Co-Design of Hardware and Software for Facial Expression Recognition Using Xilinx Zynq SoC

Ahmed Chiheb Ammari, Lazhar Khriji, Medhat Awadalla, Rami Al Hmouz
Sultan Qaboos University
Muscat, Oman
chiheb@squ.edu.om, lazhar@squ.edu.om, medhatha@squ.edu.om, r.alhmouz@squ.edu.om

***Abstract*— This paper presents the development of an advanced embedded computer vision system tailored for facial expression recognition. The system utilizes the Viola-Jones algorithm for accurate face detection in images, combined with a hybrid approach for feature extraction leveraging Local Binary Pattern (LBP) and Histogram of Oriented Gradient (HOG) features. Aiming at optimizing the performance and reducing the complexity of the system, the Relief algorithm is employed for effective feature selection, focusing on the most significant features to enhance processing time efficiency. A Zynq 7020 SoC platform is elected for system prototyping. This platform effectively integrates an ARM Cortex A9 dual-core communicating with FPGA logic. HW/SW Co-Design is developed starting from the profiling results of the original classification algorithm. This design significantly enhances the system's performance, achieving a real-time classification rate of 27.5 frames per second. This represents an approximately 18 times increase in speed compared to the original program implementation.**

I. INTRODUCTION

To enhance human-machine interactions, computers are being equipped to measure and interpret human emotions. The ability of computer technology to recognize facial expressions plays a central role in understanding a person's emotional state and intentions, which is essential in fields such as intelligent advertising systems, safe driving, and human-computer interaction [1]. A facial expression recognition system, by analyzing extracted facial features, can significantly contribute to the comprehension of human emotions. This understanding is vital in improving communication, decision-making processes, and behavioral insights [2].

Face detection, feature extraction, and facial expression classification are the three fundamental components of a typical facial expression recognition system. The process begins with face detection, which differentiates between facial and non-facial regions. This step produces a rectangular boundary around the head, pinpointing the location of the face. The next critical phase is feature extraction, which significantly influences the accuracy and is essential to the overall success of the recognition process [3]. If the features extracted are insufficient or inadequate, the system may struggle to accurately recognize facial expressions. Finally, neural networks can be employed as classifiers to identify and categorize different facial expressions.

The effectiveness of facial expression recognition is influenced by various factors, including complex backgrounds, lighting conditions, facial features, and types of classification methods [4]. Faces inherently exhibit significant variability in appearance, presenting challenges for accurate facial expression recognition. Therefore, achieving high accuracy and fast processing time in both identifying faces and discerning emotions is a critical aspect of computer vision systems.

This paper presents the development of an intelligent embedded system designed for recognizing facial expressions. Our focus is on recognizing facial expressions from static images of individual subjects. We assess seven essential facial expressions that represent universal emotions: disgust, sadness, happiness, fear, anger, surprise, and neutral expression [5]. For effective feature extraction, the system adopts a hybrid approach combining the Local Binary Pattern (LBP) [6] and Histogram of Oriented Gradient (HoG) [7]. An Artificial Neural Network (ANN) based on the back-propagation algorithm [8] is utilized for the classification process.

We conducted our experiments using the Japanese Female Facial Expression (JAFFE) database, which contains 210 grayscale images of Japanese women depicting the six basic emotions plus a neutral expression [9]. The results from our study demonstrate the efficiency of the developed hybrid approach, showing its comparative strengths against existing state-of-the-art studies. Additionally, we explored accelerating the most computationally intensive tasks. This is part of our efforts for embedded prototyping of the most effective classifier, aiming to achieve a minimum real-time classification performance of 20 frames per second.

The organization of this paper is as follows. Following this introduction, the next section reviews the architecture of the proposed system. Section 3 presents the experiments and discusses the performance characterization of the classifier. The embedded systems prototyping and the evaluation of the real-time performance for the proposed classifier are covered in section 4. The paper concludes with a summary of the key results.

II. THE PROPOSED SYSTEM ARCHITECTURE

The block diagram of the proposed system architecture is structured into four crucial stages. In the pre-processing stage, the facial region of the image is cropped, filtered, and resized to a dimension of 100x100 pixels. For extracting features, we use LBP and HOG descriptors to capture the shape and texture characteristics of the face [10]. The feature selection is carried out using a Relief algorithm [11, 12], which identifies the most significant features from the HOG and LBP datasets. An Artificial Neural Network then processes these selected features for the accurate classification of facial expressions.

