HW/SW Co-Design for Dates Classification on Xilinx Zynq SoC

Ahmed Chiheb Ammari, Lazhar Khriji, Medhat Awadalla
College of Engineering, Sultan Qaboos University, Muscat, Al-Khoud 123, Sultanate of Oman
chiheb@squ.edu.om, lazhar@squ.edu.om, medhatha@squ.edu.om

Abstract—**This paper proposes HW/SW Co-design of an automatic classification system of Khalas, Khunaizi, Fardh, Qash, Naghal, and Maan dates fruit varieties in Oman. The system implements pre-processing, segmentation of the colored input images, color and shape-size features extraction followed by ANN-tansig classification. The performance of the proposed system is experimented and 97.26% highest classification accuracy are achieved. The proposed system is prototyped using a selected Zynq 7020 SoC platform featuring, on the same chip, a dual-core ARM Cortex A9 processing System (PS) interconnected with FPGA logic (PL) though high-throughput communication channels. The original classification algorithm is profiled and then a HW/SW Co-design is developed achieving 10.9 fps real time classification performance. This performance is acceptable and represents almost 14 times speedup acceleration comparatively to the original program implementation.**

I. Introduction

Oman is one of the countries that depends on dates as a source of income. It has more than 250 varieties of dates having different colors, shapes, sizes, and texture [1, 2, 3]. Manual classification of dates into different classes needs meticulous and hard effort. Automating these tasks using a dedicated embedded computer vision system will help classify the multiple varieties of dates fast and accurately. It will also improve the date's production quality of the country [4]. However, automating the classification tasks is complex and challenging because of the irregular sample features within the same type of classes [5, 6] and of the similarities that can be found between samples from different classes [7]. Many previous works attempted to automate dates classification using multiple types of features. In 2012, 15 features are used to automate the classification of seven different categories of dates [4]. In this system, Nearest Neighbor, Linear Discriminant Analysis (LDA), and Artificial Neural Network (ANN) classifiers are tested for comparison purposes. Date classification using ANN studied in [8] achieved a maximum classification accuracy of 91.1%. In 2014, texture, color and shape features are extracted from the dates images [9]. In this work, Fisher discrimination Ratio (FDR) is used for dimensionality reduction of features and Support Vector Machine (SVM) was selected for classification. An automatic date classifier is also developed in [10]. This system uses histogram and texture features and implements LDA and ANN classifiers. In our previous work [11], the effect of color, shape, size, and texture features to classify six different date palm varieties in Oman is studied and the performance of multiple ANN, SVM, and K-Nearest Neighbor (KNN) classifiers are compared. It was demonstrated that (i) the combination of color and shape-size features achieves the highest accuracy and (ii) ANN classifiers have the highest performance.

In this paper, embedded systems prototyping of the highest performant ANN classifier is considered aiming at a minimum of 10 fps real time classification performance. In the literature, various platform alternatives for computer vision prototyping are proposed. Field Programmable Gate Arrays (FPGAs) have the advantages of being flexible and easily reprogrammable. FPGA have strong parallel processing capabilities, low power consumptions and small footprints [12]. Moreover, new FPGA devices are capable to easily connect with CMOS imagers and multiple other external sensors and communication devices. Nevertheless, the use of dedicated FPGA coprocessors to process tasks that exhibit high data-dependency and irregular control flow requires longer development time as compared to programmable processors that are therefore a prime choice for these tasks. To achieve higher degrees of operational capabilities, while maintaining the flexibility in system development, modern All Programmable System on Chip (APSoC) devices [13] are offered with customizable FPGA hardware and ARM Cortex A9 microprocessor cores in the same chip.

In light of the above, a Zynq 7020 SoC platform is selected for the HW/SW Co-design of the highest performant ANN classifier. This platform enables the design of heterogeneous and integrated architectures, in which information are exchanged at silicon speed between the processor and logics on the same chip. In reference to solutions implemented using separate processors with FPGA coprocessors communicating on the same board, Zynq 7020 APSoC provides mechanisms for high-throughput communication between the processor and programmable logic devices performing the best of processor and programmable logic in a single chip.

