IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Low-Power Embedded Systems For Object Recognition: A Deep Learning Paradigm

[1]Deebu U S, [2]Anoop S, [3]Ajeesh S

[1]Assistant professor, [2]Assistant professor, [3]Assistant professor

[1]Electronics and Communication engineering,

[1]College of engineering, Adoor

[2]Electronics and Communication engineering,

[2]College of Engineering, Kottarakkara

[3]Computer Science and engineering

[3]College of engineering, Kottarakkara

Abstract: Image recognition, a crucial facet of computer vision, involves the automated identification and categorization of visual content within images, playing a pivotal role in diverse applications such as medical diagnostics, autonomous vehicles, security systems, and augmented reality, significantly enhancing efficiency and accuracy in various domains. This study explores and compares various object detection and classification methods, incorporating LBP,Haar Cascade, HOG for detection, and for classification CNN, DNN. Hybrid methodologies, including Haar Cascade with CNN, Haar Cascade with DNN, LBP with CNN, LBP with DNN, HOG with CNN, HOG with DNN, were rigorously tested on different embedded systems utilizing the Microsoft COCO dataset. Results revealed that the Haar Cascade with CNN methodachieved the highest recognition success rate at 78.60%, surpassing other methods. These outcomes highlight the efficacy of the Haar Cascade with CNN approach, especially on powerful embedded systems, showcasing its potential for real-time object recognition applications**.**

Index Terms **- Haar Cascade algorithm, Image recognition, Embedded systems, Binary pattern.**

I. INTRODUCTION

Object recognition is a computer vision task that involves identifying and classifying objects within an image or a video. It is a fundamental aspect of image analysis and artificial intelligence, aiming to enable machines to comprehend and interpret visual data similarly to how humans do. The process typically involves the use of sophisticated algorithms and deep learning models trained on large datasets to recognize patterns, shapes, and features associated with various objects. Object recognition finds applications in diverse fields, such as autonomous vehicles, surveillance systems, augmented reality, and robotics, where the ability to accurately identify and understand the surrounding environment is crucial for decision-making and interaction [1].

Object recognition offers several merits, making it a pivotal aspect of computer vision and artificial intelligence applications. Firstly, it enables automation in various industries by allowing machines to understand and interact with their environment. In fields like robotics and manufacturing, object recognition facilitates precise handling and manipulation of objects. Additionally, it enhances security systems through the identification of people and items, contributing to effective surveillance [2]. In the realm of healthcare, object recognition aids in medical imaging, assisting with the diagnosis and analysis of medical conditions. Furthermore, in autonomous vehicles, this technology plays a crucial role in identifying obstacles and ensuring safe navigation. Overall, the merits of object recognition lie in its ability to impart visual understanding to machines, fostering advancements across a spectrum of industries [3].

Existing object recognition algorithms, while powerful, also have some limitations. One limitation is their sensitivity to variations in lighting conditions, which can affect the algorithm's performance in different environments. Additionally, these algorithms may struggle with handling occlusions, where objects are partially or completely hidden from view [4]. Scale variations and changes in viewpoint can also pose challenges, as the algorithms may not generalize well to objects appearing in different sizes or orientations. Another limitation lies in their dependence on annotated datasets, making them less effective when faced with novel or unseen objects. Real-time processing constraints can be an issue, particularly with computationally intensive algorithms, limiting their applicability in certain applications [5]. Furthermore, existing algorithms may have difficulty distinguishing between objects with similar visual features. Continuous advancements in addressing these limitations through research and technological innovation aim to enhance the robustness and versatility of object recognition systems [6].

