

Object Recognition with Hybrid Deep Learning Methods and Testing on Embedded Systems

Yavuz Selim TASPINAR*¹, Murat SELEK²

Submitted: 28/02/2020 Accepted: 18/04/2020

Abstract: Object recognition applications can be made with deep neural networks. However, this process may require intensive processing load. For this purpose, hybrid object recognition algorithms that can be created for the recognition of an object in the image and the comparison of the working time of these algorithms on various embedded systems are emphasized. While Haar Cascade, Local Binary Pattern (LBP) and Histogram Oriented Gradients (HOG) algorithms are used for object detection, Convolutional Neural Network (CNN) and Deep Neural Network (DNN) algorithms are used for classification. As a result, six hybrid structures such as Haar Cascade+CNN, LBP+CNN, HOG+CNN and Haar Cascade+DNN, LBP+DNN, HOG+DNN are developed. In this study, these 6 hybrid algorithms were analyzed in terms of success percentage and time, then compared with each other. Microsoft COCO dataset was used to train and test all these hybrid algorithms. Object recognition success of CNN was 76.33%. Object recognition success of Haar Cascade+CNN, one of the hybrid methods we recommend, with a success rate of 78.6% is higher than CNN and other hybrid methods. LBP+CNN method recognized objects in 0.487 seconds which is faster than any other hybrid methods. In our study, Nvidia Jetson TX2, Asus TinkerBoard, Raspberry Pi 3 B+ were used as embedded systems. As a result of these tests, Haar Cascade+CNN method on Nvidia Jetson TX2 was detected in 0.1303 seconds, LBP+DNN and Haar Cascade+DNN methods on Asus Tinker Board were detected in 0.2459 seconds, and HOG+DNN method on Raspberry Pi 3 B+ was detected in 0.7153 seconds..

Keywords: Deep learning, Image classification, object detection, hybrid methods

1. Introduction

The progress of research in the field of computer vision accelerated considerably as the computing capacity of computers increases. The Haar Cascade [1], Local Binary Patterns (LBP) [2], and Histogram of Oriented Gradients (HOG) [3] methods are simple and effective algorithms for object detection in the image. Haar Cascade is frequently used in many areas, such as face detection [4], eye detection [5], traffic light detection [6], real-time motion detection [7]. Similarly, the LBP algorithm is used in many areas such as to detect 3D hand movements [8], face detection [9], detection of object forms [10], and text detection [11]. The HOG algorithm is also used for object detection in areas similar to other object detection methods, such as character detection [12], handwriting detection [13], human detection [14]. All three algorithms are extensively used because of the success in object detection [15] and are used with the advanced machine learning algorithms for the classification stage [16]. Objects detected by these methods can be recognized by using various classification methods such as Artificial Neural Network, Convolutional Neural Network (CNN), Deep Neural Network (DNN), Support Vector Machine and Fuzzy Logic. Although ANN is an algorithm frequently used in object recognition [17] before, new methods such as CNN and DNN increased success of object recognition to very high levels [18]. CNN and DNN are more preferred because of their success in image processing [19-20] object detection and classification as well as they are used in natural language

processing [21], inference in large datasets [22], in signal processing. However, the performance of computers used today may be inadequate because CNN and DNN require many processes [23]. In order to increase the running success of these algorithms or to reduce their number of processes, hybridization studies are performed and the results of these hybridization studies yielded quite successful results [24-27]. In a study on the social network, the faces of people were detected by using the HOG algorithm, then face feature extraction was made by CNN, they are classified with the Support Vector Machine (SVM) and the process of labeling people with their names was achieved very fast with a 98% success [28]. In another study, objects in the images, the feature of which was extracted by CNN, were detected by Region Proposal Network and detected objects were classified. Thus, the Region Based CNN method provides better results than CNN [30]. The method that was also tested in real time provided better results than CNN again [24]. Deep Hybrid Model proposed in a study on scene classification increased the scene classification success rates obtained in previous studies [25].

