Optimization of Vehicle Speed Calculation on Raspberry Pi Using Sparse Random Projection

Charmaine C. Paglinawan
School of EECE
Mapúa University
Manila, Philippines
ccpaglinawan@mapua.edu.ph

Analyn N. Yumang
School of EECE
Mapúa University
Manila, Philippines
anyumang@mapua.edu.ph

Luke Clark M. Andrada
School of EECE
Mapúa University
Manila, Philippines
lcmandrada@mymail.mapua.edu.ph

Eduardo C. Garcia
School of EECE
Mapúa University
Manila, Philippines
ecgarcia@mymail.mapua.edu.ph

Joseph Merlin F. Hernandez
School of EECE
Mapúa University
Manila, Philippines
jmfhernandez@mymail.mapua.edu.ph

***Abstract*— In order to weaken the barrier of implementation of speed limit enforcement in the country, ways to improve vehicle speed calculation such as image processing can be explored. It was observed that existing prototypes suffer with low effective frame rate, the average rate at which the system processes video frames, considering it unfit for real-time setup. In this study, a vehicle speed calculation system was developed on Raspberry Pi with Gaussian Mixture Model for vehicle detection and Kalman Filter for vehicle tracking on OpenCV, and was optimized with Sparse Random Projection on scikit-learn by projecting the video to a low-dimensional subspace. The prototype was tested and analyzed in performance, iún terms of effective frame rate, and in accuracy, in terms of vehicle speed, by applying paired t-test and linear regression analysis to prove if the optimization improved the performance and accuracy of vehicle speed calculation. It was found that Sparse Random Projection significantly improved the performance of the system at 7.08 fps and in effect improved accuracy with significant correlation between actual and calculated vehicle speed at an average absolute difference error of ±0.76 kph. In contrary, vehicle speed calculation without optimization performed at 3.29 fps and ±1.25 kph difference error.**

Keywords—Vehicle Speed Calculation, Raspberry Pi, Gaussian Mixture Model, Kalman Filter, Sparse Random Projection

I. INTRODUCTION

In a report conducted by the World Health Organization, it was stated that speed and speed limit are critical factors in the matter of road safety. It was further reported, however, that effectiveness of local speed limit enforcement was given only an average rating. It can then be inferred that ways to improve enforcement can be explored. An instance of the case is to improve vehicle speed calculation to weaken its barrier of implementation. Some of the ways to calculate vehicle speed include Radio Detection and Ranging, Light Detection and Ranging, Single Motion Blurred Image, and Speed Detection Camera System where the latter has capacity to work as alternative with possible solutions to the limitations of the others [1].

In an effort to apply image processing on vehicle speed calculation, a study used Raspberry Pi as embedded board for its fair balance of cost and performance [2] and OpenCV as software for its efficiency and efficacy [3]. The study concluded that even if vehicle speed calculation is complex demanding high specification, OpenCV on Raspberry Pi has the capacity to deliver results but only at an effective frame rate of 14.66 fps for Raspberry Pi B+ and 15.42 fps for Raspberry Pi 2 at the given setup. In another study employing Gaussian Mixture Model and Kalman Filter for vehicle speed calculation with a single trial accuracy of 88.45%, it was reported that optimization is suggested for real-time operations for better vehicle detection, tracking and speed calculations [4].

In consideration of the aforesaid limitations and recommendations, the problem that the study undertakes is the lack of optimization in the performance of image processing systems particularly on vehicle speed calculation in terms of effective frame rate, the average rate at which the system processes video frames, implemented on an embedded board in real-time setup.

The general objective of the study is to optimize the performance of vehicle speed calculation. It entails the following specific objectives: (1) to develop a vehicle speed calculation prototype on Raspberry Pi employing Gaussian Mixture Model (GMM) for vehicle detection and Kalman Filter (KF) for vehicle tracking with OpenCV; (2) to optimize the performance of the system with Sparse Random Projection (SRP) on scikit-learn; (3) and to test its performance in terms of effective frame rate and its accuracy in terms of vehicle speed.

