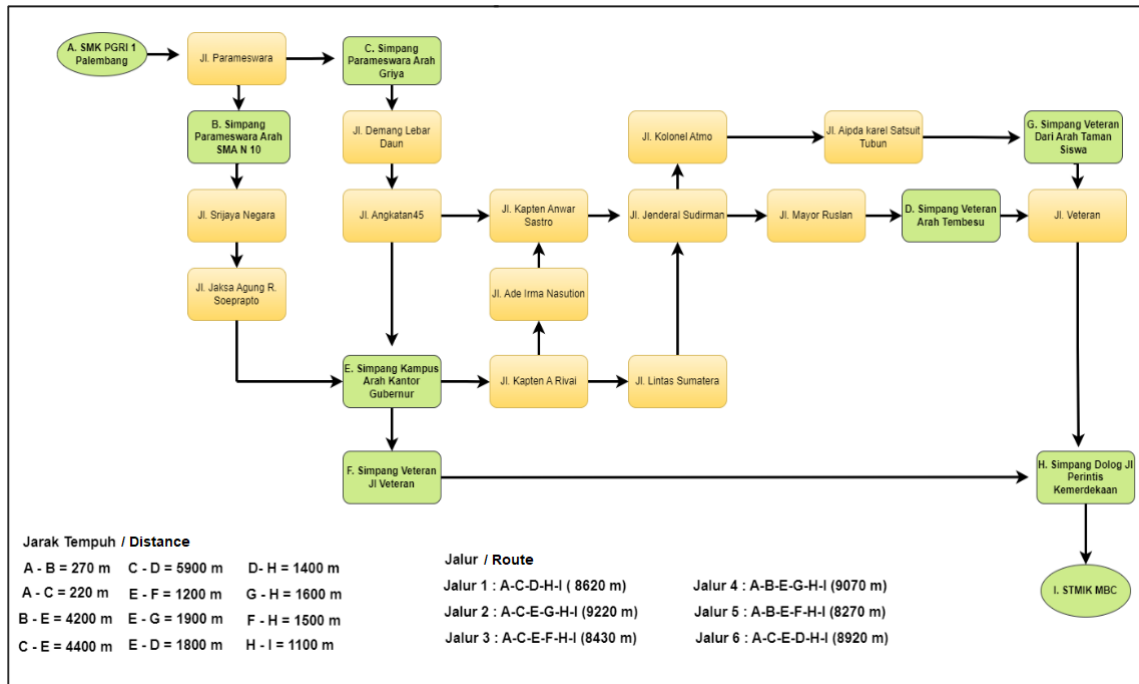


Fig. 9. Travel route graph.

travel time. The Euclidean distance was used to estimate the remaining cost (heuristic) from the current node to the target node; based on the weight of the road and the conditions faced, this algorithm attempted to determine the best path with the lowest overall weight. There are six path options that can be seen in Fig. 9.

```
def calculate_weight(distance, condition):
    if condition == "smooth":
        return distance * 1
    elif condition == "medium":
        return distance * 2
    elif condition == "traffic_jam":
        return distance * 3
```

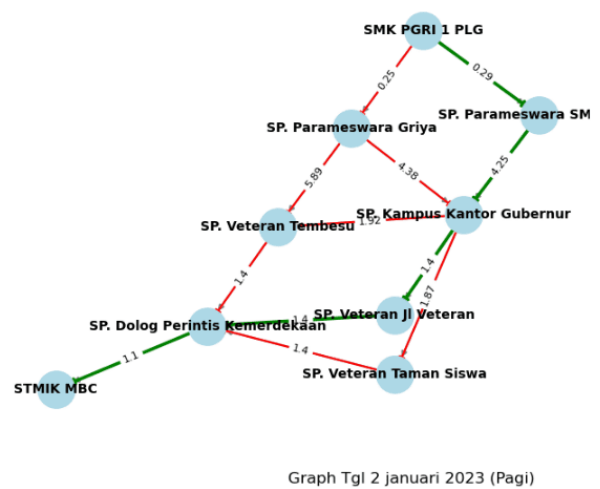
Based on the distance and condition of the road, the calculate\_weight function determines the path's weight. The A\* algorithm will consider this weight while determining the optimum course of action. There are two inputs for this function:

- Distance: The distance between the current node and its weighted surrounding nodes.
- Condition: This refers to the road's state, which may be "smooth," "medium," or "traffic jam." The resultant weight will be impacted by this circumstance.

