Performance of Optimized CSK, DFT, and LOT for Video-based Container Tracking System using Simulated Annealing

Endra Joelianto, Basuki Rahmat, *Member, IAENG*, Parsaulian I. Siregar, Endang Juliastuti, and Steven Bandong

***Abstract*— This study considers the performance of three object tracking algorithms, namely Circulant Structure of Tracking-by-Detection with Kernels (CSK), Distribution Fields for Tracking (DFT), and Locally Orderless Tracking (LOT). These methods are used to track containers in five video recordings of container loading and unloading. At each video frame change, the three methods are implemented to detect and track moving containers. The comparison is aimed to evaluate the tracking performances of the three methods for this instance by means of seven performance indexes, i.e., Frame per Second (FPS), Sample Accuracy, Average IoU, Average IoG, Object Tracking Accuracy (OTA), Precision, and Recall. The experimental results are presented in terms of the average accuracy, IoU, IoG, OTA, Precision, and Recall of the sample. Based on the results, tracking methods using DFT are recommended over those using CSK or LOT, even though its average frame rate is slower compared with CSK. Optimization is applied to each method by using simulated annealing to find their optimal parameters. The results show that DFT and LOT perform best, while CSK is not able to track the containers. DFT especially yields better performance on four of the videos.**

***Index Terms*—CSK, DFT, LOT, object tracking, video tracking, simulated annealing.**

I. Introduction

THE container unloading system is an important element of goods transportation systems based on the sea lane. It maintains the quality of the distribution of goods, so they are dispatched on time to the next party. However, the loading and unloading process can be compromised when the control system does not operate accurately, precisely, and smoothly. This can result in the quality of goods decreasing or even damage to goods when they experience a strong enough shock. In addition, the distribution will be delayed, which will be detrimental to every party involved, from distribution chain members to end consumers. Increasing global trade adds to the urgency of this problem. Cargo transportation increases by six percent every year. Accordingly, container traffic also rises every year. For example, at Klaipeda port, container traffic increased by 11 percent in 2014 [1].

In manual loading and unloading of containers, various problems can be encountered due to lack of experience of the crane operator. For example, incorrect speed can cause the load to be damaged and threaten the safety of workers and equipment. Errors due to oscillations can hinder the process of changing direction [2]-[4]. As a result, ships have to stay longer in the port, resulting in higher container transportation costs. To make container transportation safer and faster, and to reduce transportation costs, container loading-unloading must be improved [1].

One way out of this problem is to implement a control system with sensors that provides accurate container monitoring and tracking. According to Yoshikawa et al. [5], the accelerometer, a commonly used position detector to overcome oscillations, is insufficiently accurate because it produces an error of 20 to 30 mm. Using a 2-dimensional PSD camera and an LED light produces an error of 10 mm. A position detector using a 2-dimensional CCD camera and laser light yields 4 mm accuracy, but its accuracy is reduced due to interference from sunlight and rain [5].

Several object tracking methods for designing container motion monitoring have been developed, for example CSK [6], DFT [7], and LOT [8]. Artificial intelligence methods have been applied, i.e., improved mean shift [9]-[12], fuzzy logic system [13]-[18], deep learning [19]-[25], and neural network [26]. The present study focused on tracking container movement by means of the CSK, DFT, and LOT techniques. The container to be tracked was a standard container used in ports, with dimensions 5.9 m × 2.34 m × 2.38 m. In this study, video images of containers were used, recorded during the daytime as well as the nighttime, with the container far removed as well as close by. This was done to determine the performance of the models under each of these conditions, which is the actual state of container

Manuscript received June 22, 2021; revised May 12, 2022. This work was supported by an Excellent University Research Grant, Ministry of Research and Higher Education 2018-2019, Indonesia, and partly funded by University Center of Excellence Artificial Intelligence on Vision, NLP and Big Data Analytics (U-CoE AI-VLB), 2020, Institut Teknologi Bandung, Indonesia.

E. Joelianto is an associate professor of the Instrumentation and Control Research Group, and University Center of Excellence Artificial Intelligence on Vision, NLP and Big Data Analytics, Institut Teknologi Bandung, Indonesia (phone: +62-22-250 4424; fax: +62-22-250 6281; e-mail: ejoel@tf.itb.ac.id).

B. Rahmat is an assistant professor of the Informatics Department, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Surabaya, Indonesia (e-mail: basukirahmat.if@upnjatim.ac.id).

P. I. Siregar is an assistant professor of the Instrumentation and Control Research Group, Institut Teknologi Bandung, Indonesia (e-mail: psiregar@tf.itb.ac.id).

E. Juliastuti is an associate professor of the Instrumentation and Control Research Group, Institut Teknologi Bandung, Indonesia (e-mail: yuliast@tf.itb.ac.id).

S. Bandong is a Ph.D. candidate of the Engineering Physics Doctoral Program, Institut Teknologi Bandung, Indonesia (e-mail: bandong.steven@gmail.com).

implementation in ports. This study further investigated each method by parameter variation and optimization based on a simulated annealing algorithm.

II. Circulant Structure Kernels (CSK) of Tracking-by-Detection

This section discusses the CSK technique from [6]. In this method, phase correlation is used as a sequential detector to track containers. First, classifier training is carried out by sampling the window in the position considered as the target location. A bounding box is applied to this first frame. The Fourier domain is used to estimate the classifier response in all locations. Then, the position with the best response is selected. This simple tracking and detection process are a strength of CSK because the computation process is very fast. First, the tracking process creates a window. Then, the cosine window is applied to reduce the discontinuity [27]:

$$x_{ij} = (x_{ij}^{raw} - 0.5)sin(\pi i / n)sin(\pi j / n), \forall i, j = 0,...,n-1$$

$$y_{ij} = \exp\left(\frac{-\left((i-i')^2 + (j-j')^2\right)}{\left(\upsilon\sqrt{mn}\right)^2}\right), \forall i, j = 0,...,n-1 \quad (1)$$

where x^{raw}, x and n are the original image, the converted image, and the image size $n \times n$, respectively. y, υ, and m are the classifier output, the spatial bandwidth parameter and the target size $m \times n$, respectively. (i, j) and (i', j') are the target location and the predicted location, respectively. Fourier transforms on overlapping windows in two neighboring frames are applied to calculate the classifier output, which is defined as:

$$y = F^{-1}(F(k) \Theta F(\alpha)) \quad (2)$$

where Θ, F and F^{-1} are the element-wise product, Fourier transform and its inverse, respectively. For each image x, in the form of an $n \times 1$ vector, its samples are expressed as [27]:

$$k_i = \kappa(z, P^i x), \quad \forall i = 0,....,n-1$$

$$k^{\text{gauss}} = \exp\left(-\frac{1}{\sigma^2}\left(\|x\|^2 + \|x'\|^2 - 2F^{-1}\left(F(x) \odot F^*(x')\right)\right)\right) \quad (3)$$

where z, κ, k_i and P^i are the object window, the kernel function, the elements of k, and the permutation matrix, respectively. The permutation matrix cyclically moves the image vector x by one element. The kernel used is a Gaussian kernel, as expressed in Eq. (3). The parameter α is trained as in Eq. (4), where λ is a constant given by:

$$\alpha = F^{-1}[F(y) / (F(k) + \lambda)]$$

$$\alpha_f = (1-\eta)\alpha_p + \eta\alpha \quad (4)$$

The first frame is used to compute the model α. To the next alpha frames (α_f), interpolation (η) is applied [27] by using the alpha values before α_p and the new alpha α.

The CSK method provides a closed-form solution for calculating various kernels in all image locations in an efficient way that fully utilizes the structure of the problem. The formula is precise and easy to calculate. In summary, the description of the CSK algorithm for video object tracking is expressed in the algorithm below and Fig. 1 [6].

Algorithm 1 CSK

Input: Image from the frame of video $I^{(n)}$
Output: Response y in the form of a Gaussian kernel

1. Extract a window using the input image by (1)
2. Assign a label to each sample
3. Calculate k_i and α using (3) and (4)
4. Determine the response (y) based on Fourier transform (2)
5. Use (4) to train a new model (α and x)

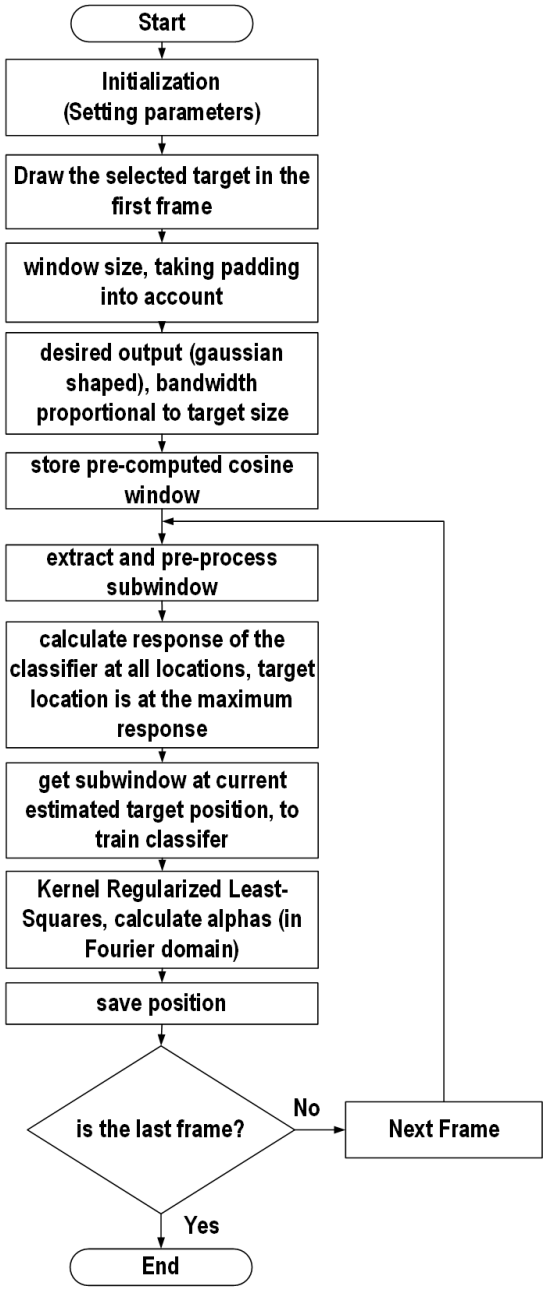


Fig. 1. CSK flowchart.