A. Input Image

The training of the system is conducted using the JAFFE dataset [9], which includes 210 images of seven different facial expressions from ten individuals. The categorized expressions, illustrated in Fig. 1, include anger, happiness, sadness, disgust, surprise, fear, and a neutral state, with each category containing 30 images. 80% of data representing the respective emotions are selected for training and 20% for testing.

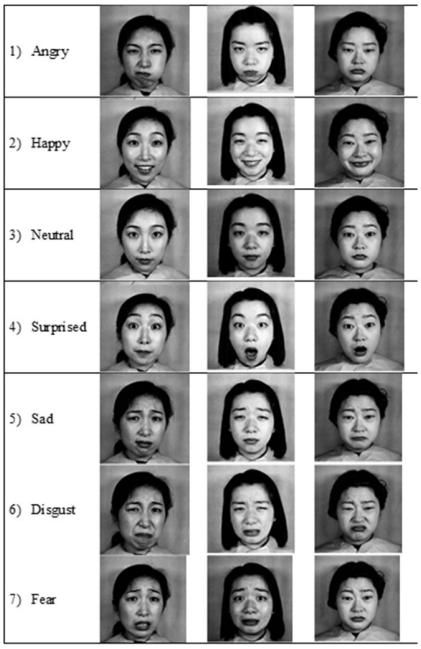

Fig. 1. Example of expressions from the JAFFE database

B. Pre-Processing

The stage of the pre-processing works to retrieve a series of face images from the input data. These images must be standardized in terms of intensity, size, and shape. The process begins with the extraction of facial patterns from the image, utilizing the algorithm of the Viola-Jones. This face-detection algorithm is known for its high detection rate in grayscale images [13]. It identifies faces and marks them with bounding boxes, as depicted in Fig. 2. These marked areas define the region of interest for the subsequent stages.

The next step involves cropping the image to focus solely on the region of interest, thereby removing any irrelevant portions. In our work, the images are resized from 256x256 to 100x100 pixels. Additionally, a Gaussian filter and normalization techniques aimed at mitigating the effects of illumination, lighting, and noise are adopted to enhance the quality of the input image.

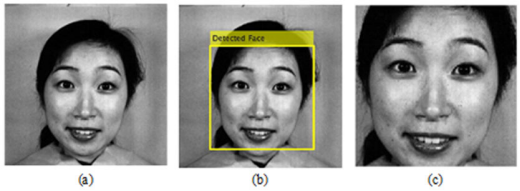

Fig. 2. Original image (a), Face detected (b) Face Cropped (c)

A. Image Acquisition System

The image acquisition system used in this study consisted of three main components: a personal computer, RGB color camera (model: EOS 1100D, Canon, Taiwan, resolution of 4272× 2848 pixel) and two fluorescent lights. An A4 white paper is used as image background and each date sample is manually positioned at 15 cm from the camera. The images are taken using the camera's self-timer mode with three images per sample. To remove any possible noise that may happen during the snap-shots, the average between three images for each sample is considered. The database of our dataset is then created for further processing.

C. Features Extraction

The second module is dedicated to extracting LBP and HOG features, effectively encoding the facial components. These extracted features are then combined into a unified feature vector. Subsequently, this feature set is forwarded to the next optimal feature selection stage.

1) Local Binary Patterns (LBP)

Originally introduced by Ojala et al. [14] for texture classification, LPB descriptors have since been adapted for various other fields. The main strengths of LBP lie in its resistance to monotonic changes in gray level and its computational simplicity. It is particularly robust against variations in illumination, easy to implement, and has a low computational complexity. As illustrated by the histograms in Fig. 3, LBP simplifies the classification algorithm by reducing the gray levels and their associated probabilities.

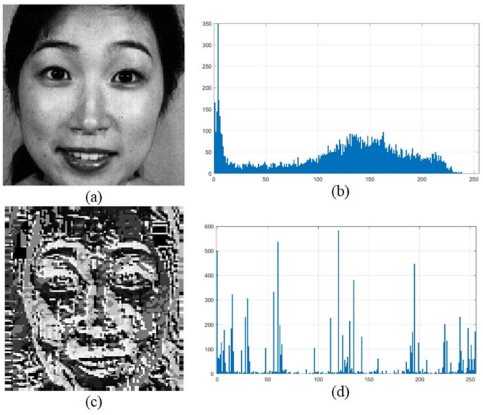

Fig. 3. (a) Face image detected, (b) HoG of (a), (c) LBP (d) HOG of (c).