The algorithm would iterate for neighbor, distance and condition in graph[current][1:] to examine the current node's neighbors. The new weight  $G(n)$  was computed for each neighbor using the formula tentative\_g\_score. Based on the values for the road length and condition, the function calculate\_weight (distance condition) would determine the weight of the road. The information was updated by altering came\_from, g\_score, and  $F(n)$  (f\_score) for that neighbor if the

newly calculated  $G(n)$  value was less than the previous  $G(n)$  value for that neighbor.

The algorithm would then produce the best path based on the minimum weight criteria. This optimal route would show the total weight of each road edge traveled between the starting and ending points. The A\* algorithm's output also showed the optimum path in detail, including the order of nodes that must be crossed. This research also represented the road graph to make the A\* algorithm results easier to understand. The edges of the road on the optimal path were indicated on this road graph by the color green, while the edges of other roads were indicated by the color red.



Graph Tgl 2 januari 2023 (Pagi)

Fig. 10. Best path on January 2, 2023 morning time (path 5).



Fig. 10 shows the best path results that resulted in the best path five on January 2, 2023, morning (06:45 WIB - 06:46 WIB). As can be seen, path 5 stands out as the best path with a weight of 8.44 based on the results of calculations using the A\* algorithm to determine the best path of the six route options or paths evaluated. This path was the best option because it had the shortest travel distance and the lightest weight when compared to the other paths, which included paths 1 and 6 (each weighing 8.63, 9.0, 8.53, 8.91, and 9.05 respectively).

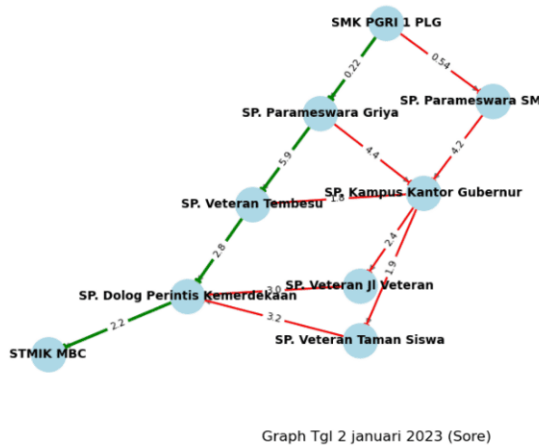


Fig. 11. Best path on January 2, 2023 at afternoon (path 1).

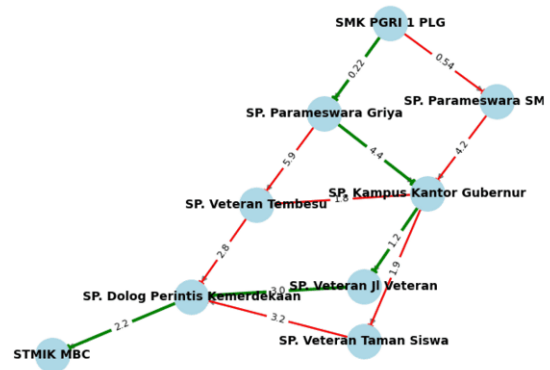
Fig. 11 shows the results of the best path selection, where path one was selected as the best choice on January 2, 2023, in the afternoon. Path 1 was chosen as the best path because it had the lowest weight of 11.12 compared to other paths, but it was not the shortest path because of the many smooth road conditions. The total weight of Path 2 was 11.92, the total weight of Path 3 was 12.22, which was greater than Path 2 and Path 1, and the total weight of Path 4 was 12.04, which was almost the same as the total weight of Path 2, and the total weight of Path 5 was 12.34. Path 5 even yielded the path with the maximum weight on this date when compared to other dates.

While on January 5, 2023, at noon, the best path chosen was path 3. It can be seen in Fig. 12 that path 3 produces the smallest weight of 11.0 compared to other paths that have greater weight, namely, path 1 (11.12), path 2 (11.92), path 4 (12.04), path 5 (11.14), and path 6 (11.42).

IV. DISCUSSION

Based on the results of determining the best path using the A\* algorithm, the best path results from 12 events (4 days three times) were obtained, where the average chooses path five as the best path from 6 options.

The best path for each date and time is shown in Table 10, where, on average, 9 out of 12 events



Graph Tgl 5 Januari 2023 (Siang)

Fig. 12. Best path on January 5, 2023 at noon (path 3).