III. Distribution Fields for Tracking (DFT)

Visually following common objects generally assumes that a gradient descent in the alignment function will achieve the global optimum. The general procedure for achieving the criteria of the objective function is to blur the image. However, the blurred image will distort the information in the image, which could cause the target to become undetectable. One method to overcome this side effect is to construct an image descriptor using a distribution field (DF) representation, which allows cleaning the objective function without damaging any information concerning pixel values [7].

The Distribution Fields Tracking (DFT) method investigated in [7] conserves the object's spatial structure by creating a distribution of each pixel. A distribution field has $(2+N)$ dimensions. The width and height are the first two dimensions. The other N dimensions are the feature space.

If the image is in greyscale, then it has 3D feature spaces with size $m \times n \times b$ where m, n and b are the width, height, and intensity of the grey color, respectively. If the image has a higher dimensional feature, then it also has a higher DF. If the image is in RGB color, then the image will have five DF dimensions, i.e., the width, height, and color intensity for red, green, and blue of each pixel. Every element of the matrix is the probability of a pixel taking its attribute values. This creates a probability distribution as defined by Eq. (5) [28]:

$$df(i,j,k) = \begin{cases} 1 & if \quad I(i,j) == k \\ 0 & otherwise \end{cases}, \tag{5}$$

where i and j are the object location expressed by the row and column index, and k denotes the pixel values. Furthermore, k gives the sign to the k-th layer of the DFs. In each DF, the total of the elements in each column is 1, where each pixel creates a probability distribution.

Subsequent to the acquired initial DFs, both the image and the feature spaces will enter a smoothing process. In Eq. (6), a 2D Gaussian filter is applied to do convolution on each layer of the 3D DFs [28].

$$df_s(k) = df(k) * h\sigma_s \tag{6}$$

where k, $h\sigma_s$, and * are the index of the k-th layer of the DFs, a 2D Gaussian kernel with standard deviation σ_s, and the convolution operator, respectively. For a grayscale image, a 1D Gaussian filter is applied in Eq. (7) to do convolution on the 3 dimensions of the DFs [28].

$$df_{ss}(\mathrm{i},\mathrm{j}) = df_s(\mathrm{i},\mathrm{j}) * h\sigma_f, \tag{7}$$

where the 1D Gaussian kernel with standard deviation σ_f is denoted by $h\sigma_f$. In order to integrate the total of each column of the DFs to 1, uniform distributions fill up the missing information outside the boundaries [7].

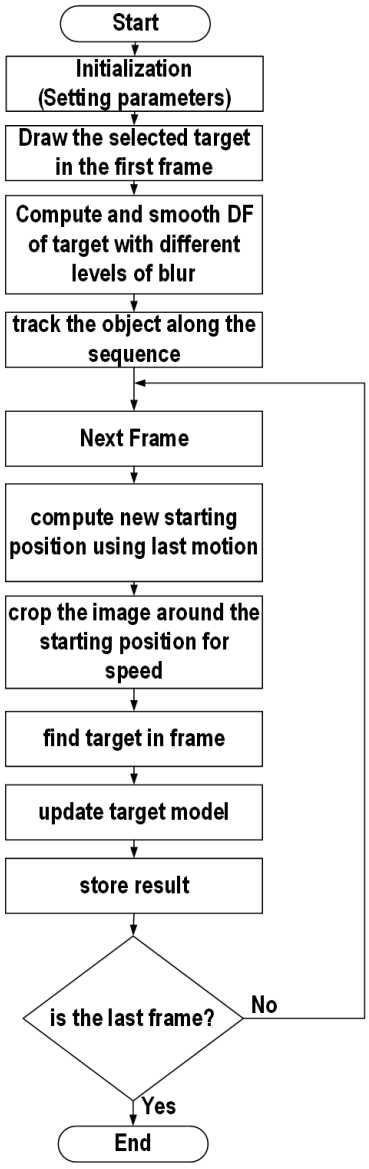


Fig. 2. DFT flowchart.

A hierarchical approach is applied to improve performance. Small sets of DFs are used to represent the target, where each is DF constructed by applying ever-increasing parameter values that govern the size of the spatial blur. These DFs contain information on various frequencies. In each frame, a coarse-to-fine approach is used. The most refined DF is utilized to start searching until it accomplishes a local minimum. This position then becomes the beginning of searching in the second DF. The DFT algorithm for video object tracking is briefly stated in Algorithm 2. Here, h_s is a 2D Gaussian filter built with σ_s, while h_f is built with σ_f. A brief description of the DFT algorithm is given in the flowchart in Fig. 2.

Algorithm 2 DFT

Input: V = video sequence

- I = patch comprising target in frame 1
- σ_s = spatial smoothing parameter set
- σ_f = brightness smoothing parameter

- b = brightness bin number (b = 16)
- λ = mixing parameter (λ = 0.95)

Output: $(x, y)_f$ {target positions at each frame f in V }

1: Initialization of $d^i_{model} = explode(I) * h_{s(i)} * h_f, i \in 1,..|\sigma_s|$

2: Initialization of target location (x, y) to center of patch I

3: **for** f = 2 $\rightarrow|V|$ **do**

4: **for** I =1 $\rightarrow|\sigma_s|$ **do**

5: $d^i_f = explode(I) * h_{s(i)} * h_f$

6: $(x', y') = \underset{(x,y)}{\operatorname{argmin}} L_1(d^i_f(x, y), d^i_{model})$

7: $(x, y) = (x', y')$

8: **end for**

9: $d_{model} = \lambda d_{model} + (1-\lambda) d_f(x, y)$

10: **end for**

IV. Locally Orderless Tracking (LOT)

The LOT technique was first introduced in [8]. LOT is a method for tracking and detecting algorithms that calculate the number of local interferences on an object. The tracker can focus on a deformable object directly without prior assumptions. This provides a probability model of the object's movements over time, using the earth mover's distance with two parameters to organize pixels and color shifting. These parameters are adjusted during tracking.

LOT operates Locally Orderless Matching (LOM) in order to the track object. LOT applies the Bayesian method with particle filtering (PF), where the probability of a particular particle coming from the object being tracked is concluded using LOM. Algorithm 3 shows how it works.

The probabilistic noise for one pixel $P_r(p \mid q, \Theta)$ needs to be defined first before defining the probability between patches $P_r(P \mid Q, \Theta)$. The standard distance of EMD is calculated by $d(p, q) = -log(p \mid q, \Theta)$. Then, by resolving the EMD problem, $P_r(P \mid Q, \Theta)$ can be calculated.

Super pixels are used instead of raw pixels to reduce the burden of computation in calculating EMD. Super pixels can be obtained using Region of Interest (ROI). Super pixels generate signatures that are a representation of the target and candidate patches. There are M clusters in a signature, whose locations, i.e., mass geometric center and average appearance, give a representation of each [8].

The target's state at each frame is acquired using PF. For every instance of N particles, a signature is built. Then, between each of the signatures, $\{P_k\}_{k=1}^N$, EMD is computed, as well as the target signature, Q_0, with standard distances. Then, the weights can be calculated using the EMD scores $\{EMD_k\}_{k=1}^N$ in the formula $\pi_k = e^{-\beta . EMD_k}$. The weighted sum of all particles is then known as the new target state [8].

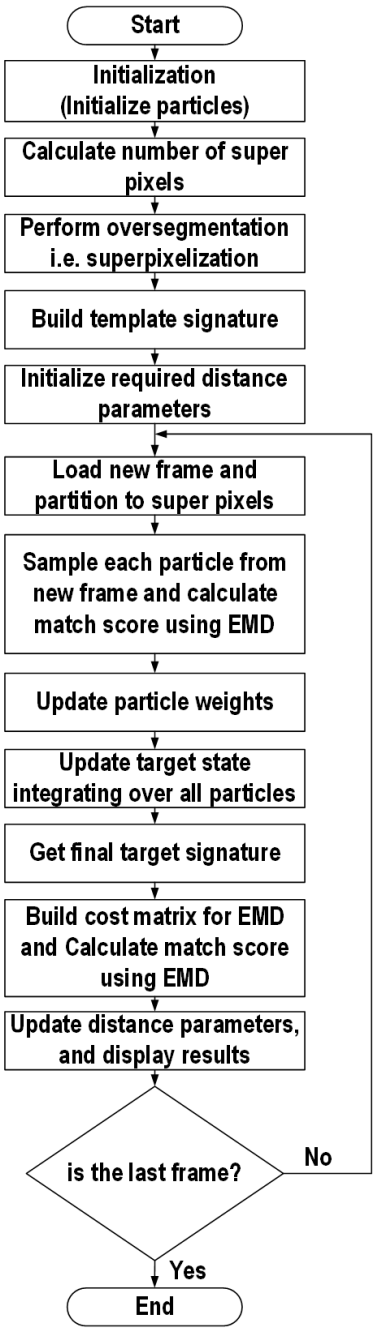


Fig. 3. LOT flowchart.