The study covered the implementation and optimization of vehicle speed calculation with OpenCV and scikit-learn on Raspberry Pi employing GMM, KF and SRP. It only tested and verified the effects of the optimization on the performance and accuracy of the system, and neither considered complex factors such as environment conditions nor develop a speed limit enforcement system. The following

978-1-5386-7767-4/18/$31.00 ©2018 IEEE

content of the study consists of the methodology, results and discussion, conclusion and future works.

II. METHODOLOGY

A. Research Design

The study observed the framework of quantitative development research for objective and conclusive results by applying t-test and linear regression analysis on effective frame rate and vehicle speed data to verify the hypothesis that SRP improves the performance and accuracy of vehicle speed calculation. In this consideration and under the viewpoint of application and quantitative type, the study followed the approach of development research wherein knowledge and observations from existing studies are applied on an existing system in the effort of improvement. On that account, the core of the study consists of optimization research, hardware and software development, optimization implementation, system testing, application of statistical treatment, and analysis and discussion of the results.

B. Conceptual Framework

The video frames captured by the camera module are introduced to the system as input. It will undergo color conversion, extraction of region of interest, implementation of SRP, GMM with morphological operations and blob detection and KF, and calculation of vehicle speed and effective frame rate to deliver an output of video display, video and data file. The input, processes and output of the system are represented by the conceptual framework shown in Fig. 1.

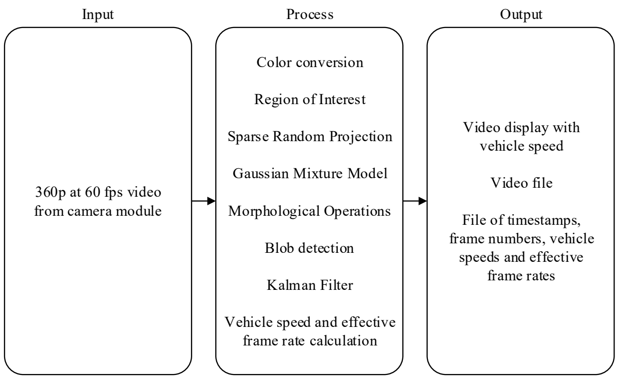

Fig. 1 Conceptual Framework

C. Hardware Development

The prototype consists of Raspberry Pi 3 Model B as embedded board, Raspberry Pi Camera Module v2 to capture video, power bank as battery to power the system, laptop display connected remotely through Virtual Network Computing (VNC) protocol over a wireless router for real-time video feed, and 8GB MicroSD card where the Raspbian operating system, open-source software, and files are installed and saved. The prototype block diagram representing the relationship between the components is shown in Fig. 2.

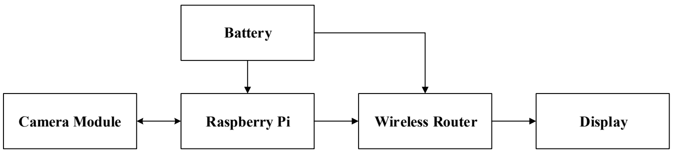

Fig. 2 Prototype Block Diagram

D. Software Development

The prototype is remotely connected to a laptop through VNC protocol where control of the system is managed and real-time video display is streamed. The system starts with the capture and introduction of video frames to the program as shown in Fig. 3. It then executes a series of preprocessing steps starting with the conversion of color into grayscale to lighten the burden of processing [5] and to work as preprocessor before object detection [6]. A region of interest is then configured for its potential and capacity to optimize object detection particularly GMM [7].