Table 10. Best Path and Weighted Value by Date and Time

No.	Date	Time	Best Path	Weight
1.	January 2, 2023	Morning	Path 5	8.44
2.	January 2, 2023	Noon	Path 5	8.44
3.	January 2, 2023	Afternoon	Path 1	11.12
4.	January 3, 2023	Morning	Path 5	8.27
5.	January 3, 2023	Noon	Path 5	10.87
6.	January 3, 2023	Afternoon	Path 5	10.87
7.	January 4, 2023	Morning	Path 5	8.27
8.	January 4, 2023	Noon	Path 5	11.14
9.	January 4, 2023	Afternoon	Path 5	13.74
10.	January 5, 2023	Morning	Path 5	8.27
11.	January 5, 2023	Noon	Path 3	11.02
12.	January 5, 2023	Afternoon	Path 1	11.12
<b>Average Weighted Value of the Best Path</b>				<b>10.13</b>

chose path five as the best path. The shortest distance influenced the selection of the best path traveled and road conditions that tended to be "smooth". Path 5 had the shortest travel distance of other paths, 8,270 m. But even though path 5 was the path with the shortest distance, if the road conditions tended to be in a state of congestion, resulting in a large weight value, it would not be chosen as the best path as of January 2, 2023, in the afternoon, January 5, 2023, in the afternoon and evening which chose paths 1 and 3 as the best path.

V. CONCLUSION

From the results of this study, the precision figure of 85.44 % showed how important the proper annotation or labeling procedure was in the YOLOv3 method to get accurate object detection results. YOLOv3 could distinguish objects well from a close distance, but when the object was too far away from the camera, the detection accuracy would decrease. The quality of images and videos in the dataset had an impact on the performance of the model during training; images and videos with higher quality would give results with higher accuracy and better object detection.

Then, during classification, the Random Forest algorithm had a reading accuracy of 91.66 % and a model accuracy of 89 %. The model accuracy increased to

92 %, and the reading accuracy to 92.36 % after being improved with Bayesian optimization. This showed how Bayesian optimization could improve model performance by 3 % and reading accuracy by 0.7 %.

Then, on average, 9 out of 12 instances selected path five as the most frequently selected path. Road congestion conditions that were often "smooth" and the shortest travel route compared to other routes were factors in this selection. However, the road with the shortest route was not necessarily the best route because if the road conditions tended to be "congested," the route would not be selected as the best route. The width of the road also influenced the selection of the ideal path. Since it could accept more vehicles, lowering the chance of congestion and causing the A\* algorithm to prefer it, routes with larger roads could reduce the amount of traffic on the road.