Embedded systems play a crucial role in the integration of deep learning methods for object recognition, fostering advancements in various applications. By combining the efficiency of embedded systems with the powerful capabilities of deep learning algorithms, there's a significant impact on real-time and edge computing scenarios [7]. These integrated systems allow for on-device processing, minimizing the need for constant connectivity to cloud services and ensuring faster response times. This proves especially beneficial in applications like autonomous vehicles, surveillance cameras, and IoT devices where quick and accurate object recognition is essential [8]. The synergy between embedded systems and deep learning methods not only enhances the computational efficiency of object recognition but also extends the deployment of intelligent vision systems to resource-constrained environments. The surge in portable devices and autonomous systems has led to a growing interest in embedded systems. While there are clear benefits to conducting computations on embedded systems, there are also drawbacks associated with their use [9].In our investigation, we examined hybrid methods such as Haar Cascade with CNN, Haar Cascade with DNN, LBP with CNN, LBP with DNN, HOG with CNN, HOG with DNN, to alleviate the computational burden of classification techniques and enhance object recognition performance [10]. The objective was to decrease processing demands and improve classification accuracy through the implementation of hybrid methods employing pipeline logic, thereby reducing the data size inputted into CNN and DNN. The models underwent training using the Microsoft COCO dataset, and subsequent testing was conducted using images sourced from the same dataset.

II. LITERATURE REVIEW

Zhang et al [11] systematically explores contextual information's role in visual recognition, investigating ten key properties. Psychophysics experiments with human observers assess recognition accuracy by altering context aspects. The proposed biologically-inspired model, employing a two-stream architecture, achieves human-level performance without task-specific retraining. Transparency is ensured through publicly available code and data. Limitations involve defining "full context" and context definition challenges in computer vision databases. The study underscores contextual cues' importance, addressing algorithm vulnerabilities. Introduced benchmarks aim to enhance intelligent computer vision systems by bridging object recognition and scene understanding. Tan et al. [12] addressed poor performance in object detection on long-tailed datasets like LVIS by introducing an equalization loss. Treating positive samples as negative for other categories discourages gradients for tail categories. The method yields notable gains, improving Average Precision by 4.1% and 4.8% for rare and common categories, respectively, demonstrating simplicity and efficacy.

Boukerche et al. [13] explored deep learning techniques for object detection in traffic scenarios crucial for autonomous driving. Reviewing over 100 papers, they categorize generic frameworks and specific applications like vehicle and pedestrian detection. The survey highlights milestone frameworks, discusses metrics, proposes an equalization loss, and identifies research fields. While limitations and datasets specifics are not detailed, it serves as a valuable resource for deep learning-based object detection in traffic scenarios. Patil et al. [14] explored the significance of object identification within deep learning, highlighting its advantages in feature learning compared to traditional methods. The paper provides an overview of conventional object recognition techniques, drawing parallels and distinctions with contemporary deep learning strategies. It delves into the evolution of object recognition, emphasizing methodologies in modern object detection using deep learning. The authors propose a deep learning model to enhance object detection accuracy, but the methodology lacks detailed exposition, and limitations are not explicitly addressed. The paper underscores challenges in object identification and suggests integrating modern embedded systems into

deep learning for enhanced opportunities. However, the absence of specific datasets and a comprehensive exploration of model performance metrics limits the paper's thoroughness.

Bhandari et al. [15] provided a valuable review of deep learning systems for navigational tools aiding the visually impaired, offering a comprehensive framework for future research. The methodology involves comparing current systems and compiling a taxonomy of essential features. This purportedly first-of-its-kind critical analysis highlights challenges in object detection for the visually impaired. Convolutional Neural Networks (CNN) emerge as efficient for feature extraction and navigation, evaluated across twelve systems. Limitations include the complexity of vision impairment and the need for tailored solutions. The research emphasizes multifunctionality in systems, addressing data, feature extraction, classification, object detection, and recognition. While acknowledging promising work with CNN and Fully Convolutional Networks (FCN), the paper underscores challenges in meeting diverse needs and enhancing detection of moving objects. Dataset limitations are not explicitly mentioned, encouraging further exploration of these critical issues in future research.

III. MATERIALS AND METHODS

This section provides descriptions of the Haar Cascade, LBP, and HOG object detection algorithms, as well as the CNN anddeep neural network classification algorithms employed in this research.