Essentially, LBP is defined as a sequence of binary comparisons of pixel intensities, comparing the central pixel with its surrounding neighbors. For a given pixel x_i, the LBP computation is conducted as follows:

$$LBP(xi) = \sum_{i=1}^{P} s(Ii - Ic)\, 2^i \quad (1)$$

Where:

$$s(I_i - I_c) = \begin{Bmatrix} 1 & if & I_i - I_c > 0 \\ 0 & if & I_i - I_c < 0 \end{Bmatrix} \quad (2)$$

I_c is the center pixel grey value, I_i is the neighbor pixel grey value, and P is the number of neighbors involved.

2) Histogram of Oriented Gradients (HOG)

HOG is a popular technique in object detection, particularly effective in pedestrian detection. HOG works by quantifying occurrences of gradient orientations within a local section of an image. The underlying principle is that the distribution of local gradient intensity and direction can effectively represent the appearance and shape of an object in that locality [7]. To utilize HOG descriptors, the first step involves counting edge occurrences in an image's local neighborhood. This process involves dividing the image into small, interconnected regions. HOG descriptors are then applied to encode the facial components. In our experiments, we set the cell size to 16x16 pixels, with an orientation range of 0 to 180 degrees. Fig. 4 illustrates the HOG features extracted for a cell size of 16x16. Given this cell size, the feature vector produced has a length of 900.

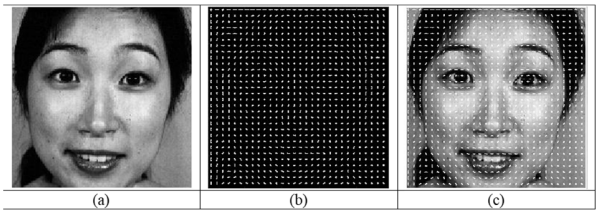

Fig. 4. (a) Detected, face (b) HOG for a cell size 16x16, (c) HOG applied to face image.

D. Feature Selection

The feature selection module reduces the dimensionality of the data and identifies the most critical features for accurately predicting facial expressions. This module is tasked with selecting an essential subset from the originally extensive, high-dimensional data. For this purpose, we employ the rank importance of predictors, known as the Relief algorithm, to execute feature selection [12]. Renowned for its effectiveness and simplicity, the Relief algorithm is a popular method for estimating feature weight and selecting the most relevant features. The process involves two key steps: firstly, features are ranked according to a consistent evaluation criterion, and secondly, the optimal set of features that meet this criterion are selected. Additional information about the Relief algorithm can be found in [11].

In the hybrid system developed in this study, we select five features from both the HOG and LBP. Consequently, these selections combine to form an input feature vector of ten features for the classification module. Following this, the feature selection policy, employing the Relief algorithm, is applied to this combined feature vector. The aim is to isolate the most significant five features during each iteration.

E. Artificial Neural Network (ANN) classifier

The ANN is visualized as a network comprising layers of neurons (computing units), where each neuron is interconnected with others in the adjacent layers via weighted links [15]. Such a network is defined by data input and output vectors, a weighting matrix, and a transfer function. Neurons in every layer receive inputs from the previous layer, utilize the weighted links and transfer function to process these inputs, and then determine the outputs for the succeeding layer's neurons. The network's topology is a critical factor in its overall performance. In our study, we developed various ANN structures (as shown in Fig. 5), which differ in the number of layers and input neurons (5, 10, 15, 20, and 25).

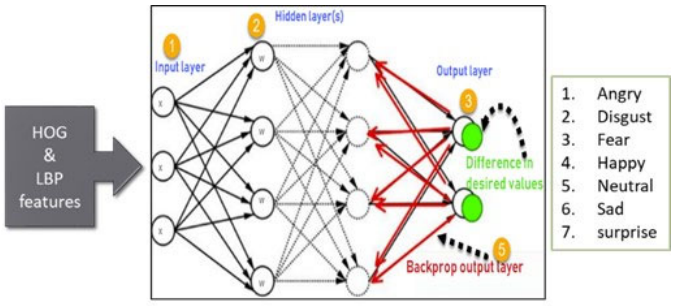

Fig. 5. Backpropagation Neural Network Architecture

III. Performance Evaluation

To design and implement the proposed facial expression recognition system, we utilized MATLAB and a computer equipped with an Intel® Core™ i7-4510U CPU @2.40GHz with 16GB of RAM. A classification system can be characterized using various performance metrics, with accuracy being the chosen metric for evaluating our system. Accuracy is defined as the proportion of correct predictions (True Positive TP and True Negative TN) out of the total number of predictions made (TP+TN+FP+FN) as given by equation 3.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

