ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Improving the Performance of CNN by Using Dominant Patterns of CNN for Hand Detection

Natthariya Laopracha[1] and Kaveepoj Bunluewong[2]

ABSTRACT

Many applications have used hand gestures for software interaction, image- and video-based action analysis, and behavioral monitoring. Hand detection is an essential step in the pipeline of these applications, and Convolutional Neural Networks (CNN) has provided superior solutions. However, CNN has similar features between hand and non-hand images, called non-dominant features. These features affect miss-classifications and long-time computation. Therefore, this paper focuses on the selection of dominant CNN features for hand detection, and it is our proposed method (DP-CNN) that selects the dominant feature patterns (DP) from the trained CNN features and classifies them using the Extreme Learning Machine (ELM) method. Evaluation results show the proposed method (DP-CNN-ELM), which can increase the accuracy and the F1-score of CNN. In addition, the proposed method can reduce the time computation of CNN in training and testing.

Article information:

Keywords: Hand Detection, Convolution Neuron Network, Dominant Patterns, Extreme Learning Machine

Article history:
Received: January 6, 2023
Revised: April 1, 2023
Accepted: June 3, 2023
Published: June 24, 2023
(Online)

DOI: 10.37936/ecti-cit.2023172.251265

1. INTRODUCTION

Nowadays, applications using the human hand are being developed for applications such as computer interaction, virtual reality, driver behavior analysis, and human activity analysis. Human hand detection plays an important role in communicating through these applications, in which two computer vision-based steps are prominent. These include (1) hand detection and (2) hand gesture recognition. Real-world environments present unknown and variable conditions that affect hand detection performance. These consist of complex backgrounds, changing illumination, blurred image, and variations in hand shapes. These problems also affect hand gesture recognition and ultimately present difficulties in correctly classifying the communicated intentions to the application.

Traditional computer vision-based hand detection methods use several visual feature extraction methods such as skin color [1], histogram of oriented gradients (HOG) features [2], Harr-Like features [3], texture [4], and shape features [5]. Sofiane et al.[6] proposed hand detection using skin color and hand verification by palm properties, which is sensitive to noise from the input device. Mittal et al.[7] proposed using model HOG features for hand-shape detection [8] and hand classification using a support vector machine (SVM) [9]. This method's performance is poorer in images with non-uniform backgrounds. Chen et al. [10] proposed Haar-like features for hand detection and hand gesture recognition under various illumination and uniform laboratory background conditions. Their method produces high performance in their laboratory test dataset. Misra et al. [11] present hand detection using sparse texture and color-texture features. The runtime execution of their method is faster when compared to Gabor segmentation-based fractal texture analysis. Qi et al. [12] proposed hand-shape features for hand detection in a human-robot interaction application under three environmental conditions: conference room, laboratory room, and outdoors. The method shows higher accuracy when compared to a HOG-SVM classifier method; however, it is less than LeNet-5 [13] and ResNet-18 [14] in some situations. These traditional vision hand detection methods have limited capacity to represent the human hand concept across factors such as varied illumination conditions, complex backgrounds, hand

[1,2] The authors are from the Department of Computer Science; Faculty of Informatics; Mahasarakham Uninversity; Mahasarakham, Thailand, E-mail: natthariya@gmail.com and kaveepoj.b@msu.ac.th
[2] The corresponding author: kaveepoj.b@msu.ac.th

shapes, and hand rotations.

CNN-based methods are becoming popular for hand detection [15][16] due to their better than traditional methods in a real-world environment. Iglesias et al. [17] proposed the R-CNN method for the detection of hand gestures using an optimized R-CNN architecture (a small number of layers) based on the Darknet [18]. Their proposed method is suitable for computationally limited devices. Neethu et al. [19] proposed a CNN for hand gesture detection and recognition. Their method used masked images for hand detection and segmentation of fingers by a connected component analysis algorithm. The segmented finger regions of the images were input into their CNN for classification. Su et al. [20] proposed a system using augmented reality that recognized hand gestures and finger-pointing interactions from the perspective of the wearer's smart glasses. A region-CNN (R-CNN) based on skin-color masks gives high performance in hand detection. However, the model was negatively affected by changes in illumination. Roy et al. [21] proposed CNN-based skin color segmentation models for hand detection. Their skin segmentation method reduced the occurrences of hand detection misclassifications of benchmark R-CNN and Faster R-CNN models. Yang et al.[22] proposed a light-CNN network for detecting different-sized hands by generating feature maps at various resolutions. Their experimental results show high hand detection performance at different hand sizes and scales. Deng et al. [23] built a common (two-task) framework for hand detection (segmentation) and hand rotation (plane) estimation using a CNN method. Their common methods slightly improved average precision performance over their separated method, indicating that rotation feature information benefited their hand detection classification.

Research works on CNN-based hand detection have improved performance by addressing the runtime complexity of the convolution layer designs, incorporating skin color segmentation, and combining rotation estimation features into the hand detection (segmentation) CNN model. The results of these works give a high average precision in a range of about 50-94 [16] [24]. However, these methods have complex computations and require expensive GPUs, which can make users increase the cost.

This paper proposed reducing costs by improving simple computations in CNN. The interesting problem of CNN is the redundancy of features in feature maps found in the convolutional layer. These features are a major contributor to the high computational requirements In addition, the fully-connected characteristics of CNN use batch backpropagation neuron networks, which affect the complexity of the computation. Therefore, the proposed method uses the dominant pattern (DP) algorithm selecting features because its high-speed computation is highly accurate and uses ELM as a classifier [26]. This work uses ELM because the ELM gives high performance for deep hybrid CNN applications for image classification [27], [28]. In addition, ELM enables high performance and fast computation. This proposed method is called DP-CNN-ELM, which is a simple structure and can improve the performance of CNN in hand detection.

This paper is organized of the rest as follows: Section 2 reviews the underlying methods involved in the hand detection technique, while Section 3 presents the proposed method. The datasets and evaluation methods followed by experiment results are presented in Sections 4 and 5, respectively. The conclusions and intentions for future work are found in Section 6.

2. RELATED WORKS

This section describes the Convolution Neural Network (CNN) and the dominant pattern (DP), which these methods used in this paper.

2.1 Convolutional Neural Network (CNN)

CNN [29][30] is a neural network for describing image features used in object detection and image classification. The CNN architecture has two stages: feature learning and classification. The feature learning stage consists of convolution layers and a pooling layer, while the classification stage has fully-connected layers. Figure 1 shows the CNN architecture.