With the increase in portable devices and autonomous systems, the interest in embedded systems is increasing rapidly. In addition to the advantages of performing calculations on embedded systems, there are disadvantages. Numerous studies have been conducted on these embedded systems, which differ in cost, performance and system characteristics. Nvidia Jetson TX2 used in our study is frequently used in robotic systems [26] and autonomous vehicles [27] due to its calculation speed. Asus TinkerBoard is used for the control of electronic circuits [28], for the control of defense systems [29] and for image processing [30]. Raspberry Pi is

¹ Doganhisar Vocational School, Selcuk University, TURKEY
ORCID ID : 0000-0002-7278-4241

² Vocational School of Technical Sciences, Konya Technical University,
TURKEY, ORCID ID : 0000-0001-8642-1823

* Corresponding Author Email: ytaspinar@selcuk.edu.tr

preferred over other systems because of its cost advantage. This system is frequently used in network applications [31], pattern recognition [39], and radiation damage detection [32].

In our study, Haar Cascade+CNN, LBP+CNN, HOG+CNN, Haar Cascade+DNN, LBP+DNN, HOG+DNN hybrid methods were analyzed in order to reduce the computational load of classification methods and increase the object recognition success. It is aimed to reduce the processing load and increase the classification success by using the hybrid methods operating with pipeline logic that reduced the data size which entered CNN and DNN. The proposed hybrid methods were trained by using Microsoft COCO dataset and the tests were performed with images taken from the same dataset [33].

2. Material and Methods

The explanations of Haar Cascade, LBP, HOG object detection algorithms and CNN, DNN classification algorithms used in this study are given in this section. Also, the explanations of hybrid object recognition methods that are generated by joining object detection algorithms and classification algorithms are included in this section.

2.1. Haar Cascade

The Haar Cascade algorithm, used to find and track objects in video or images, was proposed by Paul Viola and Micheal Jones [34]. Firstly, the image is converted from RGB level to gray level in the method which steps are Haar feature selection, creating integral image, calculating the Haar features, Adaboost and Cascade. After these steps, it is decided which Haar feature will be used. In our study, the Haar feature windows in Figure 1 were used.

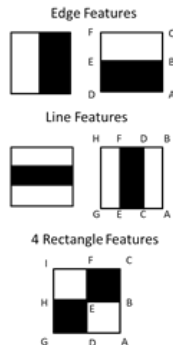


Fig. 1. Haar Features

The integral image is generated in order to increase the running efficiency of the algorithm. The pixel value of the integral image is indicated by $I(x, y)$. (x, y) refers to the point where the integral image is located. m represents the x coordinate of the integral image and n represents the y coordinate of the integral image. The integral image calculation is shown in (1).

$$I(x, y) = \sum_{x=0}^m \sum_{y=0}^n i(x, y) \quad (1)$$

$I(x, y)$ in the equation is the pixel value of the image given as input. The integral image contains the sum of the pixel values of all coordinates smaller than the (x, y) coordinate value. In every process, instead of calculating the sum of the pixel values of the window, the sum of the pixel values of the desired window is found quickly by taking the integral. In Figure 2, the calculation of the pixel values of a selected window in the image is shown in (2).

$$\sum I(x, y) = (I(D) + I(A)) - (I(B) + I(C)) \quad (2)$$

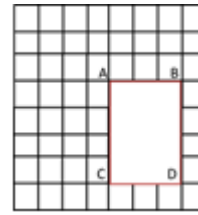


Fig. 2. The window that was taken integral value

A training dataset is needed to train the Haar Cascade algorithm. The dataset contains positive images included the object to be detected and negative images not included the object to be detected. The windows in Figure 1 are used to detect the object in the positive images. It provides a total of 160K features to be searched on a 24x24 size image by changing the window size.

The scan is performed by scrolling Haar feature windows through the entire image [34]. These windows are called Haar features. Haar features are obtained by calculating the difference between the pixel values of the bright and dark regions according to the Haar features. All the pixel values must be above a threshold value for a region considered as bright; all the pixel values must be below a threshold value for a region considered as dark. The bright dark threshold value can be found by calculating the average of the pixels in the image, or can be manually determined by the user. In our study, the bright dark threshold was found by calculating the average of the pixels.