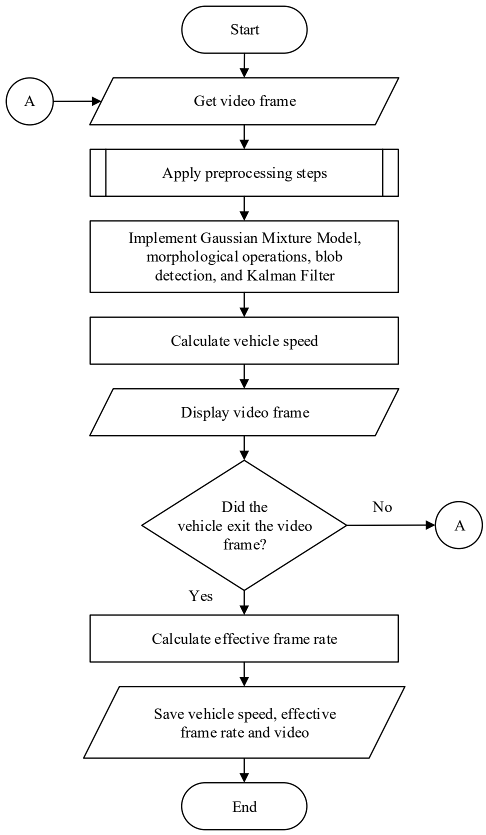

Fig. 3 System Flowchart

The optimization of system performance is implemented through SRP with scikit-learn as shown in Fig. 4 where each high-dimensional video frame is projected into a low-dimensional subspace while preserving the information content through a sparse random projection matrix. In this way, SRP works as a compression technique in the context of image processing [8].

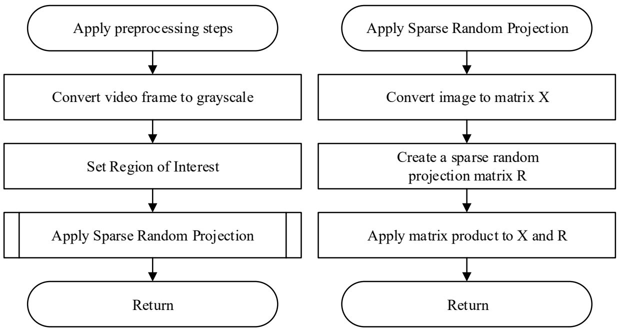

Fig. 4 Subroutine Flowchart

It was shown in [9] that random projection preserves similarities of data vectors better than other dimensionality reduction techniques. In complement with GMM, it was proven that random projection lead to a computationally efficient yet relatively accurate background subtraction algorithm that performs up to six times faster than conventional methods [10], and to a pedestrian detection system that achieves up to ten times speedup with comparable accuracy to standard systems in complement with shape prior [11]. It consumes even fewer memory blocks leading to faster computing if sparse random matrix is implemented where the probability of zero elements is greater than that of nonzero elements [12]. SRP outputs the projected video frame data, as shown in Fig. 5, preserved to a certain level of control with fewer dimensions that succeeding algorithms, in theory, will work on more efficiently.

Fig. 5 Video Projection: Original and Enlarged

GMM is then applied to the video frame with OpenCV to identify the foreground from the background as shown inverted in Fig. 6 where the former is converted into white pixels and the latter into black pixels [13]. GMM was selected for its reliability against varying light and conditions [14] and for delivering less noise at better accuracy [6]. The output of background subtraction is not perfect, thus the frame undergoes morphological operations particularly opening, erosion followed by dilation, for noise reduction and closing, dilation followed by erosion, for hole filling. The previous operations provide a more refined delineation of objects or blobs that are boxed by blob detection for segmentation.

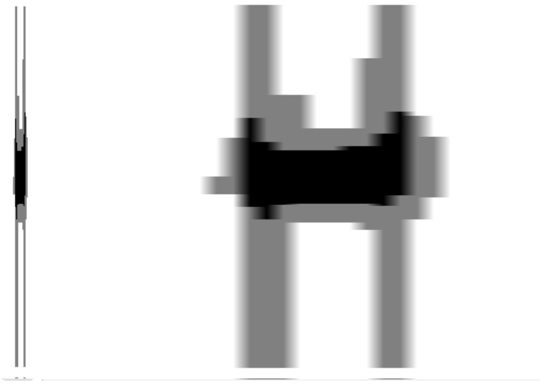

Fig. 6 Background Subtraction Output: Original and Enlarged

In order to identify the vehicle captured in the frame, KF is applied to tag it consistently by predicting the probabilities of its next position. When the vehicle is tracked, a bounding box is drawn enclosing it and the vehicle speed is calculated as shown in Fig. 7. KF was selected for its performance as recursive method for tracking objects in a video frame [14] while OpenCV was used as platform for GMM and KF for its efficiency and efficacy in complex image processing for real-time applications [2].