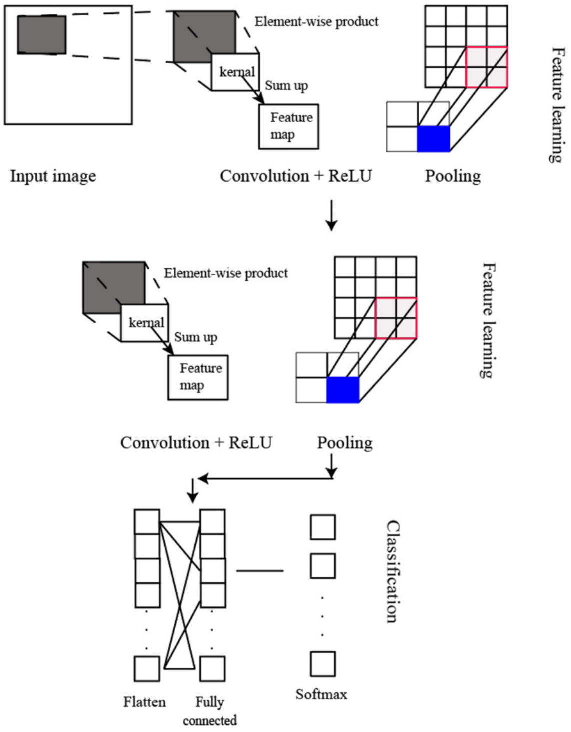


Fig.1: *CNN architecture.*

The convolution layer is the main component of CNN that performs feature extraction from the images. A kernel window (e.g., a filter or feature detector) slides across the image to multiply each image pixel (of the window region), element-wise, by the kernel matrix. The result of each full-image kernel convolution is a feature map. The pooling layer reduces the dimensions of the feature map to improve its robustness to small changes. Pooling computation types include maximum and average values.

Within the classification stage, the input from the feature learning stage is flattened to one dimension. The features are classified in the fully-connected layer. This layer can be used as an input into other machine learning models. Such examples are support vector machine (SVM) [31], random forest (RF) [32], and extreme learning machine (ELM)[33].

2.2 Selecting feature by their Dominant Patterns (DP)

DP [26] was proposed for selecting the dominant patterns of HOG in vehicle detection tasks. The underlying principle of DP is that object images belonging to different classes should have different dominant pattern subsets.

1) Training set preparation

In this step, the dataset is divided into two sets: the object set and the background set. The object set and background set consist of one-dimensional features of the object (O) and background (B), respectively, per image, as shown in equation (1). t is the number of images in the object set, d is the number of images in the background set, and m is the length of the feature vector. For example, in this proposed method, a_{t1} to a_{tm} is the one-dimensional trained CNN-feature values corresponding to image t, where that image is known to contain a pixel region with a hand *object*.

$$O = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ a_{t1} & a_{t1} & a_{t3} & a_{tm} \end{bmatrix}_{t\times m}, \quad B = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ a_{d1} & a_{d1} & a_{d3} & a_{dm} \end{bmatrix}_{d\times m} \tag{1}$$

2) Ideal vector computation

This step computes the features representing the object set and background set which are then determined as an *ideal vector*. The ideal vector of object $\bar{O}$ and background $\bar{B}$ are computed by equations (2) and (3).

$$\bar{O} = \left[\frac{1}{t}\sum_{i=1}^{t} O_{i,1}, \frac{1}{t}\sum_{i=1}^{t} O_{i,2}, \ldots, \frac{1}{t}\sum_{i=1}^{t} O_{i,m}\right]_{1\times m} \tag{2}$$

$$\bar{B} = \left[\frac{1}{d}\sum_{i=1}^{d} B_{i,1}, \frac{1}{d}\sum_{i=1}^{d} B_{i,2}, \ldots, \frac{1}{d}\sum_{i=1}^{d} B_{i,m}\right]_{1\times m} \tag{3}$$

3) Dominant patterns detection

This step segments the ideal vector into chunks of l-consecutive elements for computing the dominant patterns and non-dominant patterns. The dominant patterns are selected in this step. The number of segments is $K = m/l$. Each k^{th} segment has computed the difference between the ideal vector of the object and the background ($\bar{d}_k$). Then, the average of the differences between the object and background (D) has computed. Lastly, the dominant pattern corrects in the set I. Equations (4), (5), and (6) are computations $\bar{d}_k$, D, and I, respectively:

$$\bar{d}_k = \frac{\sum_{i=(k-l)l+1}^{kl} |\bar{O}_{(l,i)} - \bar{B}_{l,i}|}{l} \tag{4}$$

$$D = \frac{\bar{d}_1 + \bar{d}_2 + \bar{d}_3 + \cdots + \bar{d}_k}{K}, K = \frac{m}{l} \tag{5}$$

$$I = \{k | \bar{d}_k \geq (D + \alpha)\} \tag{6}$$

Empirical studies reported in [24] produced high accuracy rates by assigning the parameter values of l as 5, 10, 15, 20, 25, and 30; and α as -0.006, -0.003, 0.0, and 0.001. Intuitively, l and α parameters are the number of features, and the minimum threshold difference of the corresponding features are the values between the two targets. Importantly, both affect the accuracy and time computation in the classification stage.

2.3 Regions with Convolution Neuron Networks (R-CNN)

R-CNN is popular in hand detection because it can detect small images. R-CNN has two stages of computation. Firstly, the training stage is about learning ground truth images. These images are extracted features by CNN, and the features are fed into a SVM for computing the best model. Secondly, the detection stage is about detecting hands from images or videos. Each image is taken from an extraction region by a selective search algorithm [34] which consists of 2000 proposed regions. Then, these regions are fed into a CNN that produces a feature vector. Finally, classification features such as hand or non-hand are processed by the SVM, which uses the best model from the R-CNN training stage. Figure 2 shows the stages of R-CNN.

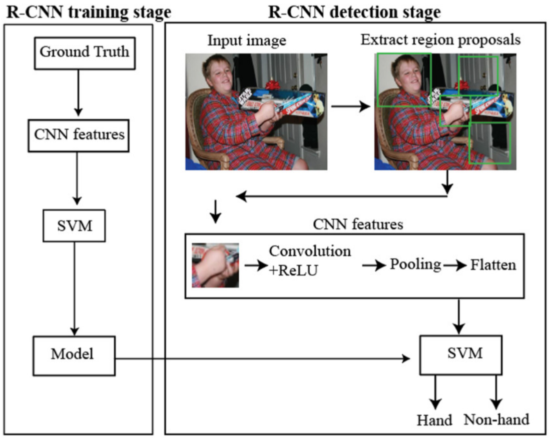


Fig.2: *CNN architecture.*

3. PROSED METHODS

This paper proposes human hand detection using DP-CNN with ELM as a classifier. The traditional computation of CNN consists of three layers: the convolutional layer, the pooling layer, and the fully-connected layer. Figure 3 shows the problem states of CNN. It considers the features produced by CNN that consist of dominant and non-dominant patterns. Figure 3(a) shows these features. The y-axis and x-axis of Fig. 3(a) are values of CNN features and the length of CNN features, respectively, in which Fig. 3 shows the length as 11. The sets of these features are patterns. Those dominant feature patterns can