It takes time to obtain the positive and negative regions by scrolling subwindows in the whole image. As a result of the features obtained with the Haar feature windows, a weak classifier is generated from each features, and a group is generated from these weak classifiers. Then a strong classifier is generated by using weak classifiers with the minimum failure rates. This method is called the Adaboost classification method. Each Haar feature selected with Adaboost is called a weak classifier. Each Haar feature differentiates a particular feature of the object. Training is the comparison of the selected feature with a trained threshold value. Any Haar feature (f) calculation is shown in (3).

$$f(x) = \sum (BlackPixels) - \sum (WhitePixels) \quad (3)$$

The threshold value of Haar feature (θ) and polarity that is the direction of the equation (p) must be determined for a trained Haar feature considered as a weak classifier. The threshold value is obtained by the method of calculating the sum of the Haar feature values obtained in the same position in the positive pictures including the object used in our study and dividing by the number of positive images or manually determined by the user [36]. With the obtained f value, each weak classifier of each image is calculated as in (4). j in the equation represents each Haar feature.

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

First, weights are given to each positive and negative image in the algorithm and the sum of these weights will be 0.5. If a is the number of positive images and b is the number of negative images, the weight of each image W_i for the positive and negative images is calculated as in (5). In each t cycle, weights are normalized as in (6). n in equation represents the total number of features in the image, i represents each image.

$$W_i = \frac{1}{2} a \quad i = 1, 2, 3, \dots, a \quad (5)$$

$$W_i = \frac{1}{2} b \quad i = 1, 2, 3, \dots, b$$

$$W_{t,i} = \frac{W_{t,i}}{\sum_{j=1}^n W_{t,i}} \quad (6)$$

After the first cycle, the weights take the initial value and the Haar features are calculated by the (4) in the positive and negative images. The failure value (ϵ_t) of the Haar features is calculated according to (7) and positive images will be $y_i = 1$ and negative images will be $y_i = 0$. n represents the total number of images used in the training in the dataset. The minimum failure value of Haar feature is the weak classifier.

$$\epsilon_t = \sum_i W_i |h_j(x) - y_i| \quad (7)$$

The addition includes all positive and negative images. Since the value in the absolute value will be 0 for the true classifications and 1 for the false classifications, the failure value is the sum of the weights of the false classified images. The Adaboost algorithm increases the weights of false classified images and enables the search for a new weak classifier. When going to the next cycle, weights are updated according to (8).

$$W_{t+1,i} = W_{t,i} \beta_t^{1-e} \quad (8)$$

i -th image is classified true, $e_i = 0$; if it is classified false, $e_i = 1$. The β value is calculated as in (9).

$$\beta_t = \frac{\epsilon_t}{1 - \epsilon_t} \quad (9)$$

With one weak classifier, it is impossible to select positive images at the desired rate and eliminate the negatives. For this reason, weak classifiers are joined to form strong classifiers. Each weak classifier has a weight and this weight is calculated as in (10).

$$\alpha_t = \log\left(\frac{1}{\beta_t}\right) \quad (10)$$

The weak classifier takes a weight that is inversely proportional to the failure resulting from the classification. The strong classifier resulting from the joining of weak classifiers goes through the control in (11).

$$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases} \quad (11)$$



Fig. 3. Cascade classification structure

As shown in Figure 3, the image entering the Cascade classifier is considered negative if it is eliminated by a weak classifier. The image passing through all classifiers is considered positive and object detection is performed.

2.2. Local Binary Pattern

The main idea is to summarize the local structure in an image by comparing each pixel with its neighbor [37]. It looks at the points surrounding a central point and gives a binary pattern by testing whether the surrounding points are higher or lower than the central point. Each pixel is labeled by converting the difference between itself and its neighbors into a binary system with the step function $u(x)$ [38]. In equation 12, the equation of finding the binary value obtained from the result of the LBP operator is shown. x_c represents the center pixel, the center of the x_p central pixel, the distance between the R neighbors to the center, the number of neighboring machined P , the difference between the Y pixel and neighboring pixel Y and the bits obtained from the $u(y)$ LBP operator.

$$LBP_{P,R}(x_c) = \sum_{p=0}^{P-1} u(x_p - x_c) 2^p \quad (12)$$

$$u_y = \begin{cases} 1 & y \geq 0 \\ 0 & y < 0 \end{cases}$$

The LBP value generated by combining neighboring values is a unique identifier for the central pixel. Figure 4 shows the processing of the pixels with the LBP operator.