Using the flow concerning the final candidate signature and the target signature, Q_0, the noise distribution parameters Θ_{ML} are calculated, which is regulated by Gaussian noise model $\Theta_{Prior} = \{\sigma_{A^{prior}}, \sigma_{L^{prior}}\}$ and a moving average (MA) parameter (α_{MA}). Each element of the Gaussian noise model has weight $w_{Prior} = \{w_{\sigma_A^{prior}}, w_{\sigma_L^{prior}}\}$. Then the final parameters Θ_n can be computed with [8]. A brief description of the LOT algorithm is given in the flowchart in Fig. 3.

$$\Theta_{MAP} = \frac{\Theta_{ML} + \Theta_{Prior}.w_{Prior}}{1 + w_{Prior}}$$
$$\Theta_n = (1-\alpha_{MA}).\Theta_{n-1} + \alpha_{MA}.\Theta_{MAP} \quad (8)$$

Algorithm 3 LOT

Input: Frame $I^{(n)}$, target signature $Q_0 = \{q_i, w_i^q\}_{i=1}^{M_{Q_0}}$, noise parameters $\Theta^{(n-1)}$, particle states $\{X_i^{(n+1)}\}_{i=1}^{N}$

1. Partition ROI in $I^{(n)}$ into super pixels I_{SP}
2. For each particle $X_k^{(n)}$ do:
   (a) Form signature $P_k = \{p_i^k, w_i^{p^k}\}_{i=1}^{M_{P_k}}$ using I_{SP}
   (b) Compute standard distances using:
   $\{d_k\}_{ij} = d(p_i^k, q_j) = -log(p_i^k \mid q_j, \Theta^{(n-1)})$
   (c) Calculate $EMD_k \leftarrow EMD(P_k, Q_0, d_k)$
   (d) Particle weight calculation $\pi_k = e^{-\beta.EMD_k}$
3. Weight normalization s.t. $\sum_{i=1}^{N} \pi_i$
4. Compute new target position $X_{Target}^{(n)} = \sum_{i=1}^{N} \pi_i X_i^{(n)}$
5. Form a new target signature P_T, then calculate EMD flow $f_{i,j} \leftarrow EMD(P_T, Q_0, d_T)$
6. Update parameters $\Theta^{(n)}$ according to (8)
7. Draw particles $\{X_i^{(n+1)}\}_{i=1}^{N}$

V. Simulated Annealing

Simulated annealing is a non-linear optimization method inspired by a metallurgical process called annealing. Annealing is the process of cooling and hardening hot metal, which is initially liquid until it reaches a solid state. Simulated annealing begins with the creation of test points obtained randomly based on a probability distribution that depends on the current temperature [33]. If the latest point has a better cost function, then it will become the next point. However, if the latest point of the cost function is worse than the previous point, then that point will be rejected unless it accomplishes the acceptance function as stated in Eq. (9).

$$P(\text{Accept } S_c) = \frac{1}{1 + \exp\left(\frac{\Delta}{\max(T)}\right)}, \quad (9)$$

where Δ is the difference between the new and the previous cost functions, and T is the current temperature as a candidate for the next point. A lower temperature value indicates a smaller probability. The algorithm will systematically decrease the temperature as the iteration increases and at the same time save the point that has the best cost function. The temperature of the next iteration is calculated with Eq. (10). T_0 is the initial temperature and k is the annealing parameter [33].

$$T = T_0 \times 0.95^k \quad (10)$$

VI. Proposed Methodology

This study compared three methods, i.e. CSK, DFT, and LOT, to assess their ability to track moving objects in video recordings. The focused object of tracking is the movement and orientation of containers that are in the video recordings. Containers in video recordings sometimes have different colors and orientations while moving. We used different and more complex performances than only the spatio-temporal context for contrasted object tracking [29].

The process was divided into three stages. The first stage was conducted by applying the default parameters of the three methods to track containers in order to compare the tracking performance of the default parameters. The default parameters can be considered to describe the general capabilities of the detector. The second stage was conducted to investigate the effect of changing the parameters of each method on the performance of the tracker. Parameter variation can be done as a reference for further research for the development of tracking algorithms specifically in the field of container loading and unloading automation. The best tracking method was studied further to identify the optimum parameters of the methods in improving container tracking accuracy. The third stage was the optimization of CSK, DFT and LOT using a simulated annealing algorithm.

In the first stage, video recordings of containers with different orientations and distances were used for container tracking. To achieve the objectives of this study, five video recordings were collected with containers under different lighting conditions and shooting angles. From each video, four frames were sampled with their respective detection results from the three methods used. Each detection result was calculated to assess the accuracy of its capability in tracking containers, so that an accuracy value was obtained for each method on four frames from each video. In addition, to measure the computational speed of each method, the frame rate was calculated in frames per second (FPS), i.e., the speed of the object tracking process across all video frames. The frame rate represents the speed at which sequential images turn up on the screen. The proposed methodology in this paper is shown in Fig. 4 in the form of a flowchart.

Intersection over Union (IoU) was performed on each sample result. An object is called detected if the value of intersection over the union in the sample exceeds a predefined threshold. The threshold is usually selected as 0.5, or 50%. Eq. (11) shows how to calculate the IoU [30]:

$$IoU = \frac{\text{area}(B_p \cap B_{gt})}{\text{area}(B_p \cup B_{gt})} \quad (11)$$

where B_p and B_{gt} are the predicted and the ground truth bounding box, respectively. The comparison of the intersection area to the ground truth area is a parameter that can be used to measure how well an object is detected. It is defined as Intersection over Ground Truth (IoG), which can be calculated by using Eq. (12).

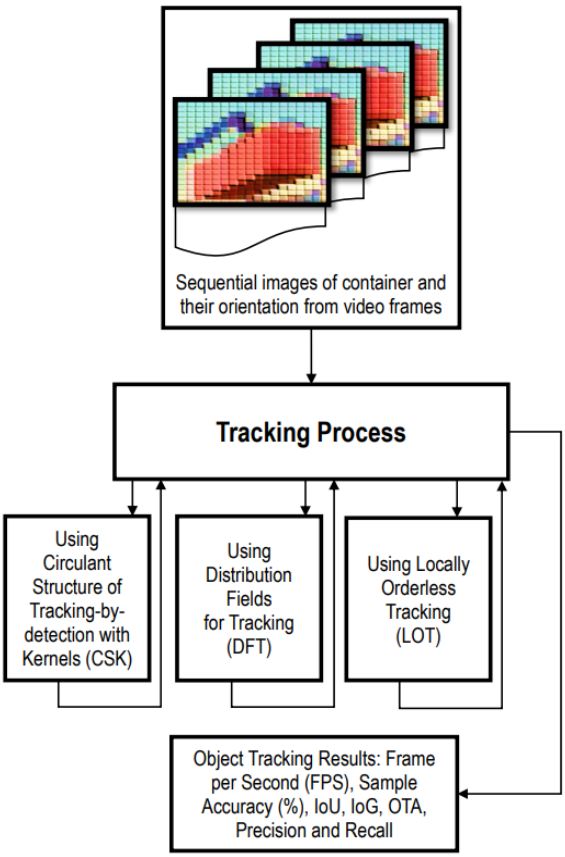


Fig. 4. Tracking process flowchart.

$$IoG = \frac{\text{area}(B_p \cap B_{gt})}{\text{area}(B_{gt})} \tag{12}$$

Object Tracking Accuracy (OTA) is a method to evaluate the tracking performance of a single object detector. The numbers of false positives (n_{fp}^i) and false negatives (n_{fn}^i) are used to compute the tracking performance.

$$OTA = 1 - \frac{\sum_i (n_{fn}^i + n_{fp}^i)}{\sum_i g^i}, \tag{13}$$

where g^i is the number of the ground truth bounding box in frame *i*. OTA indicates to what extent the result overlaps with the true bounding box [31]. An OTA value of 1 indicates that the tracker has successfully detected the object and 0 indicates failure.

Precision is the ability of the model to recognize only relevant objects. This is the positive predictive percentage, which is given by the following equation:

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{TP}{\text{all detections}} \tag{14}$$

Recall is the ability of a model to recognize all the basic truths. This is the percentage of true positive detection of all ground truths.

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{TP}{\text{all ground truths}} \tag{15}$$

where TP (true positive) is the number of times the number of detections is correct, FP (false positive) is when the detection is false, and FN (false negative) is when the ground truth is not detected [32].

In the second stage, the same five videos were used as data to track containers. However, variations of the parameters of the CSK, DFT, and LOT methods were applied. Six variations of the parameters were utilized for each method. The method with the best accuracy was tested further to obtain the parameter values that will provide the highest accuracy. In CSK, the regularization parameter λ, as in Eq. (4), was varied in the range 0.00001 to 1. In DFT, the parameter λ, see line 9 in Algorithm 2, was varied in the range 0.65 to 1. In LOT, the parameter β, see line 2d in Algorithm 3, was varied in the range 0.01 to 100. The details of the applied parameter variations are given in Table I.

TABLE I
CONTAINER TRACKING PARAMETER VARIATION

CSK (λ)	DFT (λ)	LOT (β)
0.00001	0.65	0.01
0.0001	0.75	0.1
0.001	0.85	1
0.01	0.95	10
0.1	0.98	50
1	1	100

VII. RESULTS AND DISCUSSIONS

CSK, DFT, and LOT were applied and evaluated on five videos recorded during the process of moving containers in a port [34]-[37]. The three methods were used to track containers in every frame of the five videos. The results of the application of CSK, DFT, and LOT techniques on four frames taken form five videos each to evaluate the accuracy of the methods. The results of tracking the five recorded video containers are shown in Figs. 5 to 9. Table 2 and Fig. 10 provide a summary of the tracking results.