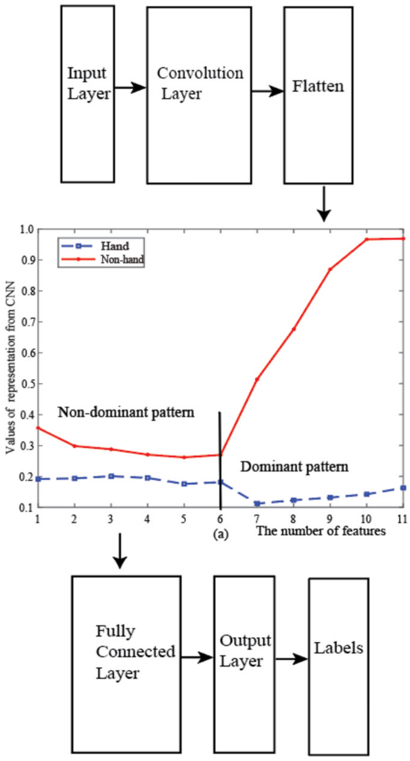


Fig.3: *The Problem State of CNN Features.*

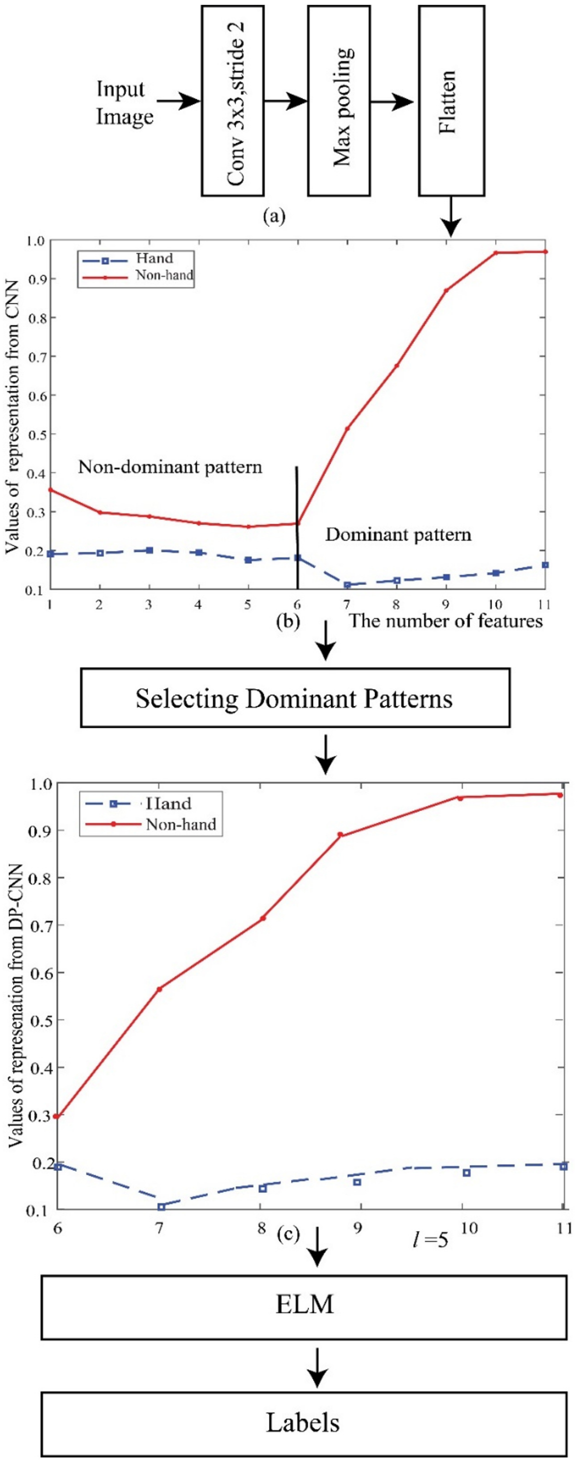


Fig.4: *Main Idea of the Proposed Method.*

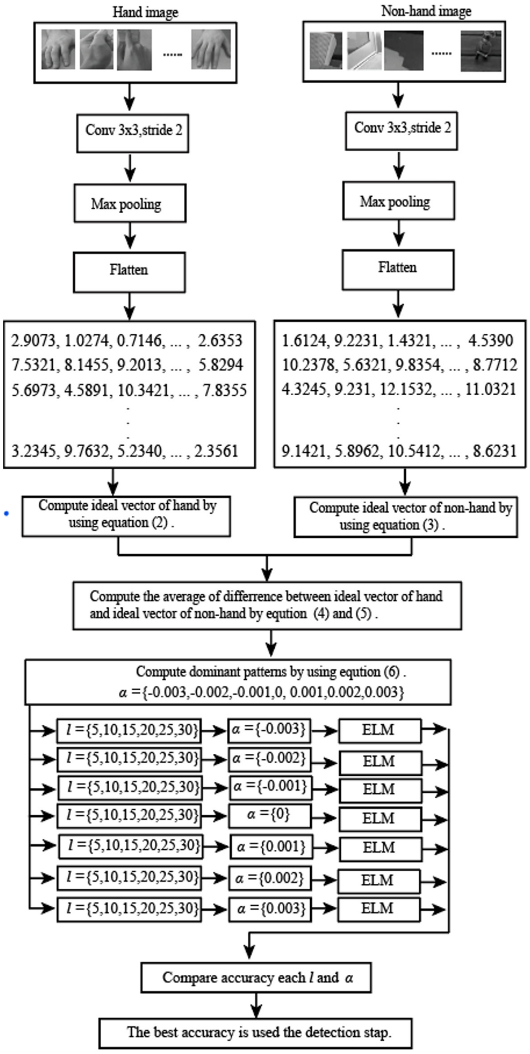


Fig.5: *DP-CNN with ELM Training Stage.*

describe differences between hand and non-hand images with a high degree of discrimination. Meanwhile, the non-dominant patterns contain features with similar values for hand and non-hand images. The non-dominant patterns affect the accuracy and time computation of hand detection.

Therefore, the proposed method improves the performance of hand detection by selecting only the dominant pattern(s) of CNN features. Figure 4 shows the main idea of the proposed method. The input layer of CNN uses a grayscale format because this format can reduce time computation. The convolution layer uses 3×3 filters and two strides. The pooling layer uses a maximum value, the detail shown in Fig. 4(a).

The CNN produces non-dominant and dominant patterns. Figure 4(b) shows an example of a non-dominant and dominant pattern of CNN features. Subsequently, the DP method selects the dominant feature pattern(s) of CNN, and these features within dominant patterns are classified as hand and non-hand images by the ELM classifier. The dominant patterns are divided features as chunks of l-consecutive elements by assigning $l = \{5, 10, 15, 20, 25, 30\}$ in which Fig. 4 shows l as 5.