Fig. 7 Video Output

In the calculation of vehicle speed, the front coordinate of the vehicle is obtained from the boundary box of the blob to serve as reference. The vehicle speed is calculated through (1) where d is calculated through pinhole model, t_1 and t_2 are the timestamps when the test vehicle crossed a certain zone, and 3.6 is the conversion factor from meters per second to kilometers per hour. The video frame is then displayed with the lines, bounding box and vehicle speed drawn.

$$s_{calculated} = \frac{d}{t_2 - t_1} \times 3.6 \tag{1}$$

If the test vehicle has not exited the video frame yet, the system loops back to capture. Otherwise, the effective frame rate is calculated through (2) where T_1 and T_2 are the timestamps of the initial frame, F_1, and the last frame, F_2. The calculated vehicle speed and effective frame rate, and the video itself are then saved.

$$E = \frac{F_2 - F_1}{T_2 - T_1} \tag{2}$$

E. *Sparse Random Projection*

In the application of SRP, the image is converted to data matrix, X, with dimensions d and n containing the intensity of each pixel. The main theoretical foundation of SRP stands on Johnson-Lindenstrauss lemma which states that a set of points in a high-dimensional space can be embedded into a low-dimensional subspace while approximately preserving the distances between points [15].

A projection matrix, R, with dimensions n and k reflecting the initial number of features and the number of components is randomly populated with values and probabilities defined in [12]. With this population technique, the projection matrix is random and sparser without the expense of quality, thus is more computationally efficient than conventional methods while maintaining to be relatively accurate.

In order to project the high-dimensional video frame to a low-dimensional subspace, the matrix product of the data matrix, $\mathrm{X}_{\mathrm{d}\times\mathrm{n}}$, and the random projection matrix, $\mathrm{R}_{\mathrm{n}\times\mathrm{k}}$, is taken to solve the projected matrix, $\mathrm{A}_{\mathrm{d}\times\mathrm{k}}$, as shown in (3) where n is reduced to k, the number of components of the projection. The projected matrix contains fewer data points than the initial video frame with its quality preserved as ensured by the lemma. In this way, the system can work with fewer resources and faster processing without significant loss of data.

$$\mathrm{A}_{\mathrm{d}\times\mathrm{k}}=\mathrm{X}_{\mathrm{d}\times\mathrm{n}}\mathrm{R}_{\mathrm{n}\times\mathrm{k}} \tag{3}$$

F. *Experimental Setup*

The prototype was mounted at an inclination angle, β, of 40° on a post with a height of 3.2 m. The test vehicle was driven around 20 kph, as per the speed limit for a two-way lane, under the prototype on the tests at Reagan St. Phase 4-A Parkwood Greens Maybunga, Pasig City, a two-lane road located inside a village where a certain level of control over the environment can be taken to reduce risks of error. Its speed data as calculated by the system was recorded past d as shown in Fig. 8 while another speed data measured by a digital speedometer was concurrently logged through screen recording synchronized with the prototype through Network Time Protocol (NTP).

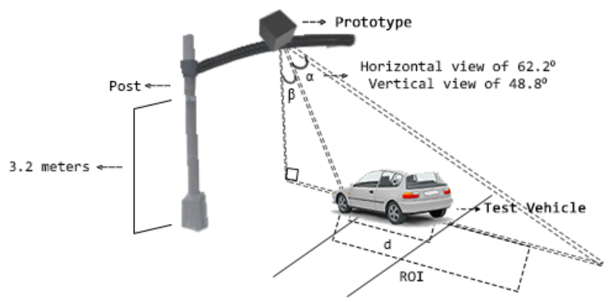

Fig. 8 Experimental Setup

III. Results and Discussion

A. *Testing*

The testing was divided into two setups: vehicle speed calculation system without and with SRP. The tests were iterated for 15 trials on the consideration that the notion that SRP optimizes and improves performance and accuracy is yet to be proven.