In this research, the ANN was trained and subsequently tested 30 times, each with new validation data. To assess the classifier's performance, we calculated a Figure of Merit (FoM) performance indicator as defined by equation 4. This indicator incorporates the average processing time, the accuracy, and the number of features. This approach allows us to evaluate the performance of the classifier while also considering the trade-offs with complexity.

$$FoM = \frac{Accuracy}{Number\ of\ Features + Time} \quad (4)$$

We conducted two experiments to analyze both complexity and performance. First HOG and LBP features are used independently. The outcomes of these tests, presenting the prediction performance of HOG and LBP features separately, are summarized in Table I. The results indicate that, on a statistical average, the prediction accuracy achieved using HOG features surpasses that of the LBP feature.

TABLE I. Performance results from HOG and LBP after selecting features.

Methods		LPB	HOG
Classes	*3*	100% (15F)	100% (5F)
	5	80% (15F)	87% (15F)
	7	38% (20F)	70% (15F)

The second experiment involved utilizing a combination of LBP and HOG features. Here, two different feature selection strategies were implemented. For the hybrid system involving seven facial expression classes, accuracy and merit indicator values were derived, as shown in figures 6 and 7 for both feature selection methods. The results indicated that the second feature selection strategy, which picks the optimal combined HOG and LBP feature vector, yielded higher accuracy and merit values. The highest Figure of Merit (FoM) values are achieved when selecting the top five features. This outcome validates the efficacy of our approach in selecting the most significant hybrid features from LBP and HOG to reduce the processing time complexity.

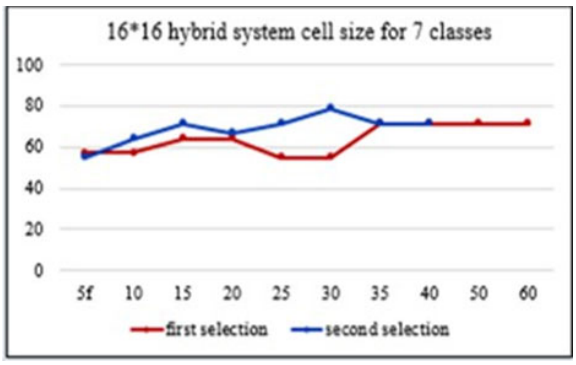

Fig. 6. Hybrid System Accuracies for Seven Classes Output

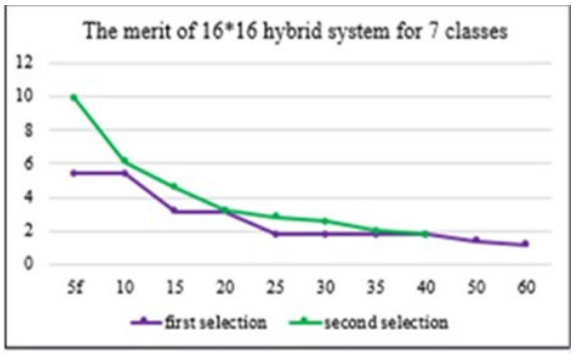

Fig. 7. Hybrid System FoM for Seven Classes Output

In conclusion, the experiments conducted affirm the proposed hybrid approach's capacity to deliver high performance of around 88 % accuracy with reduced complexity in terms of processing time and feature count.

IV. EMBEDDED SYSTEM CO-DESIGN

This section discusses the embedded system design for our proposed facial expression recognition system. There are various platforms suggested in existing literature for such implementations, with Field-Programmable Gate Arrays (FPGAs) being particularly noted for their processing time efficiency and low power consumption [15]. An FPGA can significantly speed up the feature extraction and learning phases of classification through various design techniques. However, a drawback of using FPGAs is the extended development time required for processing tasks with high data dependency and irregular control flows, which are more efficiently handled by programmable processors [17]. To strike a balance between flexibility in system development and enhanced operational capabilities, we have chosen the ZYBO Z7 development board [18]. This board features the Zynq 7020 System on Chip (SoC), which integrates a dual-core ARM Cortex A9 processor and 1GB DDR3 memory, combined with an Artix-7 FPGA. This integration leverages the strengths of both FPGA and processors in a single chip. Nevertheless, distributing the workload between the FPGA and the ARM processors requires careful design to achieve optimal real-time performance [19].