The framework of the proposed method is suitable for applications involving R-CNN, Faster R-CNN [35], YOLO, and another method based on CNN object detection [23].

The proposed method consists of two stages: training and detection. Figure 5 shows the training stage. First, divide the dataset into hand and non-hand groups. Secondly, these images become extracted features by convolution and the pooling layers of CNN. Third, the CNN features of hand and non-hand are computed as an ideal vector by equations (2) and (3), respectively. Next, DP selects the features by considering the differences between the hand and non-hand forms of the CNN pattern by equations (4), (5), and (6). Finally, ELM classifies these dominant patterns. The ELM uses RBF kernel because it results in high performance and is suitable with the DP method [26]. The detection stage uses the best model in the training stage.

The detection stage feeds the images for classification. These images are the extracted features by CNN. These images are the extracted features by CNN. Then DP method selects dominant patterns from equation (6). The computation of equation 6 uses the best l and α from the training stage. Finally, ELM classifies images as hand or non-hand images. Figure 6 shows the detection stage of the proposed method.

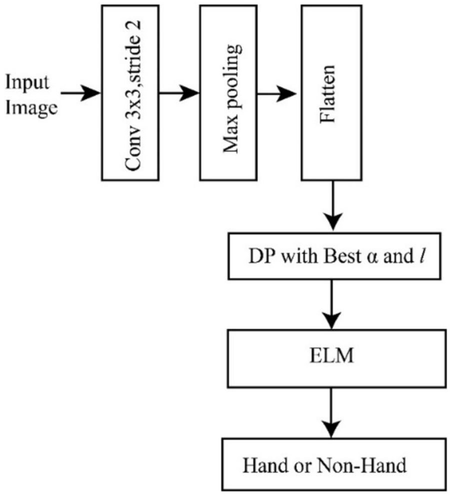


Fig.6: *DP-CNN with ELM for hand detection.*

4. DATASET AND EVALUATIOND

This study includes experiments conducted on the Egohand database [36]. The Egohand dataset [36] contains 48 videos. The video images contain hand gesture activities images: consisting of blurred and various illuminations. Figure 7 shows an example of the Egohand dataset. The datasets have 130,000 frames of videos in total. There is labeling for 4800

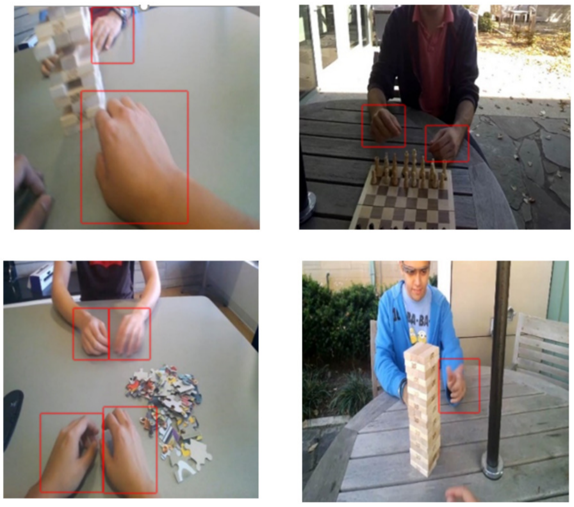


Fig.7: *Example of Egohands database.*

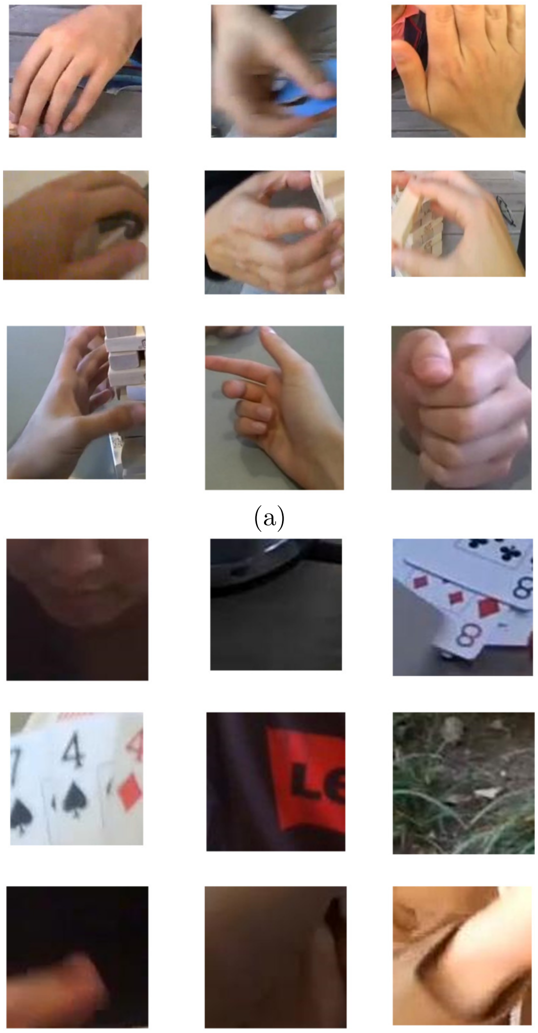


Fig.8: *Dataset for the training of the proposed method (a) hand and (b) non-hand images in the Egohands database.*

frames. In addition, the dataset includes the annotation of 15,053 hand instances. The researchers cropped 15,000 non-hands from 130,000 frames. The datasets were training and testing sets with a ratio of 0.8:0.2. Experiments on the proposed method were performed on an Intel corei5 CPU, 240 SSD RAM, and a 1080Ti GPU with 11G Memory capacity and code in MATLAB R2018a.

The proposed method uses DP [26] for selecting CNN features and ELM [37] as the classifier. The critical parameters of DP consist of l and α. The l values include 5, 10, 15, 20, 25, and 30, and the α parameter values include -0.003, -0.002, -0.001, 0, 0.001, 0.002, and 0.003. Figure 8 shows data preparation for the training of the proposed method. The authors evaluated the proposed method by accuracy (AC), precision (P), recall (R), and the F1-score [38]. The computation is shown from equations (7) - (10).