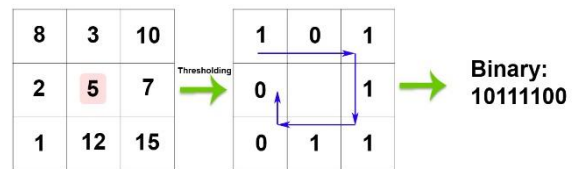


Fig. 4. LBP Operator

After these processes, each LBP value is converted to histograms. Histograms contain information about patterns at pixel level and are used as features in the classification process. The histograms obtained are compared with the histograms used in training to be decided whether the object exists in the image by Adaboost classification method used in the Haar Cascade method. The LBP method was used to detect object in our hybrid method.

2.3. Histogram Oriented Gradients

The HOG method is a method that describes the image as a group of local histograms and is first proposed by Sashua [39] and Dalal [40]. First, the image is converted to grayscale. After this process, the brightness gradient of each pixel is calculated. An orientation histogram is generated for image pixels divided into 9x9 cells. During this process, normalization is performed. The normalization is performed by taking the average of the pixel values in each 9x9 cell to minimize factors such as light intensity, shading and background. By means of normalization, the recognizer provides tolerance against changes caused by lighting and shading. In the HOG method, the brightness gradient (13) is shown [41]. m indicates the gradient, θ indicates the gradient direction, L indicates the brightness value of the pixel is shown (14). In our study, HOG was used to detect object in hybrid method.

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (13)$$

$$\theta(x,y) = \tan^{-1} \left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)} \right) \quad (14)$$

2.4. Convolutional Neural Network

Deep learning architecture, CNN is a neural network consisting of several layers [42]. The Convolution layer uses windows that move on the matrix. These windows are called filters or cores. The randomly generated filter is applied to the windows in the image matrix. The filter applied to the end of the matrix finally emerges with a new feature matrix. The matrix to which the filter is applied is at the end of the convolution layer as much as the filter. In the convolution layer, regional patterns are learned instead of global patterns. Convolutional neural networks have two interesting qualities. First, the patterns they learn do not change direction. Recognize a pattern that an image learns in any part of the image. In this way, it is able to learn the generalization power representations with less data. Second, they can learn spatial hierarchies. The first convolution layer learns regional patterns such as corners, while the second convolution layer learns wider patterns than the patterns learned by the first layer, and so on in the other layers. This helps him to learn more complex visual concepts effectively. The aim of this study was to increase the success of the convolutional neural networks used for object recognition. The image complexity entering the neural network has been reduced by other object detection methods, thus increasing the success of convolutional neural networks.

2.5. Deep Neural Network

Deep neural network [43] is a structure used for object detection. The number of hidden layers in the network structure is many, so it can give more successful results than artificial neural networks. In our study, 1 input with $28 * 28$ neurons, 1 output with 20 neurons and 9 hidden layer DNN structures were used. In our study DNN was used in object detection and classification, in hybrid methods DNN is used in object classification.

2.6. Hybrid Object Recognition Algorithm Structure

In this study, Haar Cascade, LBP and HOG were used for object detection, CNN and DNN were used for classification. As a result of these processes, Haar Cascade+CNN, LBP+CNN, HOG+CNN, Haar+DNN, LBP+DNN, HOG+DNN structures are emerged. The section of the object is taken by detecting the object that desired to be detected with the detection method in the image. The sectioned object is given as input to CNN or DNN. As a result of these processes, the size of the image is reduced with the sectioned image and the neural density of CNN and DNN is also reduced. As a result of these, the newly generated structures are able to achieve higher success and speed compared to non-hybrid methods in the study. In Figure 5, the structure of the proposed hybrid methods is shown.