This paper is organized as follows. The next section reviews the flowchart of the proposed system. Performance characterization of the classifier is then discussed. HW/SW Co-design of the proposed classifier is prototyped and real time performance evaluation of the selected design are then presented. Finally, a summary of the results are given in the conclusion section.

II. Flowchart of the Proposed System

The flowchart of the developed system is shown in Fig.1. Starting from colored images of dates with a single date per

image, preprocessing and segmentation of the colored images are applied. Mathematical morphological operations are then performed to remove incorrect segmented pixels. The segmented images are used to extract selected color and shape-size features. The last stage typically implements an ANN classifier that uses the extracted color and shape-size features to identify the right class of dates being tested. The system performance is tested by applying multiple new data instances.

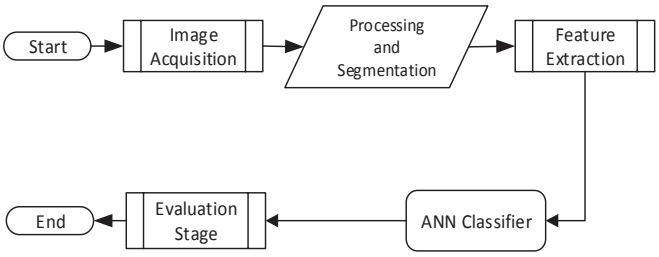

Fig. 1. Flowchart of the proposed system

A. Samples Collection

Khalas, Fardh, Khunaizi, Qash, Naghal and Maan were used in this study since they are the most popular varieties of dates in Oman (see Fig.2). All varieties were obtained from AL-Dhahira Governorate. A total of 600 date samples were selected (100 samples for each class). These samples were imaged individually, one date per image.

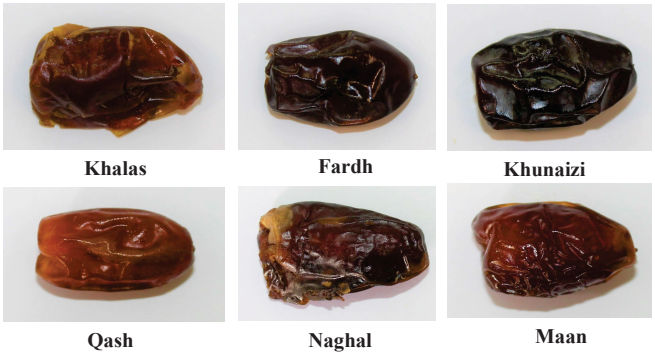

Fig. 2. Samples of the date's dataset used

B. 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. Preprocessing and Segmentation

MATLAB (Version R2014a, the Mathworks Inc., Natick, MA, USA) is selected for algorithm development. The segmentation steps for processing the images are illustrated in Fig.3. To ease and speed up the process, the colored images are resized and then converted into grayscale images. The grayscale images are segmented into foreground and background regions so that, only the region of interest is selected. Otsu's method [14] is implemented followed by morphological operations. Otsu's method uses the image histogram to select a global threshold that separates the image into two classes (background and foreground) [15]. Threshold searching process tries to maximize the variance of the two classes [7]. The segmented images are binary images where the foreground is white and the background is black. Moreover, morphological operations are performed to improve the segmentation process [8]. After segmentation, the vertical and horizontal date coordinates are identified. These coordinates are needed to crop the date images so that only the region of interest is used at the feature extraction phase. The image size is reduced and the classification process is therefore considerably speeded up.