$$AC = \frac{TP + TN}{TP + FP + TN + FN} \tag{7}$$

$$P = \frac{TP}{TP + FP} \tag{8}$$

$$R = \frac{TP}{TP + FP} \tag{9}$$

$$F1 - score = 2 \times \frac{P \times R}{P + R} \tag{10}$$

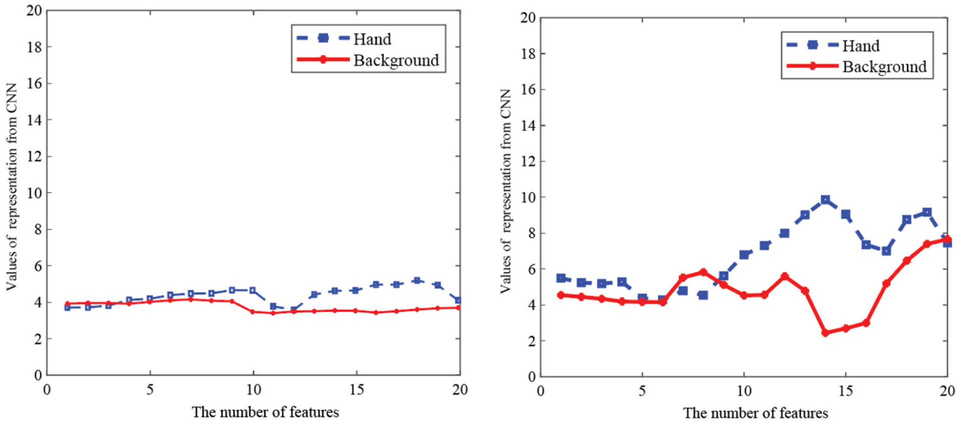
where TP is the number of hand objects correctly detected in the hand samples, TN is the number of non-hand objects correctly detected in the non-hand samples, FP is the number of hands identified as non-hands, and FN is the number of non-hands identified as hands. P is the precision at recall value R.

5. EXPERIMENT RESULTS

5.1 CNN experimentation results

This section presents the weak point of CNN in the classification stage. The CNN features identify some features of hand and non-hand as similar objects. Figure 9(a) shows an example of the ambiguous features of CNN. These features affect the performance of hand detection. In contrast, the features of CNN have differences between hand and non-hand data that give high performance. Therefore, the proposed method selects dominant features by using patterns of CNN which are shown CNN in Fig. 9(b), and discard non-dominant patterns of CNN.

One must consider the performance of CNN features within the dominant and non-dominant patterns. The researcher use the k-mean method classification of features within dominant and non-dominant patterns of CNN. This study used hand data as 100 images and a non-hand dataset as 100 images. Figures 10 and 11 show the results of the k-mean classifier of the dominant and non-dominant patterns.



(a) (b)

Fig.9: *Features of CNN (a) non-dominant patterns of CNN and (b) dominant patterns of CNN.*

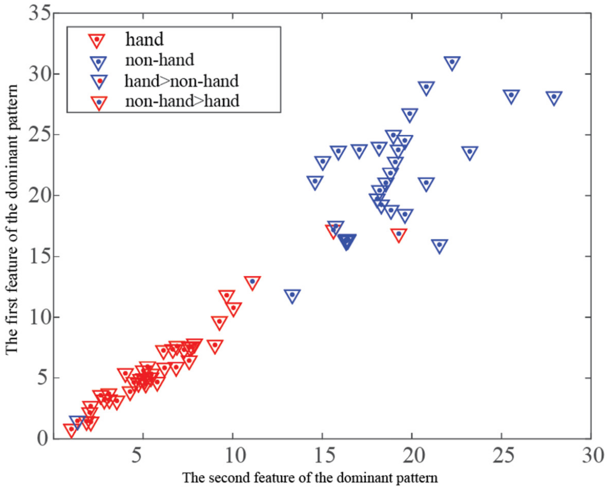


Fig.10: *Demonstrating classification of hand and non-hand data within dominant patterns of CNN by using k-mean.*

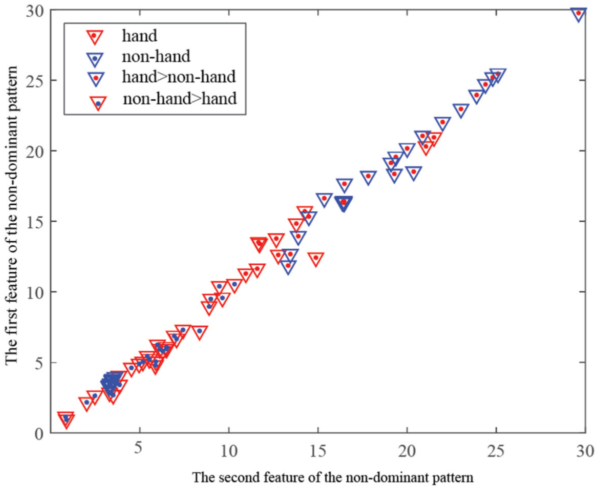


Fig.11: *Demonstrating classification of hand and non-hand data within non-dominant patterns of CNN by using k-mean.*

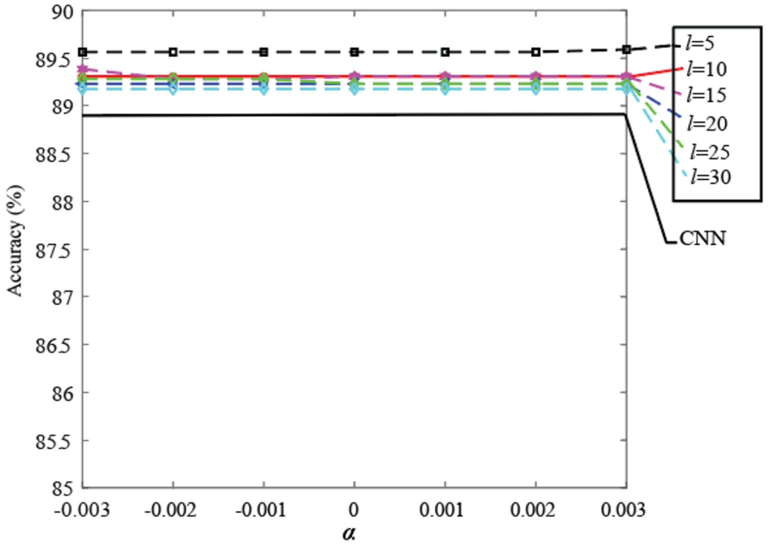


Fig.12: *Comparison of accuracy between CNN and DP-CNN with ELM with various α and l.*

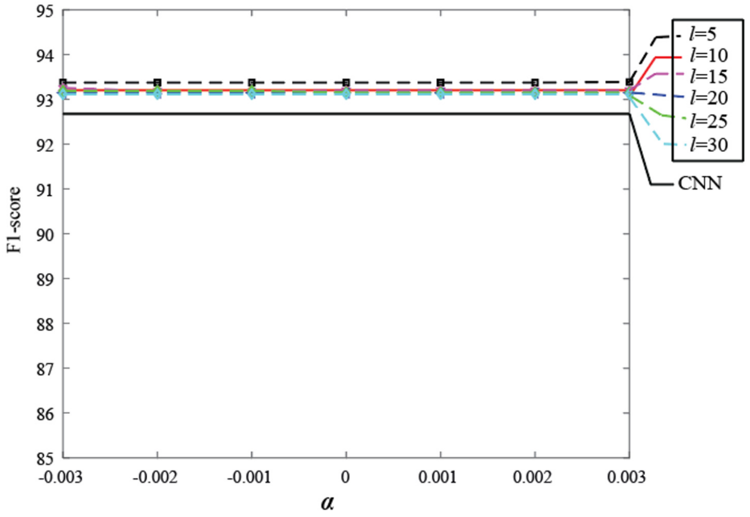


Fig.13: *Comparison of F1-score between CNN and DP-CNN with ELM with various α and l.*