The performance of the prototype was tested in terms of effective frame rate, the average rate at which the system processes video frames, calculated for each trial of each setup. The effective frame rate data for each trial and setup were retrieved from a file at the end of the testing. The statistical treatment paired t-test was applied to the effective frame rate data listed at Table I to determine if SRP optimizes the performance of the prototype significantly.

TABLE I
Performance Test in terms of Effective Frame Rate

Trial	Vehicle Speed Calculation without SRP (fps)	Vehicle Speed Calculation with SRP (fps)
1	4.03	10.23
2	4.06	7.92
3	4.01	6.28
4	4.01	7.09
5	3.83	6.68
6	3.99	7.10
7	2.08	6.01
8	2.28	7.57
9	3.15	7.15
10	3.96	6.29
11	3.05	6.85
12	2.19	6.69
13	3.90	5.47
14	2.17	8.42
15	2.65	6.39
Avg.	3.29	7.08

The average effective frame rate of the system optimized with SRP was calculated to be at 7.08 fps which is more than twice that of the system without optimization at 3.29 fps as shown in Table I. It can be observed that the former has the capacity to process more video frames than the latter that it can be inferred to perform relatively better.

The accuracy of the system was tested in terms of vehicle speed with respect to a standard. The actual average vehicle speed is calculated through (4) where s_1 and s_2 are recordings taken from a digital speedometer as the vehicle traverses d.

$$s_{actual} = \frac{s_1 + s_2}{2} \tag{4}$$

The actual speed data were retrieved from the digital speedometer that served as the standard device while the calculated speed data were retrieved from a file at the end of the testing. The data with corresponding absolute difference error to indicate the discrepancy between the actual and calculated speed for vehicle speed calculation without and with SRP are listed in Table II and III.

TABLE II
Accuracy Test for Vehicle Speed Calculation without SRP

Trial	Actual Speed (kph)	Calculated Speed (kph)	Difference Error (kph)
1	14.00	11.85	2.15
2	16.00	17.37	1.37

3	16.00	15.93	0.07
4	17.50	16.54	0.96
5	16.50	16.79	0.29
6	16.00	16.64	0.64
7	17.50	17.37	0.13
8	18.00	17.76	0.24
9	19.50	16.47	3.03
10	16.50	15.16	1.34
11	18.50	15.96	2.54
12	16.00	15.54	0.46
13	16.00	16.58	0.58
14	18.50	16.22	2.28
15	12.50	15.14	2.64
Avg.			1.25

TABLE III
ACCURACY TEST FOR VEHICLE SPEED CALCULATION WITH SRP

Trial	**Actual Speed (kph)**	**Calculated Speed (kph)**	**Difference Error (kph)**
1	17.00	16.71	0.29
2	19.00	18.54	0.46
3	21.50	21.02	0.48
4	18.00	18.56	0.56
5	19.50	21.46	1.96
6	16.00	14.98	1.02
7	18.00	17.29	0.71
8	18.50	19.37	0.87
9	19.50	19.75	0.25
10	19.00	19.3	0.30
11	16.00	16.67	0.67
12	17.50	16.21	1.29
13	20.50	21.97	1.47
14	19.00	19.69	0.69
15	16.50	16.11	0.39
Avg.			0.76

The average absolute difference error of the system optimized with SRP was calculated to be at ±0.76 kph which is relatively lower than that of without optimization at ±1.25 kph as shown in Table II and III. It can be inferred that the former has the capacity to calculate vehicle speed more accurately than the latter with respect to a standard.

B. *Statistical Treatment*

In the performance and accuracy test, paired t-test and linear regression analysis were applied to prove or disprove that performance and accuracy of vehicle speed calculation significantly improved in the application of SRP as optimization. They were selected because of the function of the former to statistically verify whether significant difference exists between two observations, and of the latter to verify whether significant correlation exists therewith.