Based on the results of the test with Video 1, shown in Fig. 5, the CSK and LOT methods did not succeed in detecting the containers. Meanwhile, the DFT method managed to track 3 out of 4 samples, or 75%. DFT had an IoU of 63.7% and an IoG of 70.2%, both exceeding the 50% threshold. It had an OTA of 0.75. The tracking speed for the CSK method was quite high at 17.5041 fps in comparison with DFT at 0.37314 fps and LOT at 0.11785 fps.

Based on the results of the test with Video 2, Fig. 6 shows that the CSK method was unable to detect the containers. The DFT method was successful in tracking four out of four samples, or 100%, while the LOT method managed to track two out of four samples, or 50%. DFT had IoU 78.6%, IoG 97.7%, and OTA 1 higher than CSK and LOT. In this case, LOT had an IoU of 65.0% and an IoG of 80.9%, both exceeding the threshold. On the other hand, CSK had an IoU of 0% and an IoG of 0%. The tracking speed of CSK was the highest, 56.3764 fps, but this method did not detect the container correctly. The speed of the two other methods was 0.4816 fps for DFT and 0.16312 fps for LOT.

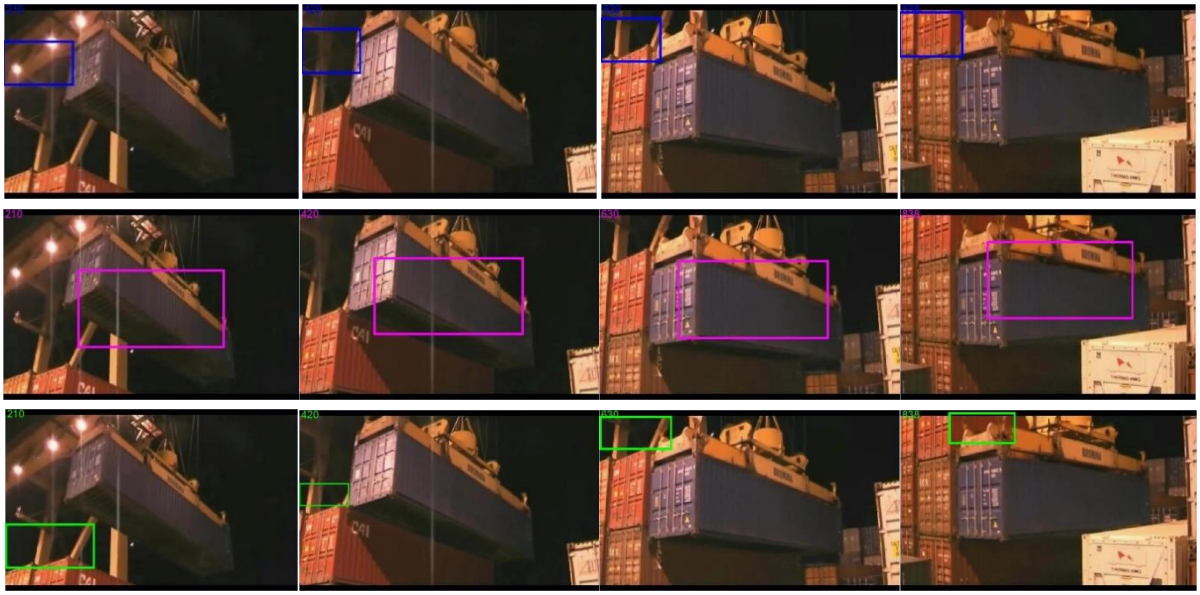


Fig. 5. Video 1 – container tracking with CSK, DFT and LOT (from the top), frames 210, 420, 630, 838 taken from [34].

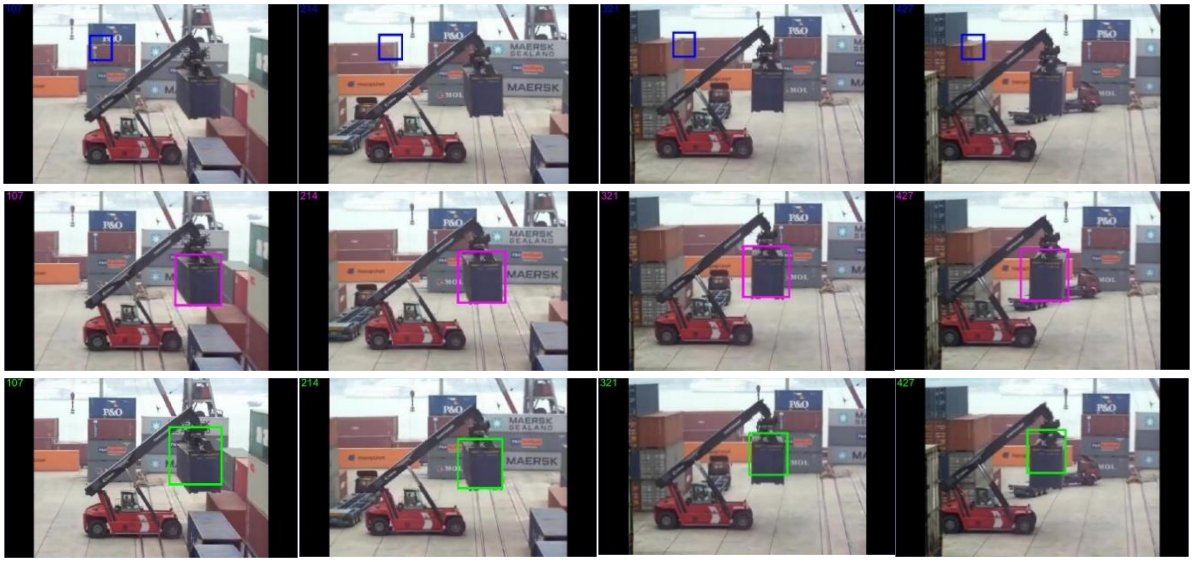


Fig. 6. Video 2 – container tracking with CSK, DFT and LOT (from the top), frames 107, 214, 321, 427 taken from [35].

Fig. 7. Video 3 – container tracking with CSK, DFT, and LOT (from the top), frames 324, 648, 972, 1294 taken from [36].

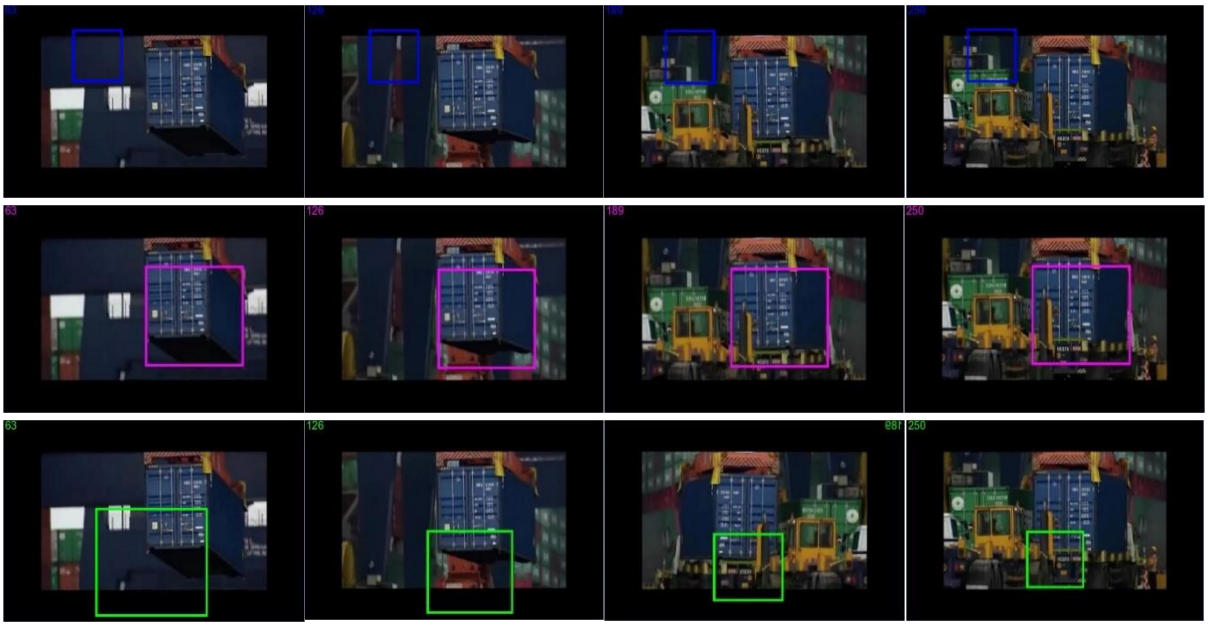


Fig. 8. Video 4 – container tracking with CSK, DFT and LOT (from the top), frames 63, 126, 189, 250 taken from [37].