3.1 Dataset Description

In our research, dataset from the Microsoft COCO were employed, specifically selecting 20 object classes to evaluate the performance of hybrid methods. The dataset comprises both labeled and unlabeled images, with categorization carried out by Microsoft Bing and real individuals. To assess algorithmic success, images not utilized in algorithm training were reserved for testing purposes. The dataset was split, with 80% used for training and the remaining 20% allocated for testing.

3.2 Convolutional Neural Network

The convolutional neural network (CNN) is a deep learning architecture composed of multiple layers. Within this structure, the Convolution layer utilizes moving windows, referred to as filters or cores, over the input matrix. These filters, randomly generated, are applied to windows within the image matrix, resulting in the emergence of a new feature matrix. CNN possess two noteworthy characteristics. Initially, the learned patterns in convolutional neural networks exhibit a consistent direction, allowing them to recognize patterns across various parts of an image, fostering generalization with limited data. Additionally, these networks have the capacity to acquire spatial hierarchies. For instance, the first convolution layer focuses on localized patterns like corners, while subsequent layers learn more extensive patterns than those discerned by the initial layer. This hierarchical learning aids in effectively grasping intricate visual concepts. The primary objective of this research was to enhance the performance of CNN in object recognition. This was achieved by employing other object detection methods to diminish the complexity of images entering the neural network.

3.3. Deep Neural Network

The DNN is a framework employed for object detection, leveraging its numerous hidden layers to achieve more effective outcomes compared to artificial neural networks. In our research, a DNN structure was implemented with 28 * 28 neurons for the input layer, 20 neurons for the output layer, and nine hidden layers. Within our study, DNN played a key role in both object detection and classification. Furthermore, in hybrid methodologies, DNN was specifically utilized for object classification purposes.

3.4 Algorithm for Haar Cascade

For object detection in images or video Haar Cascade algorithm is used in ML based approach It operates by training on positive and negative samples to create a classifier capable of identifying specific objects or features. The algorithm employs a cascade of Haar-like features, which are rectangular filters that analyze image regions for variations in intensity. These features are applied to the input image in a cascade fashion, where each stage refines the search for the object, filtering out non-relevant regions efficiently. Haar Cascade is known for its speed and efficiency, making it suitable for real-time applications. It has been widely used in various domains, including face detection, where it became particularly popular due to its accuracy and real-time performance.

The Haar Cascade algorithm, introduced by Paul Viola and Michael Jones , serves to locate and track objects in images or videos. The process begins with converting the image from RGB to gray level, followed by sequential steps involving integral image creation, Haar feature selection, calculation of Haar features, Cascade and Adaboost. These steps collectively determine the specific Haar features to be employed. In our research, we utilized the Haar feature windows depicted in Figure 1 for analysis and implementation.