In this study, the two observations correspond to the two setups: vehicle speed calculation without and with SRP. In the assumption where μ_1 is the mean data of the former and μ_2 is the mean data of the latter, then null hypothesis states that performance, in terms of effective frame rate, either declined or maintained in the application of SRP. On the other hand, the alternative hypothesis states that SRP improves the effective frame rate of the prototype significantly. The assumptions of the null and alternative hypothesis for the performance test are shown in (5) and (6).

$$H_0 : \mu_1 \geq \mu_2 \tag{5}$$

$$H_1 : \mu_1 < \mu_2 \tag{6}$$

Since it only tested if performance had improved, it was considered to be an upper-tailed test. If the calculated t is greater than or equal to $t_{critical}$ at $n-1$ df or if the $p\text{-value}$ is less than or equal the significance level, then the null hypothesis is rejected. A significance level of 0.05 was used for a sample size of 15 trials for each setup. The test results are shown in Table IV.

TABLE IV
ONE-TAIL T-TEST ON EFFECTIVE FRAME RATE

	Vehicle Speed Calculation without SRP	**Vehicle Speed Calculation with SRP**
Mean	3.29	7.08
Variance	0.66	1.32
t Stat	10.42	
t Critical One-tail	1.72	
p-value	7E-11	

Since t is greater than $t_{critical}$ and $p\text{-value}$ is less than 0.05 in the test on effective frame rate, then the null hypothesis is rejected. A significant difference exists between the effective frame rates of vehicle speed calculation without and with SRP that it can be stated in confidence that the latter significantly improves the performance of the system.

In linear regression analysis, the correlation coefficient, r, indicates the strength of the linear relationship between sets of data. A value of +1 signifies total positive correlation, 0 for no correlation, and -1 for total negative correlation. If the $p\text{-value}$ is less than or equal the significance level, then null hypothesis stating that r is not significantly different than 0 is rejected, thus there exists a significant linear relationship between the actual and calculated vehicle speed. A significance level of 0.05 was also used for a sample size of 15 trials for each setup. The test results are shown in Table V.

TABLE V
LINEAR REGRESSION ANALYSIS ON VEHICLE SPEED

	Vehicle Speed Calculation without SRP	**Vehicle Speed Calculation with SRP**
r	0.54124	0.91646
r Square	0.29294	0.83990
Standard Error	1.55748	0.66337
p-value	0.03719	0.00001

Since the $p\text{-value}$ of both systems are less than 0.05 in the test on vehicle speed, then the null hypothesis is rejected. A significant linear relationship between the actual and calculated vehicle speed exists that it can be stated in confidence that correlation is significant for both systems. It can be observed, however, that the value of r for vehicle speed calculation with SRP at 0.91646 is greater than that of without SRP at 0.54124, and that the $p\text{-value}$ of the former at 0.03719 is far lower than that of the latter at 0.00001. It can then be inferred that vehicle speed calculation with SRP exhibits better correlation and significance.

IV. CONCLUSIONS

In this study, a vehicle speed calculation system was developed on Raspberry Pi with Gaussian Mixture Model for vehicle detection and Kalman Filter for vehicle tracking. It was optimized with Sparse Random Projection in the effort to improve performance and accuracy by projecting the video to a low-dimensional subspace. In this way, the system can work with faster processing and more input without significant loss of data allowing it to calculate vehicle speed more accurately.

In the context of performance, vehicle speed calculation with SRP performed at an average effective frame rate of 7.08 fps which is more than twice that of the system without SRP at 3.29 fps. In the context of accuracy, the former has an absolute difference error of ±0.76 kph which is relatively lower than that of the latter at ±1.25 kph. In the application of paired t-test and linear regression analysis, it was concluded that SRP significantly improves the performance of vehicle speed calculation in terms of effective frame rate and that it exhibits better correlation and significance in terms of the linear relationship between the actual and calculated vehicle speed.