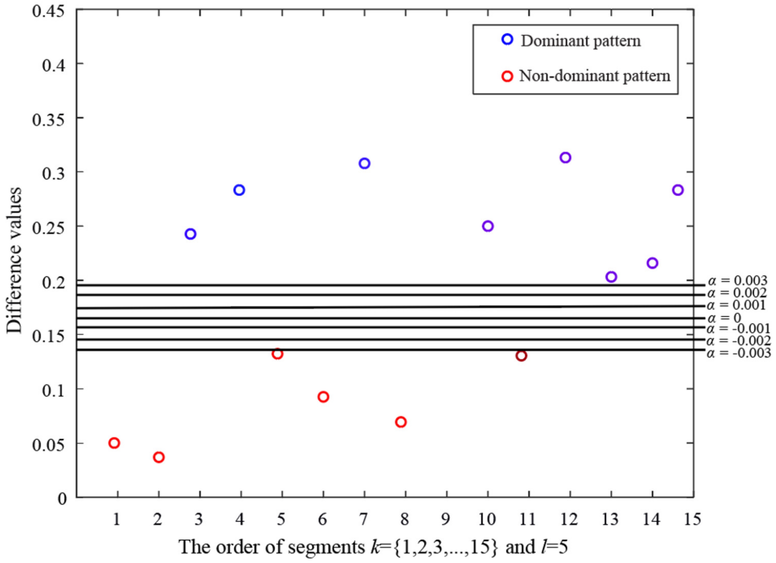


Fig.14: *Dominant and non-dominant patterns by using $\alpha = \{-0.003, -0.002, -0.001, 0, 0.001, 0.002, 0.003\}$.*

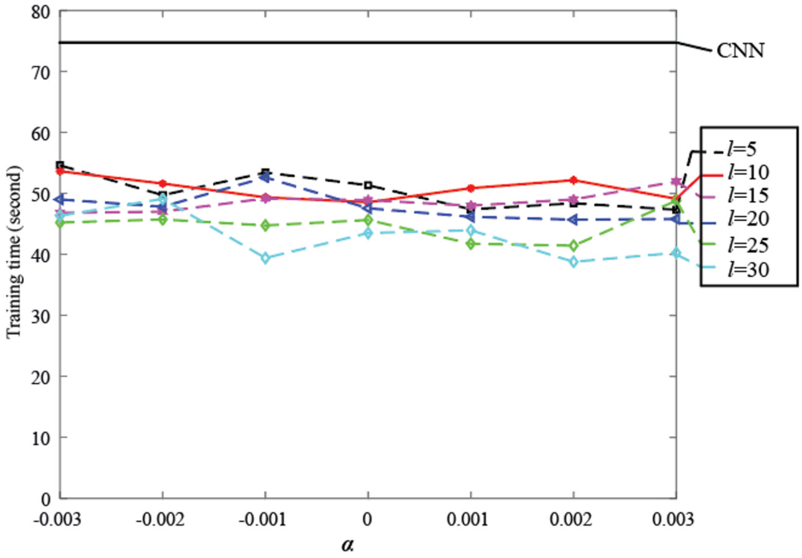


Fig.15: *Comparison of training time between CNN and DP-CNN with ELM with various α and l.*

The features within the dominant pattern that can describe differences between hand and non-hand images are explicitly better than those within the non-dominant way. Therefore, the non-dominant pattern affects misclassification. Meanwhile, the dominant pattern can adequately describe the difference between hand and non-hand images clearly, which can increase performance in the classification stage.

5.2 DP-CNN with ELM experiment results

To evaluate the performance of the proposed method (DP-CNN-ELM), the authors perform the experiments with DP parameter values of α = { -0.003, -0.002, -0.001, 0, 0.001, 0.002, 0.003} and l={5, 10, 15, 20, 25, 30}.

Figures 12 and 13 present the result of the proposed method by using the accuracy and the F1-score. Using l = 5 gives the highest accuracy and an F1-score of α = 0.003. In addition, all α and l give higher accuracy and better F1-scores than CNN. Figure 14 demonstrates different values of dominant patterns (blue color) and non-dominant patterns (red color) using k = 15 and l = 5. Most dominant values give values higher than the decision value when using α = -0.003, -0.002, -0.001, 0, 0.001, 0.002, 0.003. Therefore, the α values don't affect the performance of the proposed method. However, changing α to another value affects the performance of the proposed method. For example, if one assigns α = 0.2 with the dominant patterns consisting of k = {7, 12}, this may reduce or increase the performance of the proposed method.

Furthermore, the proposed method uses training times faster than CNN, all α and l: the result shown in Fig. 15. Figure 16 compares the number of features of CNN and the proposed method. The proposed method reduces the number of features to more than 50 percent of CNN.

The proposed method ignores non-dominant patterns of CNN. Therefore, the proposed method gives high accuracy, F1-scores, and fast computation in the training and testing stages which are better than CNN.

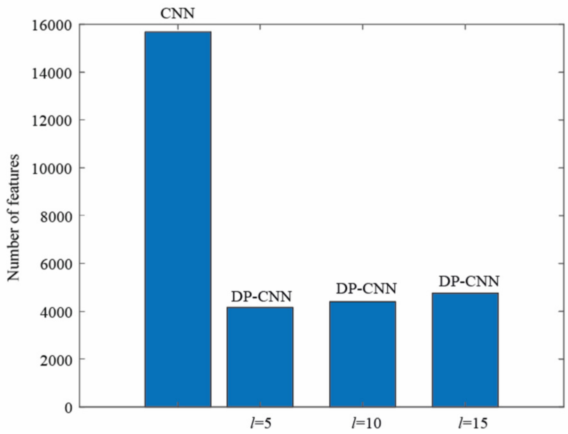


Fig.16: *Comparison of the number of features of CNN and DP-CNN with ELM using l = {5,10,15}.*

6. DISCUSSION

In this section, the researchers apply the proposed method in its application and compare it with R-CNN, faster R-CNN, and a Feature-Map-Fused Single Shot Multibox detector (FF-SSD) [39] methods on the Egohand database and Oxford database. The comparison uses average precision (AP) value.