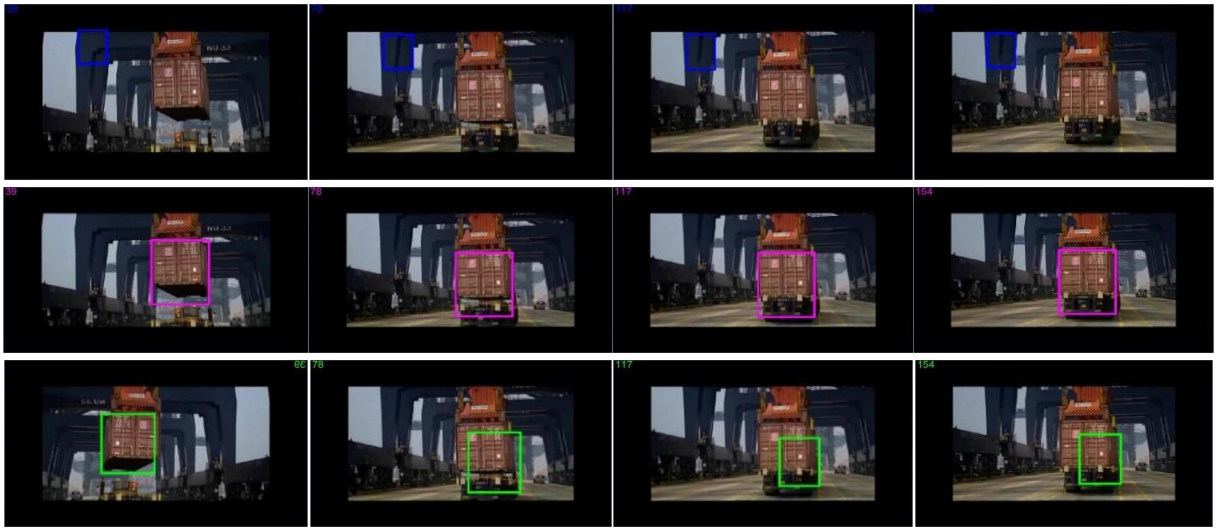


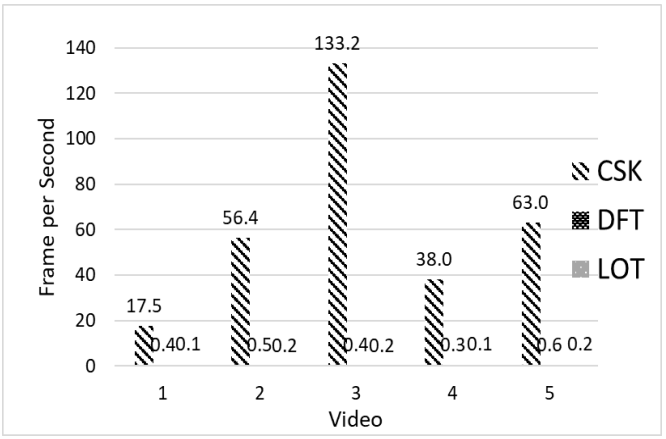
Fig. 9. Video 5 – container tracking with CSK, DFT and LOT (from the top), frames 39, 78, 117, 154 taken from [37].

TABLE II
RESULTS OF OBJECT TRACKING

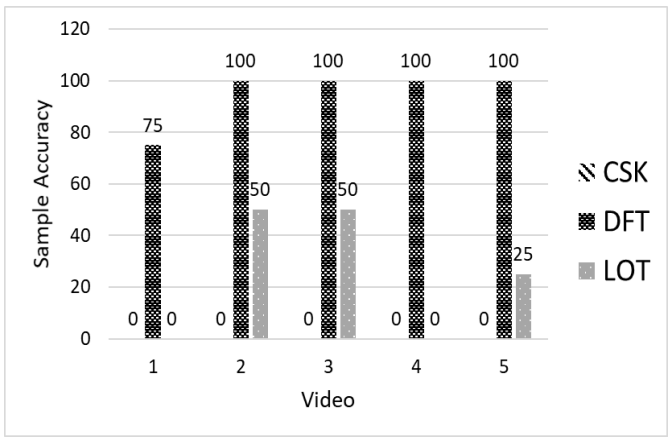
Video	Number of frames	CSK							DFT							LOT						
		a	b	c	d	e	f	g	a	b	c	d	e	F	g	a	b	c	d	e	f	g
1	838	17.5041	0	0.31	0.37	0	0	0	0.37314	75	63.7	70.2	0.75	1	1	0.11785	0	0	0	0	0	0
2	427	56.3764	0	0	0	0	0	0	0.48160	100	78.6	97.7	1	1	1	0.16312	50	65.0	80.9	1	1	1
3	1294	133.2383	0	0	0	0	0	0	0.42658	100	87.7	100	1	1	1	0.22524	50	39.7	39.8	0.5	0.5	0.5
4	250	38.0220	0	0	0	0	0	0	0.31667	100	78.6	89.4	1	1	1	0.10255	0	17.3	24.6	0	0	0
5	154	62.9757	0	0	0	0	0	0	0.60777	100	75.7	96.1	1	1	1	0.15882	25	50.9	61.3	0.25	0.25	0.25
Average		61.6233	0	0.1	0.1	0	0	0	0.44120	95	76.9	90.7	0.95	1	1	0.15350	25	34.58	41.32	0.35	0.35	0.35

where:

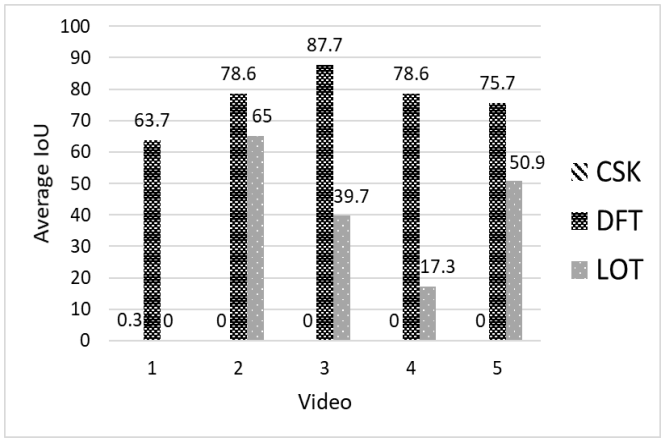
a: Frames per second (FPS)
b: Sample Accuracy (%)
c: Average IoU (%)
d: Average IoG (%)
e: OTA
f: Precision
g: Recall



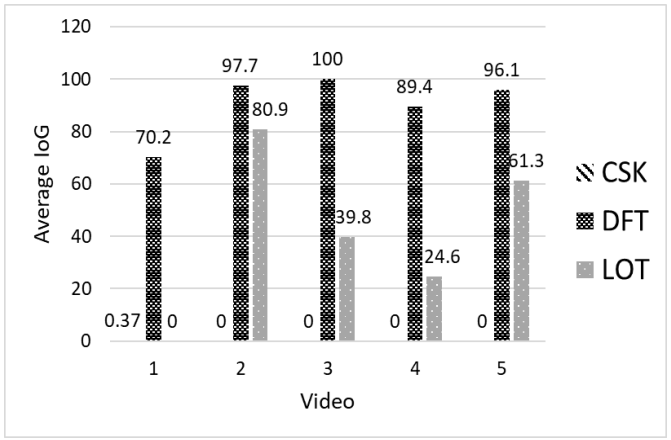
a. Frame rate per second



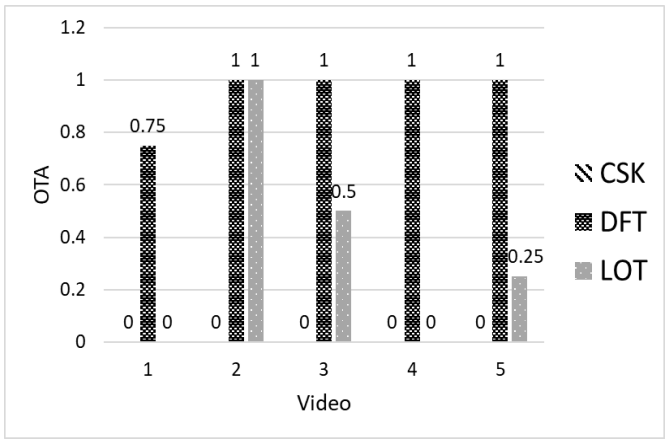
b. Sample accuracy



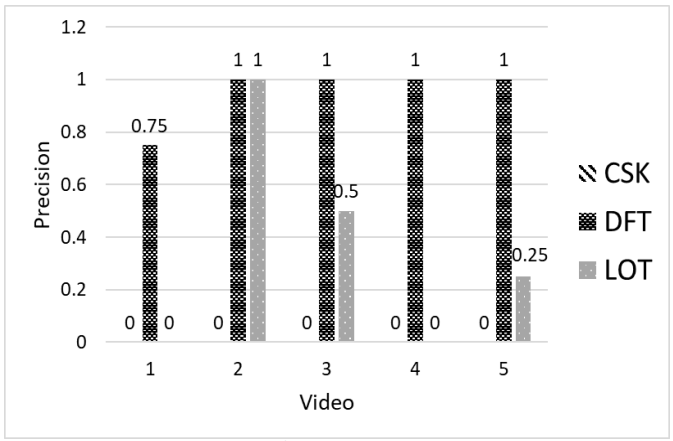
c. Average IoU



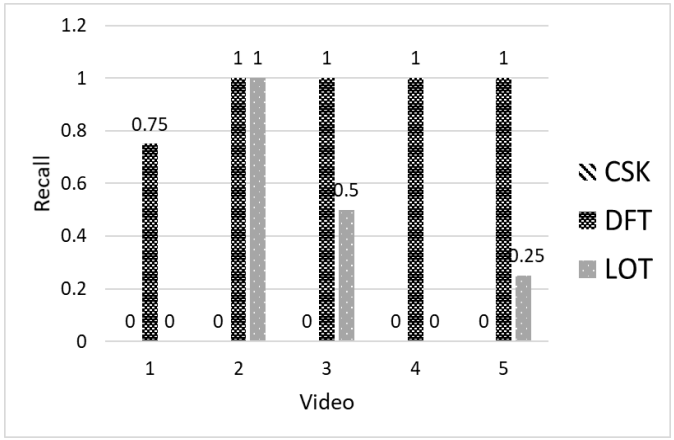
d. Average IoG



e. OTA



f. Precision



g. Recall

Fig. 10. Comparison of object tracking results.