#### REFERENCES

- [1] J. Gao, X. Xu, X. Zhang, S. Xu, and Q. Pu, "Path planning of mobile robot based on improved A\* algorithm," in *2022 5th World Conf. Mech. Eng. Intell. Manuf. WCMEIM 2022*, pp. 1098–1102, 2022, doi: 10.1109/WCMEIM56910.2022.10021353.
- [2] H. Mandaku and M. Tukan, "Studi penerapan intelligent transportation system (ITS) di Kabupaten Seram bagian barat," *ARIKA*, vol. 04, no. 1, 2010.
- [3] A. K. Dogra and J. Kaur, "Moving towards smart transportation with machine learning and Internet of Things (IoT): A review," *J. Smart Environ. Green Comput.*, pp. 3–18, 2022, doi: 10.20517/jsegc.2021.09.
- [4] S. Mukherjee, N. B. Muppalaneni, S. Bhattacharya, and A. K. Pradhan, "Intelligent systems for social good: Theory and practice," Springer, June. 2022. [Online]. Available: <https://link.springer.com/book/10.1007/978-981-19-0770-8>
- [5] K. H. Nam Bui, H. Yi, and J. Cho, "A multi-class multi-movement vehicle counting framework for traffic analysis in complex areas using CCTV systems," *Energies*, vol. 13, no. 8, 2020, doi: 10.3390/en13082036.
- [6] Tasia, "5 kegunaan CCTV lampu merah," [auto2000.co.id](http://auto2000.co.id), 2020.
- [7] M. Harahap, J. Elfrida, P. Agusman, M. Rafael, R. Abram, and K. Andrianto, "Sistem cerdas pemantauan arus lalu lintas dengan YOLO (You Only Look Once v3)," in *Semin. Nas. APTIKOM*, pp. 367–379, 2019.
- [8] F. Zhang, C. Li, and F. Yang, "Vehicle detection in urban traffic surveillance images based on convolutional neural networks with feature concatenation," *Sensors*, vol. 19, no. 3, 2019, doi: 10.3390/s19030594.
- [9] N. M. Abdulkareem and A. M. Abdulazeez, "Machine learning classification based on random forest algorithm: A review," *J. Sci. Bus.*, vol. 27, no. January, pp. 128–142, 2021, doi: 10.5281/zenodo.4471118.
- [10] E. Fitri, "Analisis sentimen terhadap aplikasi ruangguru menggunakan algoritma naive bayes, random forest dan support vector machine," *J. Transform.*, vol. 18, no. 1, p. 71, 2020, doi: 10.26623/transformatika.v18i1.2317.
- [11] R. F. Oktanugraha and S. R. Nudin, "Implementasi algoritma A\* (A star) dalam penentuan rute terpendek yang dapat dilalui non player character pada game good thief," *J. Informatics Comput. Sci.*, vol. 2, no. 01, pp. 74–85, 2020, doi: 10.26740/jinacs.v2n01.p74-85.
- [12] X. Zhang, P. Wu, C. Yuan, Y. Wang, D. Metaxas, and C. Chen, "Heuristic search for homology localization problem and its application in cardiac trabeculae reconstruction," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2019-Augus, no. c, pp. 1312–1318, 2019, doi: 10.24963/ijcai.2019/182.
- [13] A. Terentyev, M. Karelina, V. Egorov, A. Andreev, and K. Reza Kashyzadeh, "Model for determining optimal routes in complex transport systems," in *Transportation Research Procedia, Elsevier B.V.*, 2021, pp. 679–687. doi: 10.1016/j.trpro.2021.09.100.
- [14] T. Wang, X. Wang, R. Ma, X. Li, X. Hu, F. T. S. Chan, and J. Ruan, "Random forest-bayesian optimization for product quality prediction with large scale dimensions in process industrial cyber-physical systems," *IEEE Internet of Things Journal*, vol. 7, no. 9, 2020, doi: 10.1109/JIOT.2020.2992811.
- [15] D. Rachmawati, P. Sihombing, and B. Halim, "Implementation of best first search algorithm in determining best route based on traffic jam level in Medan City," in *2020 Int. Conf. Data Sci. Artif. Intell. Bus. Anal. DATABIA 2020 - Proc.*, pp. 5–12, 2020, doi: 10.1109/DATABIA50434.2020.9190626.
- [16] M. S. Chauhan, A. Singh, M. Khemka, A. Prateek, and R. Sen, "Embedded CNN based vehicle classification and counting in non-laned road traffic," in *ACM Int. Conf. Proceeding Ser.*, no. September, 2019, doi: 10.1145/3287098.3287118.
- [17] I. Reis, D. Baron, and S. Shahaf, "Probabilistic random forest: A machine learning algorithm for noisy data sets," *Astron. J.*, vol. 157, no. 1, p. 16, 2018, doi: 10.3847/1538-3881/aaf101.
- [18] M. Sun, R. Zhou, C. Jiao, and X. Sun, "Severity analysis of hazardous material road transportation crashes with a bayesian network using highway safety information system data," *Int. J. Environ. Res. Public Health*, vol. 19, no. 7, 2022, doi: 10.3390/ijerph19074002.
- [19] M. Wu, D. Shan, Z. Wang, X. Sun, J. Liu, and M. Sun, "A bayesian network model for real-time crash prediction based on selected variables by random forest," in *ICTIS 2019 - 5th Int. Conf. Transp. Inf. Saf.*, pp. 670–677, 2019, doi: 10.1109/ICTIS.2019.8883694.
- [20] H. Alibrahim and S. A. Ludwig, "Hyperparameter optimization: Comparing genetic algorithm against grid search and bayesian optimization," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 4, no. 5, pp. 740–740, 2020, doi: 10.1109/tetci.2020.3020707.
- [21] S. Belakaria, A. Deshwal, and J. R. Doppa, "Max-value entropy search for multi-objective Bayesian optimization," *Adv. Neural Inf. Process. Syst.*, vol. 32, no. NeurIPS, 2019.
- [22] J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, H. Lei, and S. H. Deng, "Hyperparameter optimization for machine learning models based on Bayesian optimization," *J. Electron. Sci. Technol.*, vol. 17, no. 1, pp. 26–40, 2019, doi: 10.11989/JEST.1674-862X.80904120.