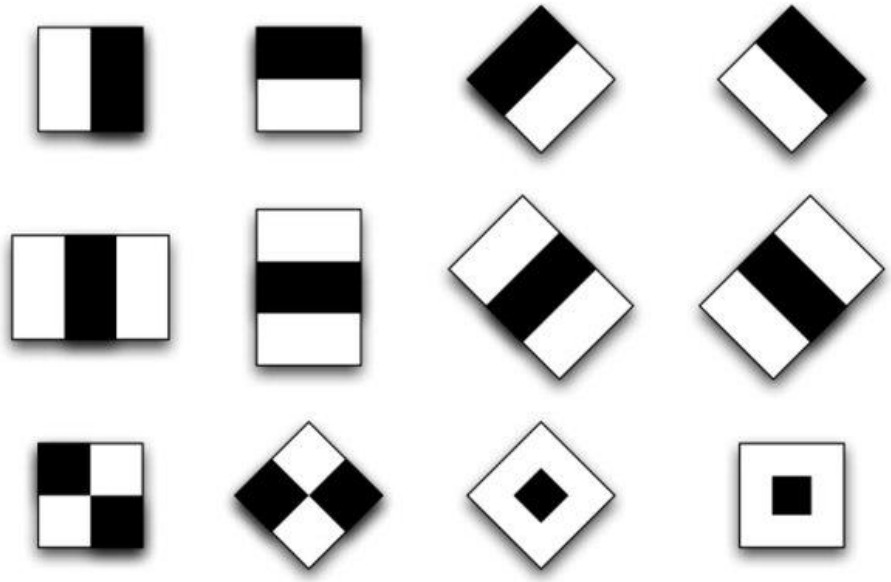

Fig.1: Haar features

To enhance the operational efficiency of the algorithm, the integral image is generated. In the integral image I(x, y) denoted the pixel values, where (x, y) represents the point within the image. Here, m corresponds to the x axis, and n as y axis. The calculation is illustrated in (1).

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

Figure 2 illustrates the of pixel value calculation for a selected window in the image, as outlined in equation (2).

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

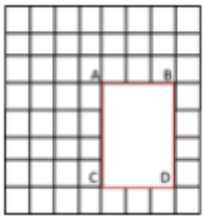

Fig. 2: Integral value of Window

A critical component in training the Haar Cascade algorithm is the utilization of a training dataset, which includes positive images containing the target object and negative images devoid of the object of interest. Positive images are subjected to object detection using the windows illustrated in Figure 2. The threshold for distinguishing brightness and darkness can be established by either calculating the average pixel value or set manually.

During the training process, the chosen feature is assessed by comparing it to a predefined threshold value. Equation (3) illustrates the computation for any Haar feature (f).

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

Employing the derived f value, the calculation of each weak classifier for every image is portrayed in equation (4), with 'j' in the equation representing each Haar feature.

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

The calculation of the weight (Wi) for both positive and negative images is demonstrated in equation (5).

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

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

Subsequent to the initial cycle, the weights revert to their initial values, and Haar features are computed utilizing equation (4). In equation (7), wherein positive images are assigned a value of 1, and negative images a value of 0. The weak classifier is established by selecting the Haar feature with the minimum failure value.

$$\varepsilon_t = \sum_i W_i \left| h_j(x) - y_i \right| \quad (7)$$

As the algorithm progresses to the next cycle, weights undergo an update following the guidelines outlined in equation (8).

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

If the i-th image is correctly classified, ei = 0; if it is misclassified, ei = 1. The β value is computed according to equation (9).

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

Utilizing a single weak classifier makes it challenging to achieve the desired rate of selecting positive images and excluding negatives.

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

The weight assigned to the weak classifier is not proportional to the failure in classification. The resulting strong classifier formed by combining weak classifiers undergoes scrutiny through the control mechanism outlined in equation (11).

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

Fig.3: Structure of Cascaded classification

As illustrated in Figure 3, an image within the Cascade classifier is considered negative if any weak classifier rejects it. Conversely, if the image successfully passes through all classifiers, it is deemed positive, leading to the execution of object detection.

3.5 Local Binary Pattern

The conversion of the disparity between the pixel and its neighbors into a binary representation is accomplished through the utilization of the step function u(x) for labeling each pixel.

$$LBP_{P,R}(x_c)=\sum_{p=0}^{P-1}u(x_p-x_c)2^p$$
$$u_y=\begin{cases}1 & y\geq o\\ 0 & y<0\end{cases} \quad (12)$$

Combining neighboring values, the LBP value functions are exemplified in Figure 4.