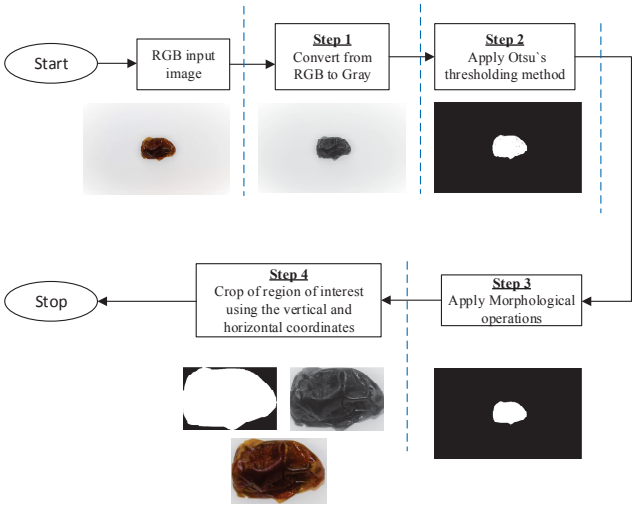

Fig. 3. Segmentation steps applied to the date samples

D. Features Extraction

A major challenge in date's varieties classification is to identify the best features that can effectively extract the appropriate variety characteristics from a date image. Date features are categorized into three groups to distinguish between the different date's varieties. These groups are; (i) colors features, (ii) size-shape features, and (ii) skin texture features. Color and shape-size features are demonstrated in [11] to be most expressive features. The categorical representation of the selected color and shape size features is shown in Fig.4.

1) Color Features

Since dates varieties are different in colors, color features provide powerful information in the field of date's classification. Nine features are extracted from the date`s color. First, the cropped RGB images are converted into three-

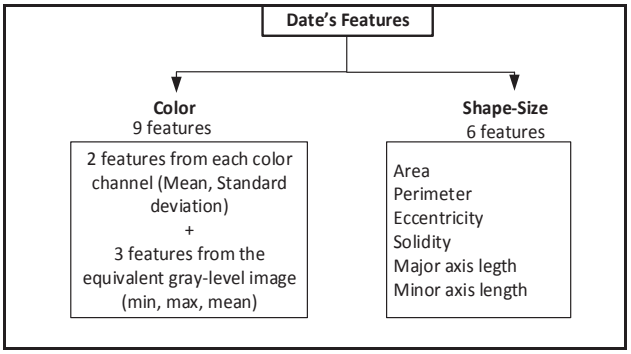

Fig. 4. Selected Color and Shape-Size Date's features.

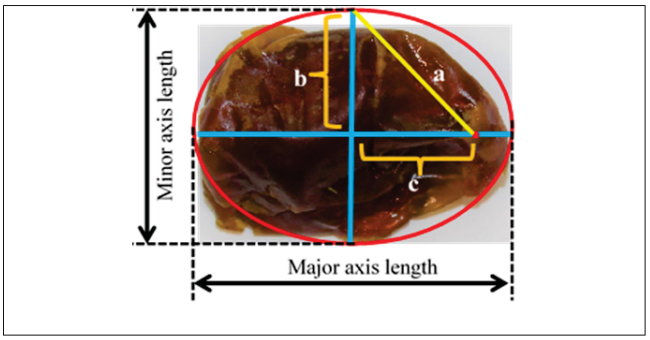

Fig. 5. Major axis, minor axis and eccentricity parameters of the ellipse

color channels, which are red (R) channel, green (G) channel and blue (B) channel. Then, from each channel, the mean and the standard deviation are computed. In addition, gray images are used to compute three more features, which are minimum, maximum and mean intensity. The value of the pixel with the smallest and greatest intensity in the region represents the minimum and maximum intensity respectively while the mean of all intensity values in the region represents the mean intensity [4].

2) Shape and size Features

The sizes and shape are important features for date's classification because different types of dates have different shapes and sizes. These features can improve the accuracy of classification if they are included in the features vector. As illustrated in Fig.5 the shape and size features are calculated from the segmented images in terms of pixels: Area, Major axis length, Minor axis length, Ellipse eccentricity, solidity and perimeter [16]. The area is obtained by counting the number of pixels in the segmented images. Due to the natural shapes of the dates, ellipse is selected as the best modeling shape. Major axis and minor axis lengths are computed by finding the length of the major and minor axes of the ellipse with the same normalized second central moments of the region. In addition, the eccentricity is defined as the ratio between the major axis length and the distance separating the two foci as shown in (1).