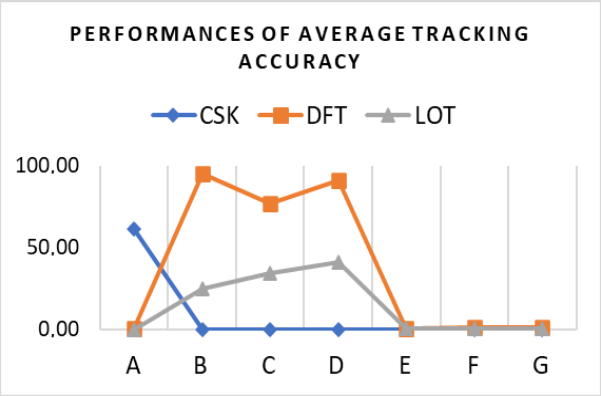


Fig. 11. Comparison of average tracking performances.

Fig. 7 shows the outcomes for the test with Video 3. The CSK method was not able to detect the containers. The DFT method was successful in tracking four out of four samples, or 100%. Meanwhile, the LOT method managed to track two out of four samples, or 50%. DFT had IoU 87.7%, IoG 100%, and OTA 1 higher than CSK and LOT. In this case, LOT could not detect the containers since IoU 39.7% and IoG 39.8% did not exceed the threshold. CSK had IoU 0%, IoG 0% and OTA 0. The tracking speed of the CSK method was the fastest at 133.2383 fps. The other methods were slower (DFT 0.42658 fps and LOT 0.22524 fps) but were better in terms of detection accuracy.

The results for the test with Video 4 can be seen in Fig. 8. The CSK and LOT methods were not successful in detecting and tracking the containers. The DFT method was successful in tracking four out of four samples, or 100%. DFT had IoU 78.6%, IoG 89.4%, and OTA 1 higher than CSK and LOT. LOT could not detect the containers, with IoU at 17.3%, IoG at 24.6%, and OTA 0 not exceeding the threshold. CSK had IoU 0%, IoG 0% and OTA 0. CSK's tracking speed was again the highest at 38.022 fps, followed by 0.31667 fps for DFT and 0.10255 fps for LOT.

The test results for Video 5 can be seen in Fig. 9. The CSK method was not successful in detecting and tracking the containers. The DFT method was successful in tracking four out of four samples, or 100%, while the LOT method managed to track one out of four samples, or 25%. DFT had IoU 75.7%, IoG 96.1%, and OTA 0 higher than CSK and LOT. LOT could detect the containers in this case, because IoU was 50.9% and IoG was 61.3%, exceeding the threshold. CSK had 0% IoU and 0% IoG. The CSK method had the fastest container tracking speed at 62.9757 fps, followed by DFT at 0.60777 fps and LOT at 0.15882 fps.

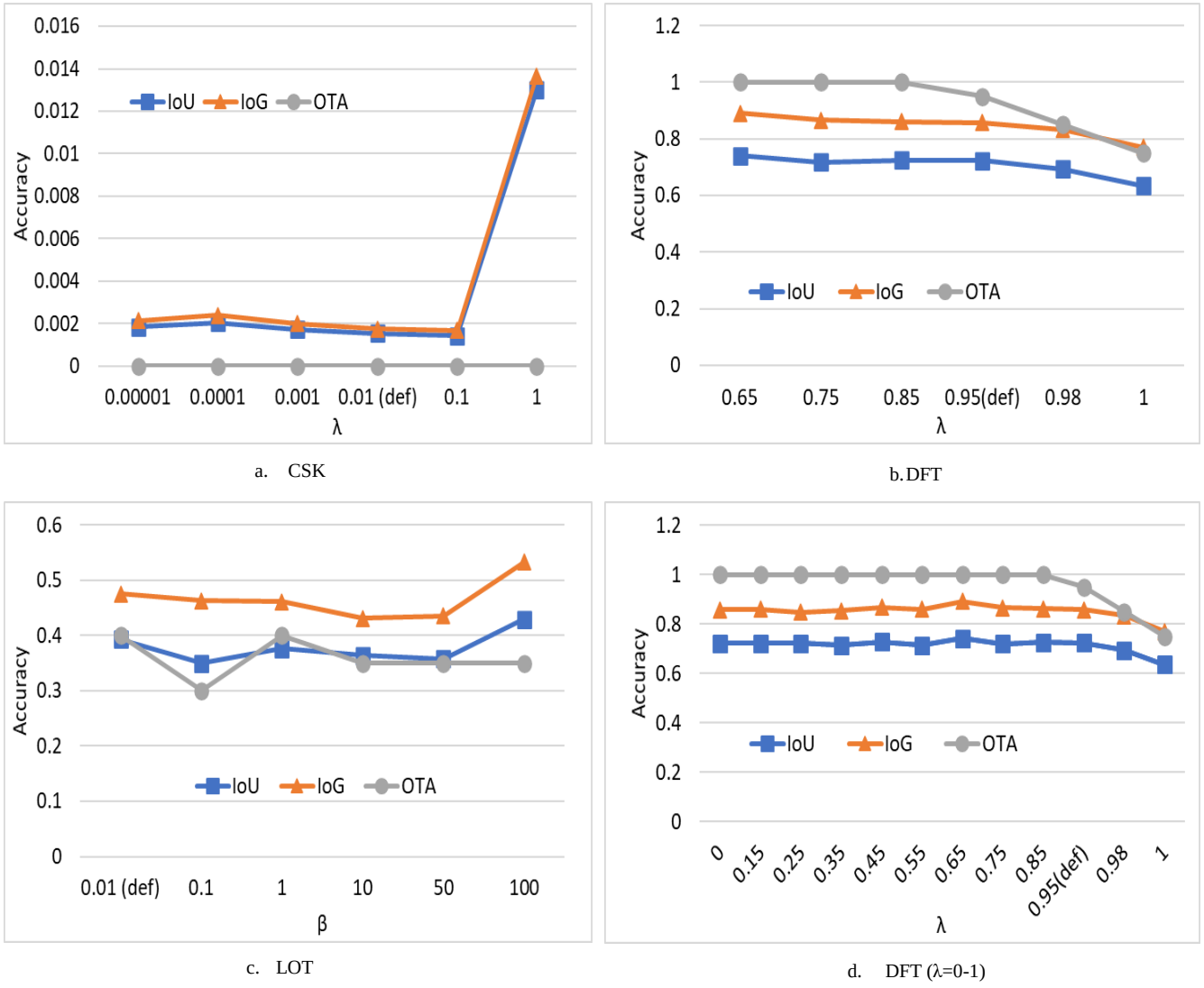
Fig. 10 presents the accuracy of tracking in terms of seven performance measures. Fig. 11 summarizes the tracking accuracy of each performance by means of the average tracking accuracy of each measure. The results show that DFT had superior performance compared with CSK and LOT on six different accuracy performance measures. However, the average frame rate performance of DFT was lower than that of CSK and slightly higher than that of LOT.

In the second stage, the parameters values in CSK, DFT, and LOT were varied, as shown in Table 1. The performance of each container tracking method after parameter variation can be seen in Fig. 12. The results of varying parameter λ in the CSK method did not produce a significant change in tracking accuracy, as it was very low when compared to LOT and DFT. The highest average IoU and IoG values were obtained at a value of 0.014 (see Fig. 12a). This means that the prediction of the container bounding box only covered a very small area of the ground truth and sometimes also gave an IoU of 0, which means that the bounding box did not intersect with the ground truth. On the basis of the capability to produce true positives, none of the containers was successfully tracked based on the OTA value, which was 0 at each variation of the λ value. Therefore, CSK is considered unable to track containers properly.

LOT produced better performance than CSK. LOT successfully detected the containers in one or two frames of the five videos. This can be seen from the OTA value, which ranged from 0.30 to 0.40 (see Fig. 12c). In addition, the IoG values ranged from 0.45 to 0.55 and the IoU values ranged from 0.35 to 0.45, which indicates that LOT succeeded in detecting a container but the container tracking bounding box did not completely coincide with the ground truth; about 0.5 of the ground truth did not fit into the container bounding box predictions. Tests with various values for parameter β did not show a significant increase in tracking accuracy, as indicated by the IoU, IoG, and AOT values, which were in the range of 0.30 to 0.55. Parameter value β = 100 gave a slight increase in IoU and IoG, but it was not significant. This tracking performance is too low to be considered good, but it did yield the highest IoU and IoG values. This means that testing variation of parameter β at higher values to improve tracking accuracy could be an interesting topic for a future study. As its accuracy was not too high, LOT was not tested further in this study.

Variation of the λ value in DFT led to better performance than CSK and LOT. This was indicated by OTA values in the range of 0.75 to 1, IoU values in the range of 0.63 to 0.72, and IoG values in the range of 0.77 to 0.87 (see Fig. 12b). An OTA value of 1 indicates that in each test frame, DFT successfully detected the container. The application of variations of the value showed a decrease in IoU, IoG, and OTA when the value got closer to 1. The value of λ regulates the proportion of combinations between the initial model and the new observation results. A value of λ = 1 means that DFT only uses the initial model, while a value of λ = 0.98 means that DFT only uses 0.02 information from new observations. This causes a decrease in DFT performance when the value of λ approaches 1.