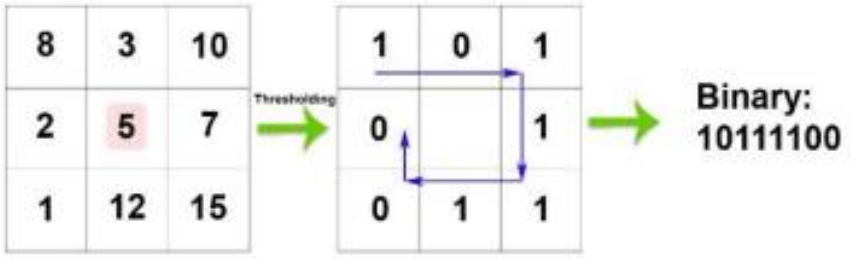

Fig.4: LBP Operator

3.6 Histogram Oriented Gradients

Histogram of Oriented Gradients (HOG) is a feature descriptor widely employed for object recognition tasks in computer vision. Introduced by Navneet Dalal and Bill Triggs in 2005, HOG represents an image by calculating the distribution of gradient orientations in localized regions. The method involves dividing the image into small cells, computing gradient magnitudes and orientations within each cell, and then creating histograms of these orientations. These histograms capture the dominant directions of edges and gradients within the image. To enhance robustness, the HOG descriptor incorporates normalization techniques to address variations in lighting and contrast. Commonly used in conjunction with support vector machines (SVMs), HOG has proven effective in detecting objects within images, particularly in pedestrian detection, face recognition, and general object recognition tasks. Its ability to capture shape and edge information while being computationally efficient makes HOG a valuable tool in various computer vision applications.

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

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

3.7 Structure of Hybrid Object Recognition Algorithm

Through these procedures, hybrid structures such as Haar Cascade with CNN, Haar Cascade with DNN, LBP with CNN, LBP with DNN, HOG with CNN, HOG with DNN, were formulated. The object of interest is identified and isolated in the image using the chosen detection method, and the segmented object is then inputted into either CNN or DNN. The proposed method is shown in figure 5. Consequently, the image size is reduced with the segmented image, leading to a decrease in density.

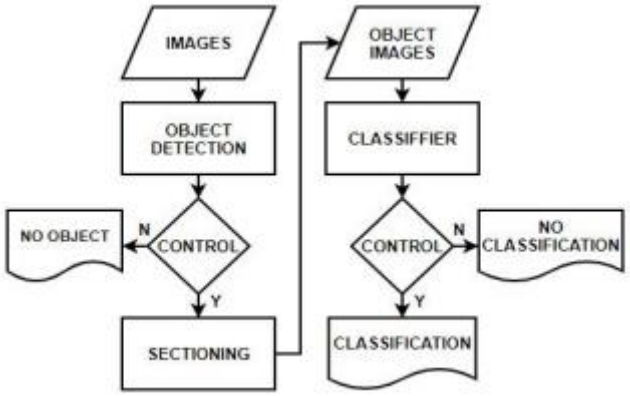

Fig.5: Proposed hybrid methods

Figure 6 depicts the workflow of detecting an image using object detection methods, followed by the segmentation and transmission of the identified object to the classifier.