Eccentricity (e) is computed as,

$$e = \frac{c}{a} = \frac{\sqrt{a^2 - b^2}}{a} \quad (1)$$

Where c is the distance from the center to the focus of the ellipse and a is the distance from the center to a vertex.

The solidity is calculated by finding the proportion of the pixels in the convex hull that are also in the region. It shows the degree to which shape is concave or convex. It is given by,

$$Solidity = \frac{Area}{Convex\ Area} \quad (2)$$

The perimeter is computed by counting the number of pixels in the boundary of the extracted object [4].

E. Artificial Neural Network classifier

Multilayer neural network is one of the common classifiers used in fruit classifications studies. It is represented as graph of layers of neurons (computing units), such that all the neurons are connected to each other in adjacent layers through weight links [17, 18]. A multilayer neural network has an input data vector and an output data vector. It is characterized by a weight matrix and a transfer function. Each neuron receives input data from neurons in its preceding layer, process these inputs according to the weight links and transfer function to define the outputs for the neurons of the succeeding layers [19]. The network topology plays an important role in determining its performance [20]. In our work, two-layer feed-forward networks trained with Levenberg-Marquardt backpropagation with tansig hidden neurons and softmax output neurons are used.

III. CHARACTERIZATION OF THE CLASSIFICATION PERFERMANCE

The 600 samples dataset are used for performance characterization of the proposed system for dates classification. The dataset is divided into 3 subsets with 68% for training (408 images), 12% for validation (72 images) and 20% for testing (120 images). Aiming at achieving the best performance, the ANN network is trained and tested 30 times using different number of hidden neurons. The average accuracy results vs the number of hidden neurons are obtained in Fig.6. The classifier performance is shown to linearly increase as the number of hidden neurons increases from 1 to 3. Very small accuracy improvements appear when the number of neurons varies from 4 to 10. Up to 97.26% classification performance are achieved with seven hidden neurons and tansig activation functions. Logsig ANN reached the same accuracy with more hidden neurons (9 neurons) compared to tansig neural network.

Moreover, confusion matrix is used to evaluate the accuracy of the proposed classification algorithm for training, validation and testing datasets. The obtained results are shown in fig.7. In this figure, the desired output represents the target class and the output class indicates the system's output. For our case, classes 1, 2, 3, 4, 5, 6 represent Khalas, Fardh, Khunaizi, Qash, Naghal, Maan, respectively. The proposed ANN with tansig network performs very well in classifying

each class perfectly (recall of 100%) except class 5, where only one sample is misclassified as class 3 instead of 5.

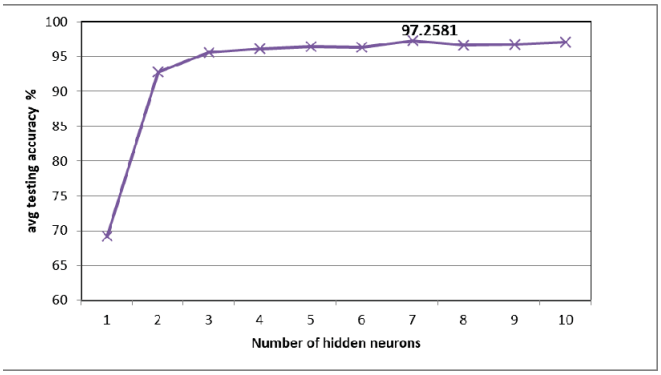

Fig. 6. Classification performance of the ANN classifier vs the number of hidden neurons (tansig-softmax) using 15 features from color and shape

Confusion Matrix

Output class \ Target class	1	2	3	4	5	6	precision
1	21 17.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
2	0 0.0%	21 17.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
3	0 0.0%	0 0.0%	21 17.5%	0 0.0%	1 0.8%	0 0.0%	95.5%
4	0 0.0%	0 0.0%	0 0.0%	22 18.3%	0 0.0%	0 0.0%	100%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	18 15%	0 0.0%	100%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	16 13.3%	100%
recall	100%	100%	100%	100%	94.7%	100%	99.2% 0.8%