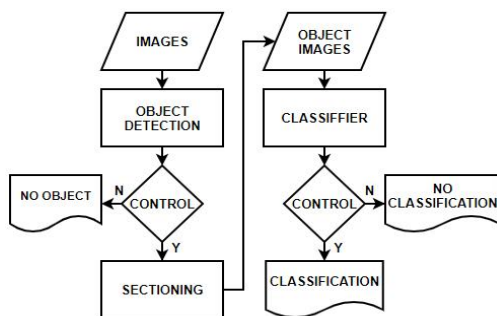


Fig. 5. Flow diagram of hybrid methods

Figure 6 shows the process of detecting an image with object detection methods and then cutting and sending it to the classifier.

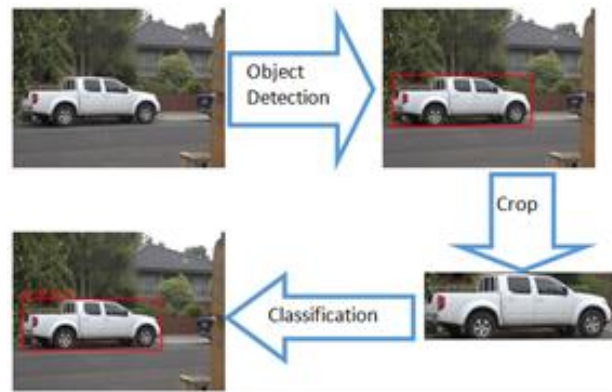


Fig. 6. An object detection and classification in hybrid systems

2.7. Availability of data and materials

In our study, Microsoft COCO dataset was used by taking 20 object classes to test hybrid methods. There are also unlabelled images in the dataset that includes labelled images. Dataset was generated by categorizing the images by Microsoft Bing and real people. In order to test the success of the algorithms, the images that were not used in the training of algorithms were used. 80% of the dataset was used for training and 20% for the test. In the training of algorithms, a computer with a Windows operating system with an i5 2.5 GHz processor and 8 GB RAM was used. Although the lack of memory is a disadvantage of the CPU, memory management is provided by using the C # programming language. In this way, the algorithms within the hybrid methods are enabled to run hierarchically by eliminating the possibility of running at the same time. In image processing, OpenCV library, Python programming language, C # programming language and Tensorflow library were used.

2.8. Embedded systems

Embedded systems used TensorFlow [44] Developed by Google. All of the embedded systems used in our study support tensorflow library. Nvidia Jetson TX2 embedded system Linux based Ubuntu 16.04, Asus Tinkerboard embedded system Linux based TinkerOS and Raspberry Pi 3 B+ embedded system Linux-based Raspbian operating system. Python programming language, Tensorflow library and all other necessary libraries can be easily installed on these operating systems. The features of all embedded systems used are shown in the Table 1.

Table 1. Properties of the systems used in our study

	CPU	Frequency	Cores	Memory	OS
PC	Intel Core i5	2.7GHz	4	8GB	Windows 10
Nvidia Jetson TX2	ARMv8+	2GHz	6	8GB	Ubuntu 16.04
	NVIDIA Pascal GPU				
ASUS Tinkerboard	ARM Cortex-A17	1.8GHz	4	2GB	TinkerOS
	- Rockchip RK3388 SoC				
Raspberry Pi 3+	ARMv8	1.4 GHz	4	1GB	Raspbian

The processors and memory capacities of the systems used in the study are different. Nvidia Jetson TX2 stands out with its GPU compared to other systems. This system is an artificial intelligence computing device and includes ARM CPU and Nvidia Pascal GPU.

3. Results

Tests were performed to determine the optimum configuration by testing the configurations used in the CNN and DNN classification algorithms in our study. The test results of the success of these configurations and the test results of the success of hybrid algorithms are given in this section.

3.1. Comparison of CNN Structure

In our CNN configuration, 10 convolution layers, 2 max pooling layers, 2 fully connected layers, 1 softmax layers and activation function Relu are used. Convolution layer, pooling layer and activation functions were modified and various tests were performed. As a result of these tests, the configuration we used was preferred because it was more successful than other configurations. The test results are shown in Table 2.