The proposed method uses l=5 and α=0.003, which is the highest performance from section 5. The researchers use selective search[34] for segmenting ob-

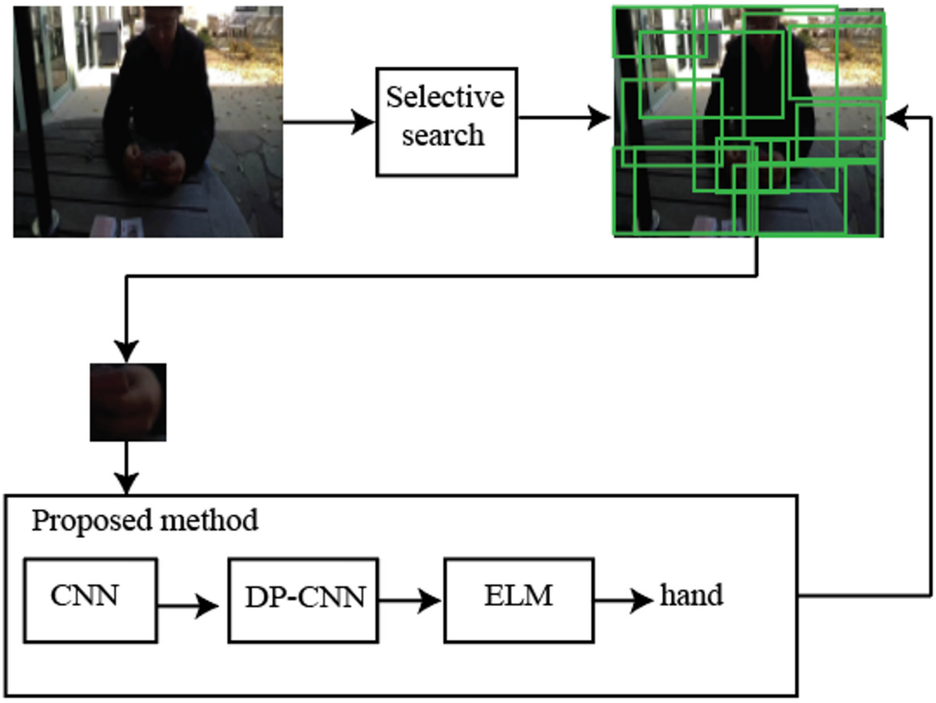


Fig.17: *Applying the proposed method in the applications.*

jects from a frame. Then each object comprises extracted features from CNN and selected dominant patterns. Finally, the ELM classifies these dominant patterns. Figure 17 shows the steps of the proposed method for use in applications. The proposed method gives an AP higher than R-CNN and faster than R-CNN within the Egohand database. As such, R-CNN and faster R-CNN are non-dominant features of CNN, while the proposed method ignores these features. However, within the Oxford database, the proposed method gives AP less than FF-SSD [39] and SF-FCNet [24]. Because of this, the Oxford database consists of complex backgrounds and various hand gestures. Table 1 shows the result. Therefore, the proposed method suitable for hand detection is with a simple background. However, the proposed method performs better than Faster R-CNN in Egohand and Oxforad databases. In addition, the proposed method does not require expensive GPUs, thus reducing the cost to the users. Since the proposed method can reduce the redundancy features of CNN by the DP method. Also, using ELM as a classifier can make simple computations within the fully-connected layers.

Table 1: *Comparison of average precision (AP) between the different methods using the Oxford hand dataset and Egohand dataset.*

Model	Oxford	Egohand
R-CNN [40]	42.3	52.27
Faster R-CNN [35]	55.7	50.00
FF-SSD [39]	73.7	83.9
SF-FCNet	**84.1**	89.40
Proposed Method (DP-CNN-ELM)	65.72	**89.58**

7. CONCLUSIONS

Hand detection is a vital pipeline task for many interactive human-hand-to-software or robotic applications. Most current research uses CNN-based models for hand detection because they produce high performance under varying conditions. This paper presents the weak point of CNN, which consists of similar features between hand and non-hand images. These features of CNN affect the performance of classification and time computation for hand detection. Therefore, this study has selected the dominant patterns of CNN features and has discarded their non-dominant features. The dominant patterns of CNN features can discriminate between hand and non-hand objects within images to a greater degree. The DP-CNN using an ELM classifier can increase the performance of CNN, consisting of accuracy, F1-Score, and time computations. R-CNN and Faster R-CNN are famous for hand detection. The authors compare DP-CNN-ELM with these methods. The result of DP-CNN-ELM gives better results than R-CNN and faster results than R-CNN. In future work, the authors shall apply DP-CNN to reduce the number of training images while using data augmentation [41][42], which is known to increase the computational cost [43] more than selecting features.

ACKNOWLEDGMENT

This research was financially supported by a grant from the Faculty of Informatics, Mahasarakham University, in 2022.

References

[1] Q. Wang, G. Zhang, and S. Yu, "2D Hand Detection Using Multi-Feature Skin Model Supervised Cascaded CNN," *Journal of Signal Processing Systems*, vol. 91, no. 10, pp. 1105–1113, 2019.

[2] J. Li, C. Li, J. Han, Y. Shi, G. Bian, and S. Zhou, "Robust Hand Gesture Recognition Using HOG-9ULBP Features and SVM Model," *Electronics*, vol. 11, no. 7, pp. 1–15, 2022.

[3] Y. G. Zhao, F. Zheng, and Z. Song, "Hand Detection Using Cascade of Softmax Classifiers," *Advances in Multimedia*, vol. 2018, no. 9204854, pp. 1-11, 2018.

[4] F. Sohel and M. Bennamoun, "Robust Pose Invariant Shape and Texture based Hand Recognition," arXiv, p. 1912:10373, 2019.

[5] J. Qi, K. Xu, and X. Ding, "Approach to hand posture recognition based on hand shape features for human–robot interaction," *Complex & Intelligent Systems*, vol. 8, no. 4, pp. 2825–2842, 2022.

[6] S. Medjram, M. C. Babahenini, A. Taleb-Ahmed, and Y. Mohamed Ben Ali, "Automatic Hand Detection in Color Images based on skin region verification," *Multimedia Tools and Applications.*, vol. 77, no. 11, pp. 13821-13851, 2018.

[7] A. Mittal, A. Zisserman, and P. H. S. Torr, "Hand detection using multiple proposals," *Proceedings of the British Machine Vision Conference*, pp. 75.1-75.11, 2011.

[8] D. Forsyth, "Object Detection with Discriminatively Trained Part Based Models," *Computer (Long. Beach. Calif)*, vol. 47, no. 2, pp. 6–7, 2009.

[9] K. -p. Feng and F. Yuan, "Static hand gesture recognition based on HOG characters and support vector machines," *2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA)*, Toronto, ON, Canada, pp. 936-938, 2013.

[10] Q. Chen, N. D. Georganas and E. M. Petriu, "Hand Gesture Recognition Using Haar-Like Features and a Stochastic Context-Free Grammar," in *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 8, pp. 1562-1571, Aug. 2008.

[11] R. H. L. Songhita Misra, "Approach toward extraction of sparse texture features and their application in robust two-level bare-hand detection," *Journal of Electronic Imaging*, vol. 27, no. 5, p. 051209, 2018.

[12] J. Qi, K. Xu, and X. Ding, "Approach to hand posture recognition based on hand shape features for human–robot interaction," *Complex & Intelligent Systems*, vol. 8, pp. 2825-2842,2021.