Fig. 7. Confusion matrix of ANN classifiers using color and shape-size features

IV. Embedded System Prototyping and Performance Evaluation

In this section we consider embedded system prototyping of the ANN-tansig classifier using a combination of 15 color and shape features. Various implementation platforms are proposed in the literature. CPU platforms are preferred for their flexibility and ease of programmability. The increasing demand for high speed has resulted in the use of Graphics Processing Units (GPUs). The parallel nature of GPUs makes them good candidates for the acceleration of many computer vision algorithms. Nevertheless, GPUs are known for their higher power consumption.

On the other hand, FPGAs are attractive in terms of processing time performance and power consumption [21, 22]. Using, FPGA, the feature extraction and the learning steps of the classification may be accelerated using multiple design techniques [23]. However, on the downside, FPGAs are most often specified directly in low level less expressive hardware description languages such as Verilog or VHDL. Moreover, longer development time is required to process tasks that exhibit high data-dependency and irregular control flow as compared to programmable processors that are a prime choice for these tasks [24]. To maintaining good flexibility in system development while achieving higher degrees of operational capabilities, All Programmable System on Chip (APSoC) devices are offered with customizable FPGA hardware and to-program ARM processor architectures.

For this paper, a ZYBO Z7 development board is selected [25]. This platform features a Zynq 7020 SoC integrating a dual-core ARM Cortex A9 processor, 1GB DDR3 and key peripherals within an Artix-7 FPGA. High-throughput communication mechanisms are provided between the programmable logic (PL) and the processing System (PS) containing the ARM CPUs performing the best of processor and programmable logic in a single chip. Otherwise, dividing the workload between ARM processors and FPGA should be carefully designed to ensure the best of real time performances. This can be very time consuming.

In addition to software development (SW) for ARM processing, dedicated hardware accelerators (HW) are implemented using specific Hardware Description Languages (HDL). Moreover, synchronization and communication between SW and HW has to be properly handled aiming at the best system integration. To simplify the development of such heterogeneous architectures, the SDSoC [26] system level design framework is selected. Using SDSoC, the developer decides specific functions of the C/C++ application to be accelerated, and then a high-level-synthesis tool (HLS) is used to generate accelerators for the FPGA [27]. Multiple synthesis optimizations can be implemented using dedicated HLS pragmas. The system integration of the generated accelerators and the data exchange with the PS are managed automatically by SDSoC. Furthermore, SDSoC supports bare metal projects, Linux and FreeRTOS operating systems. In this work, SDSoC 2018.3 is used to develop a SW/HW Co-Design for the ANN-tansig classifier.

A. Application profiling on ARM

The original MATLAB code of the ANN-tansig classifier using a combination of 15 color and shape features is converted to a C++ program. The C++ program is compatible with SDSoC and includes all modules of the original MATLAB code. The DDR3 is used to store input image data, output coordinates, and feature descriptors. The program starts allocating a contiguous memory for DMA transfers, and then the original classification algorithm is sequentially executed. ARM CPU execution profiling of the program is the first step in the design flow. It is intended to identify compute-intensive bottlenecks that could be migrated to hardware accelerators. Compiling the compute intensive functions for hardware accelerators will help achieve higher system performance. Fig. 8 shows profiling results of the ANN-tansig classification algorithm. Feature extraction is the most expensive computational workload. It consumes 54.3% of the program execution time. The preprocessing and segmentation module consumes 43.2%. 2.5% overhead is spent in the execution of the remaining functions comparatively to preprocessing, segmentation and feature extraction modules.