V. FUTURE WORKS

In the future, more general optimization techniques such as threading and parallelism can be integrated to improve the general performance of vehicle speed calculation; complex factors such as environment conditions can be accounted for with more powerful object detection and tracking algorithms in complement with more complex morphological operations; and advanced capabilities such as multiple detection, tracking and calculation considering traffic factors such as vehicle shadows and contiguity can be employed.

REFERENCES

[1] S. Bhatkar, M. Shivalkar, and B. Tandale, "Survey of Various Methods used for Speed Calculation of a Vehicle," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3 (Issue 3), pp. 1158-1161, 2015.

[2] I. Iszaidy, R. Ngadiran, R. B. Ahmad, M. I. Jais and D. Shuhaizar, "Implementation of raspberry Pi for vehicle tracking and travel time information system: A survey," *in 2016 International Conference on Robotics, Automation and Sciences (ICORAS)*, 2016, pp. 1-4.

[3] I. Iszaidy, A. Alias, R. Ngadiran, R. B. Ahmad, M. I. Jais and D. Shuhaizar, "Video size comparison for embedded vehicle speed detection & travel time estimation system by using Raspberry Pi," *in 2016 International Conference on Robotics, Automation and Sciences (ICORAS)*, 2016, pp. 1-4.

[4] R. K. C. Billones, A. A. Bandala, E. Sybingco, L. A. G. Lim and E. P. Dadios, "Intelligent system architecture for a vision-based contactless apprehension of traffic violations," *in 2016 IEEE Region 10 Conference (TENCON)*, 2016, pp. 1871-1874.

[5] J. D. S. Selda, R. M. R. Ellera, L. C. Cajayon II and N. B. Linsangan, "Plant identification by image processing of leaf veins," *in 2017 ACM International Conference Proceeding Series*, Part F131372, 2017, pp. 40-44.

[6] E. Komagal, A. Vinodhini, A. Srinivasan and B. Ekava, "Real time Background Subtraction techniques for detection of moving objects in video surveillance system," *in 2012 International Conference on Computing, Communication and Applications*, 2012, pp. 1-5.

[7] Basri, Indrabayu and A. Achmad, "Gaussian Mixture Models optimization for counting the numbers of vehicle by adjusting the Region of Interest under heavy traffic condition," *in 2015 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2015, pp. 245-250.

[8] M. Elad, M. A. T. Figueiredo and Y. Ma, "On the Role of Sparse and Redundant Representations in Image Processing," *in Proceedings of the IEEE*, vol. 98, no. 6, pp. 972-982, June 2010.

[9] E. Bingham, and H. Mannila, "Random Projection in Dimensionality Reduction: Applications to image and text data," *in Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 245-250.

[10] Y. Shen, Wen Hu, et al., "Real-Time and Robust Compressive Background Subtraction for Embedded Camera Networks," *IEEE TRANSACTIONS ON MOBILE COMPUTING*, vol. 15, no. 2, pp. 406-418, 2016.

[11] Y. Zhao, Z. Yuan, D. Chen, J. Lyu and T. Liu, "Fast Pedestrian Detection via Random Projection Features with Shape Prior," *in 2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Santa Rosa, CA, 2017, pp. 962-970.

[12] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," *in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 287-296.

[13] Indrabayu, Basri, A. Achmad, I. Nurtanio, and F. Mayasari, "Blob Modification In Counting Vehicles Using Gaussian Mixture Models Under Heavy Traffic," *ARPN Journal of Engineering and Applied Sciences*, pp. 7157-7163, 2015."

[14] Indrabayu, R. Y. Bakti, I. S. Areni and A. A. Prayogi, "Vehicle detection and tracking using Gaussian Mixture Model and Kalman Filter," *in 2016 International Conference on Computational Intelligence and Cybernetics*, 2016, pp. 115-119.

[15] V. Sulic, et al., "Efficient Dimensionality Reduction Using Random Projection," *Computer Vision Winter Workshop*, 2010, pp. 29-36.