This confirms the importance of information from the latest observations in each frame. The default parameter λ = 0.95 was less than optimal, as shown in the graph, as there is a decrease in OTA. Therefore, to find the optimal parameter value of λ, it is necessary to test it over a wider range. In addition, this method can be used to assess how the initial model and new observations contribute to tracker performance. Thus, the value of λ was further varied in the range from 0 to 1. The results are shown in Fig. 12d. The OTA, IoU and IoG values did not experience a significant decrease when λ was close to 0. This shows that new observational data play a very important role in forming accurate tracking. Parameter variation led to obtaining the optimal parameter for container tracking. The value of λ = 0.65 gave higher IoU and IoG values than the others.



a. CSK

b. DFT

c. LOT

d. DFT (λ=0-1)

Fig. 12. Performance comparison of varied container tracker parameter values (CSK, DFT and LOT).

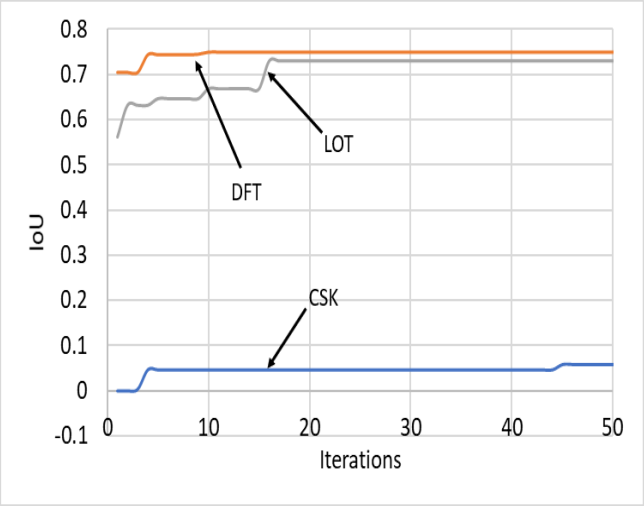


Fig 13. Optimization of CSK, DFT and LOT using simulated annealing.

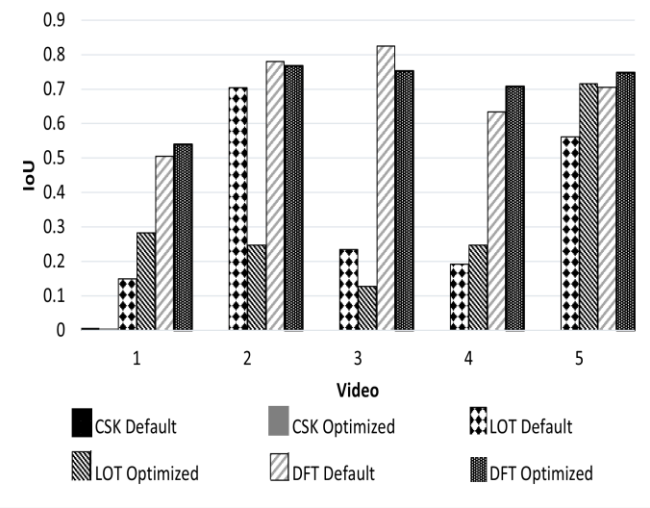


Fig 14. Comparison of CSK, DFT and LOT optimization using simulated annealing.

Fig. 15. Video 1 – container tracking with optimized CSK, DFT and LOT (from the top), frames 210, 420, 630, 838 taken from [34].

Fig. 16. Video 2 – container tracking with optimized CSK, DFT and LOT (from the top), frames 107, 214, 321, 427 taken from [35].

Fig. 17. Video 3 – container tracking with optimized CSK, DFT, and LOT (from the top), frames 324, 648, 972, 1294 taken from [35].

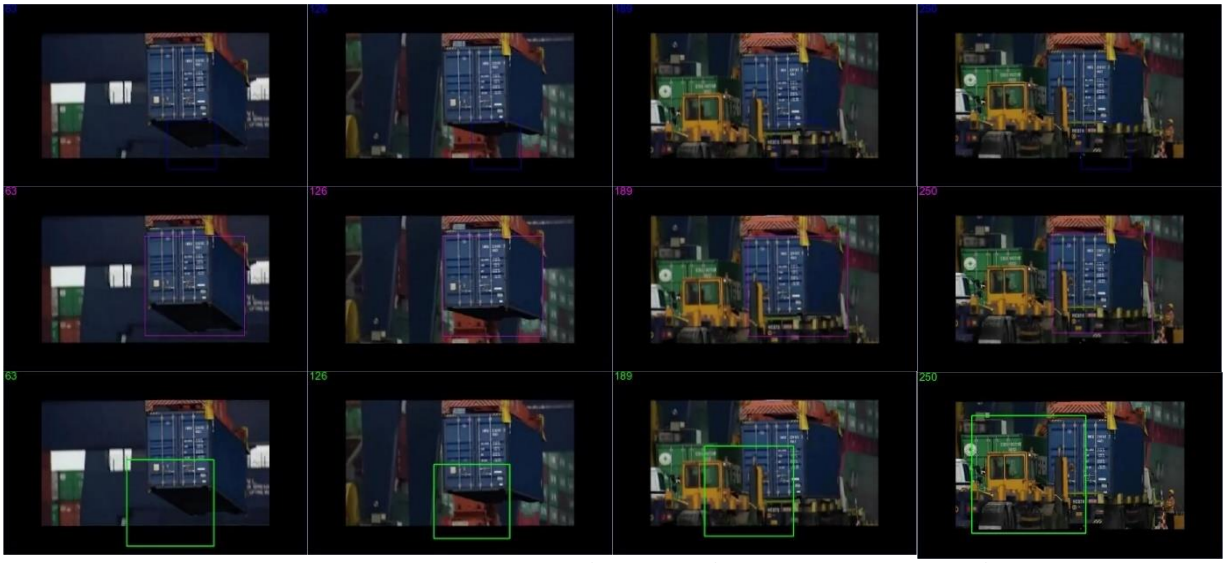


Fig. 18. Video 4 – container tracking with optimized CSK, DFT and LOT (from the top), frames 63, 126, 189, 250 taken from [36].

Fig. 19. Video 5 – container tracking with optimized CSK, DFT and LOT (from the top), frames 39, 78, 117, 154 taken from [36].

TABLE III
CSK OPTIMIZATION DETAIL

Parameter	Optimization range (min-max)	Default	Optimization Results
υ (1)	0-1	1/16	0.01407
σ (3)	0-1	0.2	0.70405
λ (4)	0-1	0.01	0.40183
η (4)	0-1	0.075	0.90600
IoU			0.05684

TABLE IV
DFT OPTIMIZATION DETAIL

Parameter	Optimization range (min-max)	Default	Optimization Results
h_{σ_f} (alg.2.1)	0-1	0.625	0.32581
h_{σ_s} (alg.2.1)	0-10	1	8.12897
h_{σ_s} (alg.2.5)	0-10	2	7.88552
λ (alg.2.9)	0-1	0.95	0.48185
IoU			0.748052

TABLE V
LOT OPTIMIZATION DETAIL

Parameter	Optimization range (min-max)	Default	Optimization Results
α_{MA} (8)	0-1	0.3	0.036139
$\sigma_{A^{prior}}$ (8)	0-1	0.05	0.989002
$\sigma_{L^{prior}}$ (8)	0-1	0.1	0.967029
$w_{\sigma_A^{prior}}$ (8)	0-1	0.25	0.969137
$w_{\sigma_L^{prior}}$ (8)	0-1	0.25	0.015084
IoU			0.730843

Simulated annealing was applied to obtain optimal models for CSK, DFT, and LOT. First, simulated annealing was applied using the default value as optimization reference or initial temperature, and 50 iterations of searching for optimal parameters. Optimization was applied to Video 5 because it had the shortest duration so that the optimization computation took less time. The optimization objective value was IoU, which was calculated for four frames in Video 5. The optimization details of each method

along with the search parameter range are described in the following.

The optimization results of the CSK, DFT, and LOT methods can be seen in Tables III to V and Figs. 13-19. The range of parameter values was taken around the default value of the related parameters. Parameters whose default values were between 0 and 1 were given an optimal parameter search area in the range of 0 to 1. Parameters whose values were 1 or higher were assigned a search area in the range of 0 to 10.

The optimization results from 0 to 50 iterations can be seen in Fig 13. The optimization results showed that CSK did not show a significant increase in IoU, but DFT and LOT did show an increase in IoU when optimized. LOT showed an increase from the default value of 0.56 to 0.71 and DFT from 0.71 to 0.75. This shows that simulated annealing was successful in improving the container tracking performances of DFT and LOT.

The optimal parameters obtained from optimization of container tracking on Video 5 were then tested on the other four videos. The results are shown in Fig. 14. CSK generally failed to track the containers with the default parameters. The optimized DFT showed better performance, except on Video 3, where the default tracking value was better. LOT optimization led to better tracking performance on three videos. The results with the default parameters were more accurate on Video 2 and Video 3. It can be concluded that apart from DFT providing more accurate tracking accuracy than CSK and LOT, the obtained optimal parameters also showed better accuracy and more stable performance than the default parameters.

VII. Conclusion

This paper discussed the influence of parameter variation of the CSK, DFT, and LOT techniques for container tracking using video images in the process of loading and unloading at the port. Based on the test results on five different videos (with object tracking capabilities measured in terms of IoU, IoG, OTA, precision, and recall), the DFT tracking approach is recommended as the best method compared to CSK and LOT, even though DFT has a lower average frame rate than CSK. Further study was done by testing the change of performance of the three methods when the parameter values were varied. CSK was unable to detect the containers and LOT did not show a significant improvement. Variation of parameter λ gave better performance compared to the default value. Further optimization based on a metaheuristic approach by using simulated annealing was also done. The optimized DFT had better and more stable performance compared to the optimized CSK and LOT.