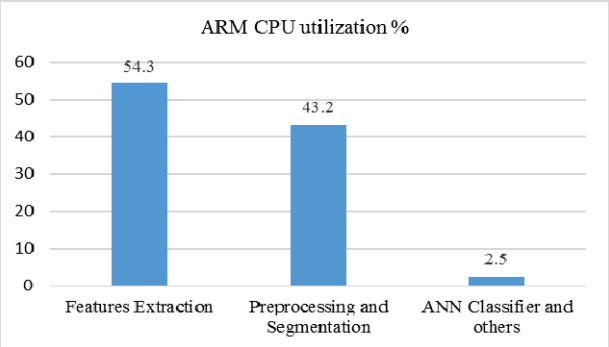

Fig. 8. SDSoC profiling results of ANN-tansig classification program on ZYBO Z7

B. HW/SW Codesign

On the basis of the profiling results, features extraction, preprocessing and segmentation modules are synthesized as dedicated hardware accelerators. Vivado HLS is executed in the background to synthetize these accelerators. Image data are sent in a stream to dedicated row buffers. This would avoid storing the entire image and therefore minimize BRAM memory utilization as much as possible. HLS::stream class in Vivado HLS is used to implement the row buffers. Each function argument interface of the accelerated modules is properly specified using SDSoC Pragmas [26]. To achieve the highest amount of data transfer, we configured the interface as an AXI DMA in scatter gather mode. This AXI DMA controls the transfer of data from the FPGA to the DDR memory and vice versa. Moreover, the data exchange between the ARM core and HW accelerators is synchronized using a data motion network automatically generated by SDSoC. The data motion network can run at different frequencies from the connected processing elements. Fig. 9 shows an overview of the SDSoC developed design. In this design, both HW accelerated features extraction, preprocessing and segmentation modules are configured at 166.67 MHz maximum feasible frequency. Fixed-point arithmetic is used for both software and hardware computations. Moreover, loops inside the HW accelerated modules are unrolled and pipelined with proper HLS pragmas [27] aiming at optimizing as much as possible the high-level synthesis of these blocks.

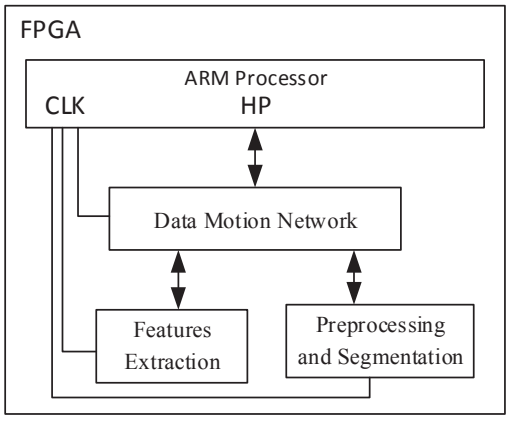

Fig. 9. SDSoC generated design Overview

C. Performance evaluation

Vivado 2018.3 is used to generate the resource utilization of the FPGA. The obtained results are shown in Table I. Processing the original sequential C++ classification program on the ARM processor has achieved 0.78 fps. The SDSoC integration of features extraction, preprocessing and segmentation modules into the FPGA produced 10.9 fps. This represents almost 14 times speedup acceleration comparatively to the original program implementation.

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

Module	LUTs	FFs	DSPs	BRAMs
Features description	3243	4859	4	5
Preprocessing and Segmentation	4254	5783	4	5
Data motion Network	2174	7864	0	18
Total	9671 (19%)	18506 (17%)	8 (3%)	28 (20%)

Fig. 10 presents more details about each module execution time before and after acceleration. The hardware implementation of the features extraction module is 20.44 times faster comparatively to its software implementation. Accelerating the preprocessing and segmentation module procured a speedup of 23.04 times. Using SDSoC, these significant speedup results are obtained at a viable programming overhead. However, the execution of every accelerated module needs significant data transfer expenses with the DDR memory. Particularly, multiple hardware accelerated modules, as implemented for the HW/SW Co-design of this work, require multiple transfers of data from DDR to FPGA and back from FPGA to DDR. This creates an additional performance overhead since for every hardware accelerator, data must be accessed in the DDR memory. This overhead particularly increases for high resolution images and is the main drawback of SDSoC high-level-synthesis framework. In summary, the HW/SW Co-design for this system achieved 10.9 fps real time classification performance. This performance is acceptable. However, it would have been improved if it had been possible to stream the data directly from the preprocessing and segmentation module to the feature extraction module without transferring data by DDR memory.