Table 2. Classification success rate of cnn configurations (%)

Pooling Layers Activation Function	Avarage Pooling			Max Pooling		
	Tanh	Sigmoi d	Relu	Tanh	Sigmoi d	Relu
6 Con.Layer	57,16	52,2	57,22	50,82	51,37	52,5
8 Con.Layer	41,28	45,79	42,97	51,01	44,77	48,33
10 Con.Layer	80,98	80,99	80,49	80,69	80,68	81,26
12 Con.Layer	54,29	64,62	73,04	75,44	72,43	77,02

As a result of these tests, algorithms are generated by 10 convolution layer, max pooling layer and Relu activation function and they were used in our study.

3.2. Hybrid Method Test Results

Haar Cascade+CNN, LBP+CNN, HOG+CNN, Haar Cascade+DNN, LBP+DNN, HOG+DNN object detection and classification methods were tested with the dataset used in our study and their success was compared. Tests were performed by selecting the images, in which all the methods used in the study can detect object. Object recognition successes and times were obtained by testing each image with all methods as shown in Figure 7.



Fig. 7. Hybrid object recognition methods on multi-object image

Table 3. Avarage success rate

Avarage Success Rate (%)						
CNN	Haar +	LBP +	HOG +	Haar +	LBP +	HOG +
	CNN	CNN	CNN	DNN	DNN	DNN
76,3	78,6	75,8	76,8	77,9	77,5	72,3

As a result of these results, Haar Cascade+CNN, one of the hybrid methods using CNN classifier, recognized objects with 78,60%

success rate which is higher than other methods used in the study. The run times of the algorithms were measured in all embedded systems for each image. The average run time of each algorithm was found for each embedded system. These periods are shown in Table 4.

Table 4. The average object recognition times of the methods (second)

	CNN	Haar +	LBP +	HOG +	Haar +	LBP +	HOG +
		CNN	CNN	CNN	DNN	DNN	DNN
PC	0,58	0,58	0,48	0,58	0,69	0,49	0,79
NVIDIA JETSON TX2	0,135	0,130	0,131	0,135	0,132	0,134	0,129
ASUS TINKERBOAR D	0,247	0,246	0,245	0,246	0,245	0,247	0,251
RASPBERRY PI 3 B+	0,715	0,720	0,717	0,716	0,718	0,718	0,715

According to the data in Table 4, the tests performed on the PC, LBP+CNN hybrid method in 0.487 seconds in a shorter location than the other methods in the search and ranked. Tests on the Nvidia Jetson TX2 system have the shortest operating time of the Haar Cascade+CNN method with 0.1303 seconds. Asus Tinker Board system tests were performed with 0.2458 seconds and gave faster results than LPB+CNN method. When we look at the other systems that have the weakest hardware when looking at Raspberry Pi 3 B+ system, the CNN method works fastest with 0.7159 seconds. Looking at all systems, the Nvidia Jetson TX2 is ultimately higher than the other systems before the GPU's advantages. When the success story of hybrid methods is examined, Haar Cascade+CNN method is designed as the most successful method according to Table 4. Accordingly, there is a 2.79% difference between the recognition success of the LBP+CNN hybrid method and the object recognition success of the Haar Cascade+CNN hybrid method. This ratio is not to be underestimated in artificial intelligence calculations. On the Nvidia Jetson TX2 system, Haar Cascade+CNN method is 0.1303, LPB+CNN method is 0.11310. When we look at all these results, Haar Cascade+CNN method is the most successful method in our study, Nvidia Jetson TX2 is seen here as the fastest embedded system.

4. Conclusion

In this study, different hybrid methods were developed and tested on different systems. It can be said that the processes on the PC take longer than the processes performed on other systems because of the background applications in the operating system. It is thought that the object labeling processes on the image are the reasons why hybrid methods are more successful than CNN method. The dataset must be correctly labeled in the labeling process before training. Otherwise, object recognition may be low. It can be said that the object detection methods increase the success rate by first cutting the object properly from the image and then sending it to the classifier and shorten the processing time due to the decrease in the size of the data entering the classifiers.