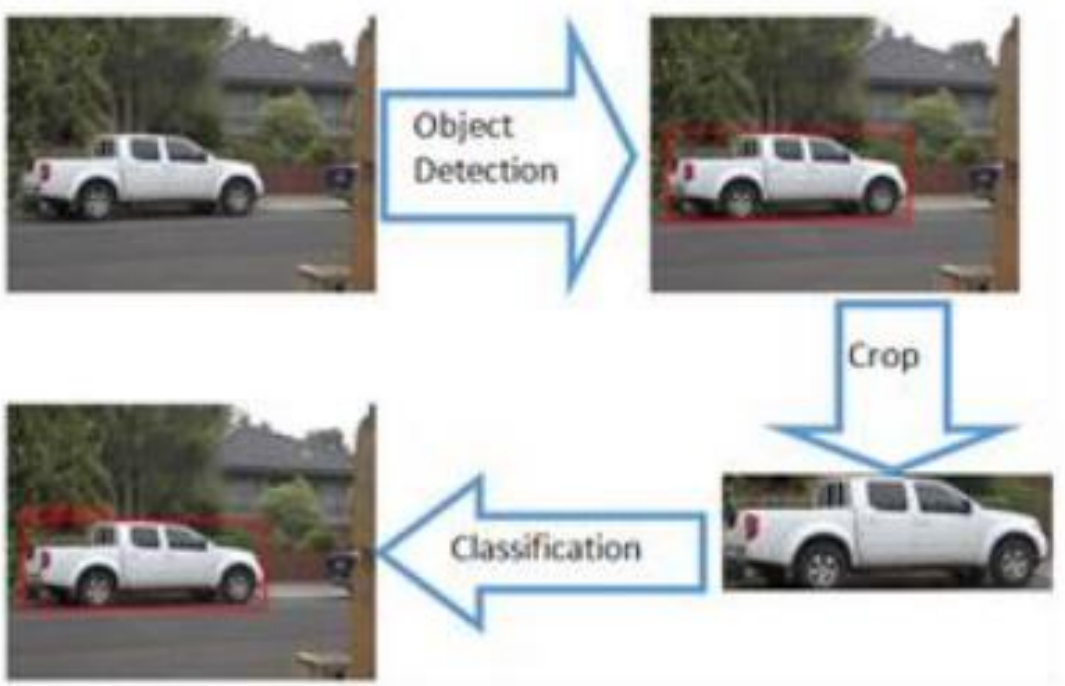

Fig. 6: Object detection

3.8 Embedded systems

Embedded systems for object recognition refer to the integration of specialized hardware and software components designed to perform real-time object detection and identification within a constrained environment. These systems are tailored for deployment in devices with limited computational resources, such as smartphones, cameras, drones, and IoT devices. The key challenge in developing embedded systems for object recognition lies in achieving efficient and accurate processing with minimal power consumption and memory usage. Various architectures, including field-programmable gate arrays (FPGAs), graphics processing units (GPUs), and custom application-specific integrated circuits (ASICs), are often employed to accelerate the computationally intensive tasks associated with object recognition. Additionally, lightweight machine learning models, optimized algorithms, and compression techniques are implemented to strike a balance between accuracy and resource efficiency. Embedded systems for object recognition find applications in diverse fields, including smart surveillance, autonomous vehicles, and augmented reality, enabling devices to interpret and respond to their surroundings in real-time.

Employing TensorFlow, developed by Google, the embedded systems utilized in our study were equipped to support the TensorFlow library. For a comprehensive overview of each embedded system's features, refer to Table 1.

TABLE 1. SYSTEM PROPERTIES

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

IV. RESULT AND DISCUSSION

Conducting tests aimed at discerning the most effective configuration, various setups within the CNN and deep learning classification algorithms employed in our study were thoroughly evaluated. The average success of the system is shown in Table 2.The object detection and classification methods, namely Haar Cascade with CNN, Haar Cascade with DNN, LBP with CNN, LBP with DNN, HOG with CNN, HOG with DNNwere evaluated using the dataset employed in our study.The testing each image with all methods, as depicted in Figure 7.

Fig. 7: Multi-object recognition

Table 2. Success Rate Of The Approach

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

Derived from the results, the hybrid method Haar Cascade+CNN, leveraging the CNN classifier, demonstrated a recognition success rate of 78.60%, outperforming the other methods applied in the study. These average run times are detailed in Table 3.

Table 3. Average Recognition Time Of Each Method

	CNN	Haar + CNN	LBP + CNN	HOG + CNN	Haar + DNN	LBP + DNN	HOG + 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 TINKERBOARD	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

Based on Table 3, tests conducted indicated that the LBP with CNN method performed in 0.487 seconds, exhibiting a shorter processing time than other methods and ranking first. On the Nvidia Jetson TX2 system, the Haar Cascade+CNN method exhibited the shortest operating time at 0.1303 seconds. Tests on the Asus Tinker Board system were performed in 0.2458 seconds, presenting faster results than the LBP+CNN method. When examining the Raspberry Pi 3 B+ system, which has comparatively weaker hardware, the CNN method operated most rapidly at 0.7159 seconds. Overall, the Nvidia Jetson TX2 system outperformed others, especially due to its GPU advantages.