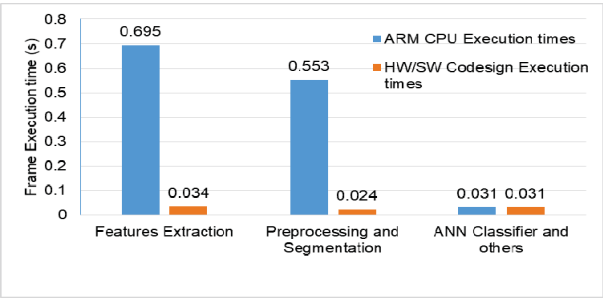

Fig. 10: HW/SW Co-design execution times of each module comparatively to the ARM CPU SW execution times

V. Conclusions

In this paper, a dedicated embedded computer vision system is proposed aiming at automatic classification of Khalas, Khunaizi, Fardh, Qash, Naghal, and Maan dates fruit varieties. The system starts with preprocessing and segmentation of the colored input images. 15 color and shape-size features are extracted and are then fed into an ANN-tansig classifier to identify the appropriate class of dates being tested. The performance of the proposed system at classifying perfectly each class is demonstrated. Confusion matrix results are experimented and 97.26% highest classification accuracy are achieved. HW/SW Co-design of the ANN-tansig classifier using a combination of 15 color and shape features is implemented using a selected Zynq 7020 SoC platform. This platform features, on the same chip, a dual-core ARM Cortex A9 processing System (PS) intercommoned with FPGA logic (PL) though high-throughput communication channels. On the basis of the profiling results of the original classification algorithm, features extraction, preprocessing and segmentation modules are synthesized as dedicated hardware accelerators. The proposed HW/SW Co-design has achieved 10.9 fps real time classification performance. This performance is acceptable and represents almost 14 times speedup acceleration comparatively to the original program implementation.

Acknowledgment

This project was funded partially by OMANTEL under grant number “EG/SQU-OT/18/01”, and partially by Sultan Qaboos University (SQU), Deanship of Scientific Research (DSR), under grant number “IG/ENG/ECED/19/01”. The authors, therefore, acknowledge OMANTEL and SQU for their financial support.

References

[1] Ghnimi S., Umer S., Karim A., & Kamal-Eldin A. (2017). Date fruit (Phoenix dactylifera L.): An underutilized food seeking industrial valorization. *NFS journal,* 6, pp. 1-10.

[2] Bhargava A. & Bansal A. (2018). Fruits and vegetables quality evaluation using computer vision: A review. *Journal of King Saud University-Computer and Information Sciences*.

[3] Janecek A., Gansterer W., Demel M., & Ecker G. (2008). On the relationship between feature selection and classification accuracy," in *New challenges for feature selection in data mining and knowledge discovery*, 90-105.

[4] Haidar A., Dong H., & Mavridis N. (2012). Image-based date fruit classification. In *Proceedings of the IV International Congress on Ultra Modern Telecommunications and Control Systems*. IEEE. St. Petersburg, Russia.

[5] Mizushima A. & Lu R. (2013), An image segmentation method for apple sorting and grading using support vector machine and Otsu’s method. *Computers and Electronics in Agriculture*, 94, pp. 29–37.

[6] Naik S. & Patel B. (2017). Machine Vision based Fruit Classification and Grading-A Review. *International Journal of Computer Applications,* 170, pp. 22-34.

[7] Hameed K., Chai D., & Rassau A. (2018). Comprehensive review of fruit and vegetable classification techniques. *Image and Vision Computing*, 80, 24-44.