References

[1] T. Eglynas, M. Bogdevicius, A. Andziulis, and T. Lenkauskas, "Overview of crane control system and the related problems: Analysis of container oscillation using different types of cargos," in *2015 Mechanics, Material Science and Industrial Engineering*, vol. 6, pp. 642-648, 2015.

[2] Y. Yoshida and K. Tsuzuki, "Visual tracking and control of a moving overhead crane load," in *9th IEEE International Workshop on Advanced Motion Control*, Istanbul, Turkey. IEEE, pp. 630–635, 2006.

[3] M. Vukosavljev and M. E. Broucke, "Control of a gantry crane: A reach control approach," in *53rd IEEE Conference on Decision and Control*, Los Angeles, CA. IEEE, pp. 3609–3614, 2014.

[4] S. Bruins, "Comparison of different control algorithms for a gantry crane system," *Intelligent Control and Automation*, vol. 1, no. 2, pp. 68-81, 2010.

[5] H. Yoshikawa, S. Kunimitsu, H. Hoshina, N. Miyata, and M. Kobayashi, "Development of hoisting load position sensor for container handling cranes," *Mitsubishi Heavy Industries, Ltd. Technical Review*, vol. 38, no. 2, 2001.

[6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Computer Vision - ECCV 2012*, pp. 702-715, 2012.

[7] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1910-1917, 2012.

[8] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan, "Locally orderless tracking," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1940-1947, 2012.

[9] B. Rahmat, E. Joelianto, I. K. E. Purnama, and M. H. Purnomo, "An improved mean shift performance using switching kernels for Indonesia vehicle license plate tracking video," *Internetworking Indonesia Journal*, vol. 10, no. 2, pp. 49-56, 2018.

[10] B. Rahmat, E. Joelianto, I. K. E. Purnama, and M. H. Purnomo, "An improved mean shift using adaptive fuzzy gaussian kernel for Indonesia vehicle license plate tracking," *IAENG International Journal of Computer Science*, vol. 45, no. 3, pp. 458-471, 2018.

[11] Y. Yang, X. Wang, J. Wu, H. Chen, and Z. Han, "An improved mean shift object tracking algorithm based on ORB feature matching," in *27th Chinese Control and Decision Conference (2015 CCDC)*, pp. 4996-4999, 2015.

[12] N. Li, D. Zhang, X. Gu, L. Huang, W. Liu, and T. Xu, "An improved mean shift algorithm for moving object tracking," in *Electrical and Computer Engineering (CCECE), 2015 IEEE 28th Canadian Conference on*, pp. 1425-1429, 2015.

[13] S. Zhang, W. Lu, W. Xing, and L. Zhang, "Using fuzzy least squares support vector machine with metric learning for object tracking," *Pattern Recognition.*, vol. 84, pp. 112-125, 2018.

[14] L. Liang-qun, Z. Xi-yang, L. Zong-xiang, and X. Wei-xin, "Fuzzy logic approach to visual multi-object tracking," *Neurocomputing*, vol. 281, pp. 139-151, 2018.

[15] H. Lee and A. Banerjee, "Non-rigid body object tracking using fuzzy neural system based on multiple ROIs and adaptive motion frame method," in *2009 IEEE International Conference on Systems, Man and Cybernetics*, pp. 3871-3876, 2009.

[16] S. B. Kang and T. K. Yang, "Tracking for moving object using adaptive Rao-Blackwellized particle filter and fuzzy control system," in *2007 International Conference on Control, Automation and Systems*, pp. 420-424, 2007.

[17] S. M. Mirhassani, B. Yousefi, and M. J. R. Fatemi, "Fuzzy based foreground background discrimination for probabilistic color-based object tracking," in *2009 5th IEEE GCC Conference Exhibition*, pp. 1-5, 2009.

[18] P. Chiranjeevi and S. Sengupta, "Rough-set-theoretic fuzzy cues-based object tracking under improved particle filter framework," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 3, pp. 695-707, 2016.

[19] X. Qian, L. Han, Y. Zhang, and M. Ding, "An object tracking method using deep learning and adaptive particle filter for night fusion image," in *2017 International Conference on Progress in Informatics and Computing (PIC)*, pp. 138-142, 2017.

[20] C. Phanikrishna and A. V. N. Reddy, "Contour tracking based knowledge extraction and object recognition using deep learning neural networks," in *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pp. 352-354, 2016.

[21] C. Özer, F. Gürkan, and B. Günsel, "Object tracking by deep object detectors and particle filtering," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pp. 1-4, 2018.

[22] X. Qian, L. Han, Y. Wang, and M. Ding, "Deep learning assisted robust visual tracking with adaptive particle filtering," *Signal Process. Image Commun.*, vol. 60, pp. 183-192, 2018.

[23] A. Brunetti, D. Buongiorno, G. F. Trotta, and V. Bevilacqua, "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey," *Neurocomputing*, vol. 300, pp. 17-33, 2018.

[24] P. Li, D. Wang, L. Wang, and H. Lu, "Deep visual tracking: Review and experimental comparison," *Pattern Recognition*, vol. 76, pp. 323-338, 2018.

[25] A. R. Pathak, M. Pandey, and S. Rautaray, "Application of deep learning for object detection," *Procedia Computer Science*, vol. 132, pp. 1706-1717, 2018.

[26] Y. Kim and Y. S. Kim, "Optimizing neural network to develop loitering detection scheme for intelligent video surveillance systems," *International Journal of Artificial Intelligence*, vol. 15, no. 2, pp. 30-

39, 2017.
[27] A. Osman and M. Krystian, "Online Learning and Detection with Part-based Circulant Structure," *22nd International Conference on Pattern Recognition*, 2014.
[28] N. Jifeng, S. Wuzhen, Y. Shuqin, and Y. Paul, "Visual tracking based on Distribution Fields and online weighted multiple instance learning," *Image and Vision Computing*, vol. 31, pp. 853-863, 2013.
[29] Z. Wang, B. Chen, J. Wu, and T. Yan, "Real-time image tracking with an adaptive complementary filter," *IAENG International Journal of Computer Science*, vol. 45, no 1, pp. 97-103, 2018.
[30] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303-338, 2010.
[31] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442-1468, 2013.
[32] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PloS One*, vol. 10, no. 3, p.e0118432, 2015.
[33] L. Van, J. M. Peter, and H. L. Emile, "Simulated annealing," in Simulated annealing: Theory and applications, pp. 7-15. *Springer*, Dordrecht, 1987.
[34] thunfisch1967. Container laden und laschen auf einem Frachter [Video]. YouTube.https://www.youtube.com/watch?v=F55Zcvo0F8k
[35] automos. Reach Stacker Unload Container. [Video]. Veoh. https://www.veoh.com/watch/v14221373552qBkWD6
[36] automos. Reach stacker load container to truck. [Video].Veoh. https://www.veoh.com/watch/v1422137372PAm7kEW
[37] Agmachine. Automatic Container Landing Systems (ACLAS) [Video].YouTube.https://www.youtube.com/watch?v=dKo2IF6lfaA

Endra Joelianto received a bachelor's degree in Engineering Physics from Institut Teknologi Bandung (ITB), Indonesia in 1990. He received Ph.D. in Engineering, from the Australian National University (ANU), Australia in 2002. Currently, he is an associate professor of the Instrumentation and Control Research Group, Faculty of Industrial Technology, Institut Teknologi Bandung, Indonesia and research staff at Centre for UnManned System Studies (CentrUMS), Center for Defense and Security Technology, Artificial Intelligence Center, National Center for Sustainable Transportation Technology (NCSTT), ITB, Indonesia. His research interests are hybrid/discrete event control systems, advanced control, embedded control systems, artificial intelligence, and intelligent systems. Dr. Joelianto is a senior member of IEEE.

Basuki Rahmat received a bachelor's degree in Instrumentation Physics from Institut Teknologi Sepuluh Nopember Surabaya in 1995. He received a master's degree in Instrumentation and Control from Institut Teknologi Bandung, Indonesia in 2000. He received a doctoral degree in Electrical Engineering at Institut Teknologi Sepuluh Nopember, Surabaya. He is an assistant professor of the Informatics Department, Universitas Pembangunan Nasional "Veteran" Jawa Timur. His research interests are intelligent systems, soft computing, image and video processing, and intelligent control.

Parsaulian I. Siregar received a bachelor's degree from Institut Teknologi Bandung (ITB), Indonesia. He received a Dr.-Ing. degree in Engineering from Technische Universitat Carolo-Wilhemina zu Braunschweig, Germany. Currently, he is an assistant professor of the Instrumentation and Control Research Group, Faculty of Industrial Technology, Institut Teknologi Bandung, Indonesia. His research interests are system identification and control.

Endang Juliastuti received the bachelor's degree in engineering physics from Institut Teknologi Bandung (ITB), Indonesia in 1978. She received a Dr. degree in Engineering, from Institut Teknologi Bandung (ITB), Indonesia in 2004. Currently, she is an associate professor of the Instrumentation and Control Research Group, Faculty of Industrial Technology, Institut Teknologi Bandung, Indonesia. Her research interests are optical instrumentation, image processing and 3D imaging.

Steven Bandong received a bachelor's degree from Universitas Sam Ratulangi (UNSRAT), Indonesia in 2015 and a master's degree from Institut Teknologi Bandung (ITB), Indonesia in 2019. Currently, he is a Ph.D. candidate of the Engineering Physics Doctoral Program, Faculty of Industrial Technology, Institut Teknologi Bandung, Indonesia. His research areas are energy systems and intelligent systems.