To facilitate the development of our system's heterogeneous architecture, we have opted for the Xilinx Vitis unified design framework [20]. This framework simplifies the process of designing systems with mixed hardware and software components. With Vitis, developers can select specific functions from a C/C++ application for acceleration. These selected functions are then processed using a high-level synthesis (HLS) tool, which generates corresponding accelerators for the FPGA. Vitis allows for multiple synthesis optimizations through dedicated HLS pragmas. Furthermore, Vitis streamlines the integration of these FPGA accelerators into the system. It manages the data exchange between the programmable logic and the processing system automatically, ensuring seamless operation. For the present Software/Hardware (SW/HW) Co-Design for the facial expression recognition system, Vitis version 2021.3 is employed [20]. This tool significantly aids in the efficient and effective development of our proposed system.

A. Application Profiling on ARM

The original MATLAB code for the proposed facial expression recognition system has been transformed into a C++ program, designed to be compatible with Vitis. DDR3 memory is utilized for storing input data image data and feature descriptors. The process begins with allocating continuous memory for DMA transfers, followed by executing the classification modules.

A crucial initial step in the design flow is the ARM CPU execution profiling of this program. The objective is to pinpoint compute-intensive bottlenecks that are potential candidates for migration to hardware accelerators. Accelerating these compute-intensive functions is anticipated to significantly enhance overall system performance. Fig.8 illustrates the profiling results for the system. Feature extraction using HOG and LBP emerges as the most computationally demanding tasks, accounting for 49.4% processing time. The preprocessing and Gaussian filtering is also significant, consuming 38.6% of the time. Computing the remaining functions, including feature selection and classification, represents a 12% overhead.

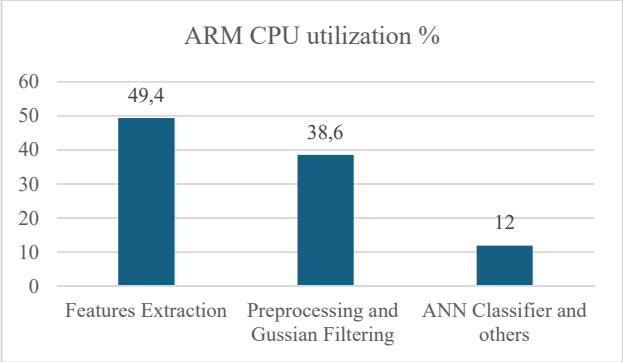

Fig.8. Profiling results of proposed facial recognition system on ZYBO Z7

B. HW/SW Codesign

In light of the obtained profiling results, we selected the feature extraction, preprocessing, and Gaussian Filtering modules for synthesis as hardware accelerators. To optimize memory usage, image data are streamed to dedicated row buffers, thus avoiding the need to store the entire image and reducing the use of BRAM memory. The HLS::stream class within Vivado HLS (a high-level synthesis tool integrated into the Vitis unified design framework) facilitates the implementation of these buffers.

For the accelerated modules, each argument interface is defined properly using HLS Pragmas [21]. To maximize data transfer efficiency, AXI DMA in scatter-gather mode is adopted. This mode manages the data flow between the FPGA and DDR memory, and in the opposite direction as well. Additionally, the data exchange between the hardware accelerators and the ARM core is synchronized through a data motion network, which is automatically generated by Vivado HLS. This network can operate at various frequencies, independent of the frequencies of the processing elements it connects [22]. For the developed hardware design, the hardware-accelerated feature extraction, preprocessing, and filtering modules are set to operate at a maximum feasible frequency of 166.67 MHz.

Both software and hardware computations utilize fixed-point arithmetic for efficiency. Moreover, to optimize the high-level synthesis of these modules, loops within the hardware-accelerated modules are unrolled and pipelined, applying appropriate HLS pragmas. This approach is aimed at maximizing the performance and efficiency of the synthesized blocks.