[8] Alrajeh K.M. & Alzohairy T.A.A. (2012). Date fruits classification using MLP and RBF neural networks", *International Journal of Computer Applications*, 41(10), 36-41.

[9] Muhammad G. (2014). Automatic Date Fruit Classification by Using Local Texture Descriptors and Shape-Size Features. European *Modelling Symposium (EMS),* IEEE, Pisa, Italy.

[10] Manickavasagan A., Al-Shekaili N.H., Al-Mezeini N.K., Rahman M.S., & Guizani A. (2017). Computer vision technique to classify dates based on hardness. *Journal of Agricultural and Marine Sciences*, 22(1), 36-41.

[11] AL-Abri S., Khriji L., Ammari A. C., Awadalla M. (2019). Classification of Omani’s Dates Varieties Using Artificial Intelligence Techniques. In *Proceedings of the FRUCT’25, FRUCT Oy, Finland*, ISSN 2305-7254, pp. 407-412, Helsinki, Finland.

[12] Bhowmik, E. and Appiah, K. 2018 Embedded Vision Systems: A Review of the Literature, In: Proceeding of 14th International Symposium on Applied Reconfigurable Computing (ARC 2018), Santorini, Greece, May 2018.

[13] Xilinx (2017). Zynq-7000 All Programmable SoC Dara Sheet Overview, Retrieved January 25, 2020 from http://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf

[14] Otsu N., (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on systems, man, and cybernetics*, 9, 62-66.

[15] Gonzalez, R.C., Woods R.E., & Eddins S.L. (2011). *Digital image processing using MATLAB*. New Delhi: Tata McGraw Hill Education Private Limited.

[16] Zdilla M.J., Hatfield S.A., McLean K.A., Cyrus L.M., Laslo J.M., & Lambert H.W. (2016). Circularity, solidity, axes of a best fit ellipse, aspect ratio, and roundness of the foramen ovale: a morphometric analysis with neurosurgical considerations. *The Journal of craniofacial surgery,* 27, 222.

[17] Boumahdi M., El Amrani C., & Denys S. (2019). An Innovative Air Purification Method and Neural Network Algorithm Applied to Urban Streets. *International Journal of Embedded and Real-Time Communication Systems*, 10(4), 1-19.

[18] Sivaganesan S., Antony M.S., & Udayakumar E. (2020). An Event-Based Neural Network Architecture with Content Addressable Memory. *International Journal of Embedded and Real-Time Communication Systems* 11(1), 23-40.

[19] Singh S.K. & Goyal A. (2020). Performance Analysis of Machine Learning Algorithms for Cervical Cancer Detection. *International Journal of Healthcare Information Systems and Informatics*, 15(2), 1-21.

[20] Jain A.K., Duin R.P.W., & Jianchang M. (2000). Statistical pattern recognition: a review. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 4–37.

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

[22] Ammari A. C. (2020). Adaptive battery aware power management of a computer with self power-managed components. *Microprocessors and Microsystems*, 72, 1-12

[23] Chiang C., Chen Y., Ke K., Yuan K. (2015). Real-time pedestrian detection technique for embedded driver assistance systems. In Proc. of ICCE, *IEEE International Conference on Consumer Electronics*, 2015, pp. 206–207, 9-12 January 2015.

[24] Rettkowski J., Boutros A., Göhringer D. (2015) 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.

[25] Digilent (2018). *Zybo Z7 Board Reference Manual.* Retrieved January 25, 2020 from Web: https://reference.digilentinc.com/_media/reference/programmable-logic/zybo-z7/zybo-z7_rm.pdf

[26] Xilinx (2018-a). *SDSoC Environment – User Guide UG1027,* v2018.3. Retrieved January 25, 2020 from Web*:* https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_3/ug1027-sdsoc-user-guide.pdf.

[27] Xilinx (2018-b). *Vivado High level synthesis-User Guide UG 902, v2018.3.* Retrieved January 25, 2020 from Web: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_3/ug902-vivado-high-level-synthesis.pdf