V. CONCLUSION

In conclusion, this study employed a comprehensive evaluation of object detection and classification methods, including Haar Cascade,HOG, LBP for detection, and for classification CNN and DNN. Hybrid approaches, such as Haar Cascade with CNN, Haar Cascade with DNN, LBP with CNN, LBP with DNN, HOG with CNN, HOG with DNN were tested using the Microsoft COCO dataset across various embedded systems, including Nvidia Jetson TX2, Asus Tinker Board, and Raspberry Pi 3 B+. The results indicated that the Haar Cascade+CNN hybrid method, utilizing the CNN classifier, achieved the highest recognition success rate of 78.60%, outperforming other methods. Additionally, on the Nvidia Jetson TX2 system, this method exhibited the shortest processing time at 0.1303 seconds, demonstrating superior efficiency. These findings underscore the effectiveness of the Haar Cascade+CNN hybrid method, particularly on powerful embedded systems like Nvidia Jetson TX2, showcasing its potential for real-time object recognition applications.

REFERENCES

[1] Zhang, M., Tseng, C., &Kreiman, G. (2020). Putting visual object recognition in context. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 12985-12994).

[2] Tan, J., Wang, C., Li, B., Li, Q., Ouyang, W., Yin, C., & Yan, J. (2020). Equalization loss for long-tailed object recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11662-11671).

[3] Pathak, A. R., Pandey, M., &Rautaray, S. (2018). Application of deep learning for object detection. Procedia computer science, 132, 1706-1717.

[4] Alom, M. Z., Hasan, M., Yakopcic, C., Taha, T. M., &Asari, V. K. (2020). Improved inception-residual convolutional neural network for object recognition. Neural Computing and Applications, 32, 279-293.

[5] Nguyen, N. D., Do, T., Ngo, T. D., & Le, D. D. (2020). An evaluation of deep learning methods for small object detection. Journal of electrical and computer engineering, 2020, 1-18.

[6] Alom, M. Z., Hasan, M., Yakopcic, C., Taha, T. M., &Asari, V. K. (2021). Inception recurrent convolutional neural network for object recognition. Machine Vision and Applications, 32, 1-14.

[7] Sobrinho, A. S. F. (2020). An Embedded Systems Remote Course. Journal of Online Engineering Education, 11(2), 01-07.

[8] Mao, H., Yao, S., Tang, T., Li, B., Yao, J., & Wang, Y. (2016). Towards real-time object detection on embedded systems. IEEE Transactions on Emerging Topics in Computing, 6(3), 417-431.

[9] Jagannathan, S., Desappan, K., Swami, P., Mathew, M., Nagori, S., Chitnis, K., ... & Jain, A. (2017, January). Efficient object detection and classification on low power embedded systems. In 2017 IEEE International Conference on Consumer Electronics (ICCE) (pp. 233-234). IEEE.

[10] Javed Mehedi Shamrat, F. M., Majumder, A., Antu, P. R., Barmon, S. K., Nowrin, I., & Ranjan, R. (2022). Human face recognition applying haar cascade classifier. In Pervasive Computing and Social Networking: Proceedings of ICPCSN 2021 (pp. 143-157). Springer Singapore.

[11] Zhang, M., Tseng, C., &Kreiman, G. (2020). Putting visual object recognition in context. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 12985-12994).

[12] Tan, J., Wang, C., Li, B., Li, Q., Ouyang, W., Yin, C., & Yan, J. (2020). Equalization loss for long-tailed object recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11662-11671).

[13] Boukerche, A., & Hou, Z. (2021). Object detection using deep learning methods in traffic scenarios. ACM Computing Surveys (CSUR), 54(2), 1-35.

[14] Patil, S. M., Raut, C. M., Pande, A. P., Yeruva, A. R., &Morwani, H. (2022). An efficient approach for object detection using deep learning. Journal of Pharmaceutical Negative Results, 563-572.

[15] Bhandari, A., Prasad, P. W. C., Alsadoon, A., &Maag, A. (2021). Object detection and recognition: using deep learning to assist the visually impaired. Disability and Rehabilitation: Assistive Technology, 16(3), 280-288.