C. Performance Evaluation

The Vitis 2021.3 framework was employed to evaluate the FPGA usage of resources, with the results presented in TableII. Running, solely on ARM, the original sequential C++ program processed one image in 0.651 seconds, equivalent to 1.53 frames per second (fps). However, integrating the feature extraction, preprocessing, and filtering modules into the FPGA significantly enhanced the system's performance, achieving 27.5 fps. This marks an approximately 18-fold increase in speed compared to the original implementation of the program.

TABLE II. FPGA RESOURCE UTILIZATION FOR THE HW/SW CO-DESIGN ON ZYNQ 7020 SOC

Module	LUTs	FFs	DSPs	BRAMs
Features description	3154	5163	5	5
Gaussian Filtering and Preprocessing	3983	5692	6	4
Data motion Network	2225	8153	0	19
Total	9362 (18%)	19008 (19%)	11 (4%)	28 (20%)

Fig. 9 provides a detailed comparison of the execution times for each module, both before and after acceleration. Accelerating the feature extraction module achieved a speed increase of 51.62 times compared to its software counterpart. Similarly, the preprocessing and Gaussian filtering module acceleration resulted in a 57.6 times speedup. These significant speed improvements, achieved using Vitis, come with a manageable programming overhead.

However, it's important to note that executing each accelerated module entails considerable data transfer costs to and from the DDR memory. This is especially true in our HW/SW Co-Design, where multiple accelerated modules necessitate frequent data transfers between the DDR and FPGA. This process introduces a performance overhead, as DDR memory accesses are required for every hardware accelerator. With high-resolution images, this overhead is more significant and represents a considerable limitation of the high-level synthesis (HLS).

In conclusion, our system's HW/SW Co-Design attained a real-time performance of 27.5 fps, which is deemed acceptable. Nonetheless, performance could have been enhanced if data could be streamed directly from the filtering and preprocessing module to the feature extraction module, eliminating the need for transfers via DDR memory.

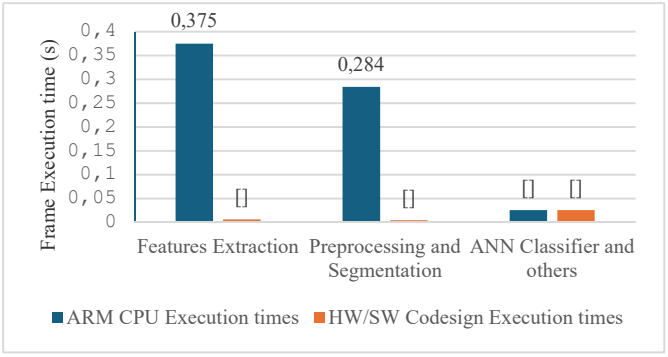

Fig. 9. Comparison of Execution Times Between HW/SW Co-Design and ARM CPU for Each Module

V. Conclusion

This paper introduces a specialized embedded computer vision system designed specifically for facial expression recognition. The system employs the Viola-Jones algorithm for face detection in images. For feature extraction, LBP and HOG approaches are adopted. Additionally, the Relief algorithm is utilized to pinpoint the essential features, thereby optimizing the processing time and reducing overall complexity. The results from simulated experiments indicate that this hybrid approach can achieve a maximum recognition rate of 88% across seven facial expressions.

The system is prototyped on a Zynq 7020 SoC platform, which integrates a dual-core ARM Cortex A9 processing system (PS) with FPGA logic (PL), connected through high-throughput communication channels. Based on the profiling results of the initial classification algorithm, a HW/SW Co-Design was developed. This design achieves a real-time classification performance of 27.5 fps, which is a satisfactory outcome. This performance marks a significant improvement, nearly 18 times faster than the original implementation, demonstrating the efficiency and effectiveness of the proposed system.

Acknowledgment

This project was funded by the Deanship of Scientific Research (DSR) at Sultan Qaboos University (SQU), under grant number “IG/ENG/ECED/19/01”. The authors gratefully acknowledge SQU for their financial support.

References

[1] N. Raut, “Facial Emotion Recognition Using Machine Learning”, *Proceedings of ICETIT 2019*, pp 543-558, 2019.

[2] D. Y. Liliana and T. Basaruddin, “Review of Automatic Emotion Recognition Through Facial Expression Analysis”, *Inter. Conf. on Electrical Eng. & Computer Science (ICECOS)*, Pangkal Pinang, pp. 231-236, 2018.