Acknowledgements

This work was supported by Selcuk University Scientific Research Programs Coordination project number 17101015.

References

- [1] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* (Vol. 1, pp. I-1). IEEE.
- [2] Cevikalp, H., & Triggs, B. (2017). Visual object detection using cascades of binary and one-class classifiers. *International Journal of Computer Vision*, 123(3), 334-349.
- [3] Dehghani, A., & Moloney, D. (2016, August). Speed improvement of object recognition using boundary-bitmap of histogram of oriented gradients. In *Image, Vision and Computing (ICIVC), International Conference on* (pp. 51-56). IEEE.
- [4] Reschke, J., & Sehr, A. (2017). Face Recognition with Machine Learning in OpenCV_ Fusion of the results with the Localization Data of an Acoustic Camera for Speaker Identification. *arXiv preprint arXiv:1707.00835*.
- [5] Utamingrum, F., Praetya, R. P., & Sari, Y. A. (2017). Image Processing for Rapidly Eye Detection based on Robust Haar Sliding Window. *International Journal of Electrical and Computer Engineering (IJECE)*, 7(2), 823-830.
- [6] Yadav, Y., Walavalkar, R., Yedurkar, A., Suchak, S., & Gharat, S. (2017). Street Light Intensity Controller Using Density Mapping Mechanism.
- [7] Lee, D., Kim, D., Lee, J., Lee, S., Hwang, H., Mariappan, V., ... & Cha, J. (2017). Design of Low Cost Real-Time Audience Adaptive Digital Signage using Haar Cascade Facial Measures. *International Journal of Advanced Culture Technology (IJACT)*, 5(1), 51-57.
- [8] Kim, J., Yu, S., Kim, D., Toh, K. A., & Lee, S. (2017). An adaptive local binary pattern for 3D hand tracking. *Pattern Recognition*, 61, 139-152.
- [9] Karczmarek, P., Kiersztyn, A., Pedrycz, W., & Dolecki, M. (2017). An application of chain code-based local descriptor and its extension to face recognition. *Pattern Recognition*, 65, 26-34.
- [10] Wasim, M., Aziz, A., Ali, S. F., Siddiqui, A. A., Ahmed, L., & Saeed, F. (2017). Object's Shape Recognition using Local Binary Patterns. *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, 8(8), 258-262.
- [11] Liu, L., Lao, S., Fieguth, P. W., Guo, Y., Wang, X., & Pietikäinen, M. (2016). Median robust extended local binary pattern for texture classification. *IEEE Transactions on Image Processing*, 25(3), 1368-1381.
- [12] Tian, S., Bhattacharya, U., Lu, S., Su, B., Wang, Q., Wei, X., ... & Tan, C. L. (2016). Multilingual scene character recognition with co-occurrence of histogram of oriented gradients. *Pattern Recognition*, 51, 125-134.
- [13] Lawgali, A. (2016). Recognition of Handwritten Digits using Histogram of Oriented Gradients.
- [14] Chowdhury, S. A., Uddin, M. N., Kowsar, M. M. S., & Deb, K. (2016, October). Occlusion handling and human detection based on Histogram of Oriented Gradients for automatic video surveillance. In *Innovations in Science, Engineering and Technology (ICISSET), International Conference on* (pp. 1-4). IEEE.
- [15] Tanışık, G., Aselsan, A. S., Güçlü, O., & İkizler-Cinbis, N. Bölüt ve Kontur Özniteliklerini Kullanarak İmgelerdeki İnsan Hareketlerini Tanıma Recognizing Human Actions in Images Using Segment and Contour Features.
- [16] Sharifara, A., Rahim, M. S. M., & Anisi, Y. (2014, August). A general review of human face detection including a study of neural networks and Haar feature-based cascade classifier in face detection. In *Biometrics and Security Technologies (ISBAST), 2014 International Symposium on* (pp. 73-78). IEEE.
- [17] Goerick, C., Noll, D., & Werner, M. (1996). Artificial neural networks in real-time car detection and tracking applications. *Pattern Recognition Letters*, 17(4), 335-343.
- [18] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
- [19] Başer, E., & Altun, Y. (2017, April). Classification of vehicles in traffic and detection faulty vehicles by using ANN techniques. In *Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT), 2017* (pp. 1-4). IEEE.
- [20] Yang, A., Yang, X., Wu, W., Liu, H., & Zhuansun, Y. (2019). Research on feature extraction of tumor image based on convolutional neural network. *IEEE Access*, 7, 24204-24213.