[13] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[14] V. Sangeetha and K. J. R. Prasad, "Deep Residual Learning for Image Recognition Kaiming," *Indian J. Chem. - Sect. B Org. Med. Chem.*, vol. 45, no. 8, pp. 1951–1954, 2016.

[15] A. A. Q. Mohammed, J. Lv, and M. D. S. Islam, "A deep learning-based end-to-end composite system for hand detection and gesture recognition," *Sensors*, vol. 19, no. 23, 2019.

[16] C. Xu, W. Cai, Y. Li, J. Zhou, and L. Wei, "Accurate hand detection from single-color images by reconstructing hand appearances," *Sensors*, vol. 20, no. 1, pp. 1–21, 2020.

[17] A. Tellaeche Iglesias, I. Fidalgo Astorquia, J. I. Vázquez Gómez, and S. Saikia, "Gesture-Based Human Machine Interaction Using RCNNs in Limited Computation Power Devices," *Sensors*, vol. 21, no. 24, 2021.

[18] Nguyen, Van-Toi, et al., "A method for hand detection based on Internal Haar-like features and Cascaded AdaBoost Classifier," *Proceedings of The Fourth International Conference on Communications and Electronics (ICCE 2012)*, pp. 608-613, 2012.

[19] P. S. Neethu, R. Suguna, and D. Sathish, "An efficient method for human hand gesture detection and recognition using deep learning convolutional neural networks," *Soft Computing*, vol. 24, no. 20, pp. 15239–15248, 2020.

[20] M. C. Su, J. H. Chen, V. Trisandini Azzizi, H. L. Chang, and H. H. Wei, "Smart training: Mask R-CNN oriented approach," *Expert Systems with Applications*, vol. 185, no. May, p. 115595, 2021.

[21] K. Roy, A. Mohanty and R. R. Sahay, "Deep Learning Based Hand Detection in Cluttered Environment Using Skin Segmentation," *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, Venice, Italy, pp. 640-649, 2017.

[22] L. Yang et al., "An embedded implementation of CNN-based hand detection and orientation estimation algorithm," *Machine Vision and Applications*, vol. 30, no. 6, pp.

[23] X. Deng et al., "Joint Hand Detection and Rotation Estimation Using CNN," in *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1888-1900, April 2018.

[24] B. Qiang et al., "SqueezeNet and Fusion Network-Based Accurate Fast Fully Convolutional Network for Hand Detection and Gesture Recognition," *IEEE Access*, vol. 9, pp.

77661–77674, 2021.

[25] Z. Cui and N. Lu, "Feature selection accelerated convolutional neural networks for visual tracking," *Applied Intelligence*, vol. 51, no. 11, pp. 8230–8244, 2021.

[26] N. Laopracha, K. Sunat and S. Chiewchanwattana, "A Novel Feature Selection in Vehicle Detection Through the Selection of Dominant Patterns of Histograms of Oriented Gradients (DPHOG)," in *IEEE Access*, vol. 7, pp. 20894-20919, 2019.

[27] Y. Shen, L. Xiao, J. Chen, and D. Pan, "A Spectral-Spatial Domain-Specific Convolutional Deep Extreme Learning Machine for Supervised Hyperspectral Image Classification," in *IEEE Access*, vol. 7, pp. 132240–132252, 2019.

[28] J. Wang, S. Lu, S. Wang, and Y. Zhang, "A review on extreme leanring machine," *Multimedia-based Healthcare Systems using Computational Intelligence*, vol. 81, pp. 41611–41660, 2022.

[29] M. Langkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, no. 1, pp. 11–24, 2014.

[30] S. Phiphitphatphaisit and O. Surinta, "Deep feature extraction technique based on conv1d and lstm network for food image recognition," *Engineering and Applied Science Research*, vol. 48, no. 5, pp. 581–592, 2021.

[31] D. U. N. Qomariah, H. Tjandrasa and C. Fatichah, "Classification of Diabetic Retinopathy and Normal Retinal Images using CNN and SVM," *2019 12th International Conference on Information & Communication Technology and System (ICTS)*, Surabaya, Indonesia, pp. 152-157, 2019.

[32] U. Knauer et al., "Tree species classification based on hybrid ensembles of a convolutional neural network (CNN) and random forest classifiers," *Remote Sensing*, vol. 11, no. 23, 2019.

[33] T. Hu, M. Khishe, M. Mohammadi, G. R. Parvizi, S. H. Taher Karim, and T. A. Rashid, "Real-time COVID-19 diagnosis from X-Ray images using deep CNN and extreme learning machines stabilized by chimp optimization algorithm," *Biomedical Signal Processing and Control*, vol. 68, p. 102764, 2021.

[34] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.

[35] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 1 June 2017.

[36] S. Bambach, S. Lee, D. J. Crandall and C. Yu, "Lending A Hand: Detecting Hands and Recognizing Activities in Complex Egocentric Interactions," *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, pp. 1949-1957, 2015.

[37] H. Fu, C.-M. Vong, P.-K. Wong, and Z. Yang, "Fast detection of impact location using kernel extreme learning machine," *Neural Computing and Applications*, vol. 27, no. 1, pp. 121–130, 2016.

[38] Y. Liu, P. Sun, N. Wergeles, and Y. Shang, "A survey and performance evaluation of deep learning methods for small object detection," *Expert Systems with Applications*, vol. 172, no. October 2020, p. 114602, 2021.

[39] Q. Gao, J. Liu, and Z. Ju, "Robust real-time hand detection and localization for space human–robot interaction based on deep learning," *Neurocomputing*, vol. 390, pp. 198–206, 2020.

[40] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp. 580-587, 2014.

[41] M. Z. Islam, M. S. Hossain, R. ul Islam and K. Andersson, "Static Hand Gesture Recognition using Convolutional Neural Network with Data Augmentation," *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, Spokane, WA, USA, pp. 324-329, 2019.

[42] [Y. Wang, X. Wei, X. Tang, H. Shen, and L. Ding, "CNN tracking based on data augmentation," *Knowledge-Based Systems*, vol. 194, p. 105594, 2020.

[43] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, no. 60, 2019.

Natakapriya Laopracha received her B.Sc. degree in computer science from Mahasarakham University and her M.S. degree in computer science and Ph.D. degree in computer science from Khon Kean University, Thailand. She is currently a lecturer in the department of computer science, Mahasarakham University, Thailand. Her research interests include computer vision, pattern recognition, intelligent systems and machine learning.

Kaveepoj Bunluewong received his bachelor degree in computer science from Udon Thani Rajabhat Institute, Thailand. He received a masters degree in computer science from Khon Kaen University, Thailand. He is currently a lecturer at the Department of computer science, Faculty of Informatics, Mahasarakham University, Thailand. His research interests include artificial intelligence, machine learning, and big data.