[3] AR. Khan Facial emotion recognition using conventional machine learning and deep learning methods: current achievements, analysis and remaining challenges. *Information,* vol.13, n° 6, pp.268, 2022.

[4] M. Karnati, A. Seal, J. Jaworek-Korjakowska and O. Krejcar, "Facial Expression Recognition in-the-Wild Using Blended Feature Attention Network," *in IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1-16, 2023.

[5] D. Yang, A. Alsadoon, P. Prasad, A. Singh, and A. Elchouemi, “An emotion recognition model based on facial recognition in virtual learning environment”, *Procedia Computer Science,* vol. 125, pp. 2-10, 2018.

[6] N. Abid, T. Ouni, A.C. Ammari, and M. Abid, “Efficient and high-performance pedestrian detection implementation for intelligent vehicles”, in *Multimedia Systems*, vol. 28, n°. 1, pp.69-84, 2022.

[7] W. Zhou, S. Gao, L. Zhang and X. Lou, "Histogram of Oriented Gradients Feature Extraction From Raw Bayer Pattern Images," *in IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 5, pp. 946-950, 2020.

[8] L. Khriji, A.C. Ammari, and M. Awadalla, “Artificial intelligent techniques for palm date varieties classification”, *International Journal of Advanced Computer Science and Applications*, vol. 11, n°.9, pp. 489-495, 2020.

[9] F.E. Chaves, T.P. de Araujo, and J.E.B. Maia, “Facial Expression Recognition: A Cross-Database Evaluation of Features and Classifiers” *Journal of Intelligent Computing* Volume, vol. 10, n°. 1, pp.34, 2019.

[10] N Abid, K Loukil, T Ouni, W Ayedi, AC Ammari, M Abid, “An improvement of multi-scale covariance descriptor for embedded system”, *Journal of Real-Time Image Processing*, vol. 17, pp. 419-435, 2020.

[11] J. K and N. K. Prakash, "Prediction of Brain Stroke using Machine Learning with Relief Algorithm," *2022 International Conference on Edge Computing and Applications (ICECAA)*, Tamilnadu, India, pp. 1255-1260, 2022.

[12] R. Durgabai and Y. R. Bhushan, "Feature selection using ReliefF algorithm", *Int. Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, pp. 8215-8218, 2014.

[13] W. -y. LU and M. YANG, "Face Detection Based on Viola-Jones Algorithm Applying Composite Features," *2019 International Conference on Robots & Intelligent System (ICRIS)*, Haikou, China, pp. 82-85, 2019.

[14] T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *in IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, 2002.

[15] S. Sivaganesan, M.S. Antony, E. Udayakumar, “An Event-Based Neural Network Architecture with Content Addressable Memory. *International Journal of Embedded and Real-Time Communication Systems*”, vol. 11, n°. 1, pp. 23-40, 2020.

[16] J. Rettkowski, A. Boutros, D. Göhringer, “HW/SW Co-Design of the HOG algorithm on a Xilinx Zynq SoC”, *Journal of Parallel and Distributed Computing*, vol. 109, pp. 50-62, 2017.

[17] J. Rettkowski, A. Boutros, D. Göhringer, “Real-time pedestrian detection on a Xilinx Zynq FPGA using the hog algorithm”, in: *Proc. of the International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, Cancun, Mexico, December 2015.

[18] Digilent (2022). Zybo Z7 Board Reference Manual. Retrieved January 25, 2022 from https://reference.digilentinc.com/_media/reference/programmable-logic/zybo-z7/zybo-z7_rm.pdf

[19] N Abid, AC Ammari, A Al Maashri, M Abid, M Awadalla, “Accelerated and optimized covariance descriptor for pedestrian detection in self-driving cars”, *Design Automation for Embedded Systems*, vol. 27, n°. 3, pp. 139-163, 2023.

[20] Vitis Unified Software Platform, https://www.xilinx.com/products/design-tools/vitis.html, January 2024.

[21] Xilinx (2021). Vivado High level synthesis-User Guide UG 902, v2021.1. Retrieved January 25, 2023 from https://docs.xilinx.com/v/u/en-US/ug902-vivado-high-level-synthesis

[22] A Maraoui, S Messaoud, S Bouaafia, AC Ammari, L Khriji, M Machhout, “PYNQ FPGA hardware implementation of LeNet-5-based traffic sign recognition application”, in: *Proc. of the 2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)*, Monastir, Tunisia, pp. 1004-1009, March 2022.