- [21] Passricha, V., & Aggarwal, R. K. (2019). A hybrid of deep CNN and bidirectional LSTM for automatic speech recognition. *Journal of Intelligent Systems*, 29(1), 1261-1274.
- [22] Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2015). Traffic flow prediction with big data: A deep learning approach. *IEEE Trans. Intelligent Transportation Systems*, 16(2), 865-873.
- [23] Jin, Z., Iqbal, M. Z., Bobkov, D., Zou, W., Li, X., & Steinbach, E. (2019). A flexible deep CNN framework for image restoration. *IEEE Transactions on Multimedia*.
- [24] Yang, W. J., Su, Y. S., Chung, P. C., & Yang, J. F. (2017). Moving Object Detection Using Histogram of Uniformly Oriented Gradient. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 11(6), 649-653.
- [25] Kushwaha, A. K. S., Srivastava, S., & Srivastava, R. (2017). Multi-view human activity recognition based on silhouette and uniform rotation invariant local binary patterns. *Multimedia Systems*, 23(4), 451-467.
- [26] Muthevi, A., & Uppu, R. B. (2017, January). Leaf classification using completed local binary pattern of textures. In *Advance Computing Conference (IACC), 2017 IEEE 7th International* (pp. 870-874). IEEE.
- [27] Sharifara, A., Rahim, M. S. M., & Anisi, Y. (2014, August). A general review of human face detection including a study of neural networks and Haar feature-based cascade classifier in face detection. In *Biometrics and Security Technologies (ISBAST), 2014 International Symposium on* (pp. 73-78). IEEE.
- [28] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017, October). Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on* (pp. 2980-2988). IEEE.
- [29] Julina, J. K. J., & Sharmila, T. S. (2017, January). Facial recognition using histogram of gradients and support vector machines. In *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)* (pp. 1-5). IEEE.
- [30] Gkioxari, G., Girshick, R., & Malik, J. (2015). Contextual action recognition with r* cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1080-1088).
- [31] Lu, M., Hu, Y., & Lu, X. (2020). Driver action recognition using deformable and dilated faster R-CNN with optimized region proposals. *Applied Intelligence*, 50(4), 1100-1111.
- [32] Guo, S., Huang, W., Wang, L., & Qiao, Y. (2017). Locally supervised deep hybrid model for scene recognition. *IEEE transactions on image processing*, 26(2), 808-820.
- [33] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740-755). Springer, Cham.
- [34] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* (Vol. 1, pp. I-1). IEEE.
- [35] Papageorgiou, C. P., Oren, M., & Poggio, T. (1998, January). A

- general framework for object detection. In *Computer vision, 1998. sixth international conference on* (pp. 555-562). IEEE.
- [36] Ojala, T., Pietikäinen, M., & Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1), 51-59.
- [37] Günay, A., & Nabiye, V. V. (2011). LBP Yardimiyla Görüntüdeki Kişinin Yaşının Bulunması. *Çankaya University Journal of Science and Engineering*, 8(1), 27-41.
- [38] Shashua, A., Gdalyahu, Y., & Hayun, G. (2004, June). Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. In *Intelligent Vehicles Symposium, 2004 IEEE* (pp. 1-6). IEEE.
- [39] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (Vol. 1, pp. 886-893). IEEE.
- [40] Kim, S., & Cho, K. (2014). Fast Calculation of Histogram of Oriented Gradient Feature by Removing Redundancy in Overlapping Block. *J. Inf. Sci. Eng.*, 30(6), 1719-1731.
- [41] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
- [42] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [43] Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep neural networks for object detection. In *Advances in neural information processing systems* (pp. 2553-2561).
- [44] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016). Tensorflow: A system for large-scale machine learning. In 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16) (pp. 265-283).