Support Vector Machine Hardware Accelerator for Tongue Color Diagnosis

Amanda Thiah Su Lin, Mohd Shahrizal Rusli,
Nur Diyana Kamarudin, Ab Al-Hadi Ab Rahman,
Usman Ullah Sheikh, Michael Tan Loong Peng,
Shahidatul Sadiah and Mohd Ibrahim Shapiai

October 20, 2021

Support Vector Machine Hardware Accelerator for Tongue Color Diagnosis

Amanda Thiah Su Lin[1*], Mohd Shahrizal Rusli[1*], Nur Diyana Kamarudin[2], Ab Al-Hadi Ab Rahman[1], Usman Ullah Sheikh[1], Michael Tan Loong Peng[1], Shahidatul Sadiah[1], Md. Ibrahim Shapiai[3]

[1]*School of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor, email: *amandasulin@gmail.com, *shahrizalr@utm.my*

[2]*Department of System Safety and Communication, National Defence University of Malaysia, Kem Sg Besi, 57000 Kuala Lumpur.*

[3]*Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Jalan Sulatan Yahya Petra, 54100 Kuala Lumpur.*

Abstract. Tongue body features such as color is used in Traditional Chinese Medicine (TCM) practices to diagnose a patient's state of health. However, the diagnosis of one patient's health condition varies between practitioners. This gives rise to the need of a standard method such as using Support Vector Machine (SVM) to identify the tongue body color. Typically, SVM classifiers are implemented in software where the classification performance is very dependent on the architecture of the general-purpose CPU. Since classification of tongue images is a recurring event, the design of a hardware accelerator is proposed in this project. The purpose of designing a hardware accelerator is to boost the classifier performance, execution time and latency so that it meets real-time constraints. Architectural optimization methods, such as loop unrolling, memory array partitioning and pipelining of the SVM classification algorithm are utilized and the final hardware architecture is synthesized to Xilinx Virtex-7 FPGA. To further optimize the resource utilization, 18-bits IEEE-754 floating-point representations for the floating-point units are used. The proposed SVM hardware demonstrates rougly 140x speed up with only 1% of classification accuracy loss when compared to the software implementation in MATLAB.

1. Introduction

Traditional Chinese Medicine (TMC) uses the color of the tongue body to diagnose the patients' health condition [1]. The tongue body colour can be categorised into two main groups, pink and red. The colour range of the tongue body is rather small as they contain many overlapping and similar pixels [2]. This makes it hard for medical practitioners to distinguish between the different colour groups. Support Vector Machine (SVM) can be used in tongue colour diagnosis due to the algorithm's efficiency and high accuracy in image classification systems [3]. SVM is essentially a supervised machine learning algorithm to perform a binary classification of a given data based on the hyperplane separation derived from the training data. The hyperplane separation function is typically a mathematical function. Each image data may involve multiple hyperplane, so a single image that needs to be classified may require multiple exeution of these functions. This constitutes the main processing delay or latency of the SVM system.

Software implementation of the SVM algorithm can perform classification with high precision, but they cannot efficiently meet real time embedded systems' constraints, such as resource utilization and computation time. This is because the computation of the SVM classification algorithm is limited by the instruction set and the architecture of the CPU used [4]. Therefore, the design of a dedicated hardware for the SVM classification algorithm is proposed in this paper.

The organization of the remaining part of the paper is as follow: Related works on the hardware implementation of the SVM classification algorithm is given in section 2, followed by background on the SVM algorithm for tongue color diagnosis in section 3. The proposed hardware accelerator architecture is given in section 4, experimental results are discussed in section 5, and section 6 concludes the paper.

2. Related Works

Among the most commonly and widely used architecture for the hardware implementation of the SVM classification phase is systolic array architecture. Systolic array architecture enables efficient memory management and data transfer mechanism with a minimum level of complexity. This method was used by [5] for the implementation of a multiclass SVM classifier, targeted for facial expression recognition. On top of that, partial reconfiguration technique was also used, and it was reported that this work was able to achieve a total power reduction of 3-5%. [6] implemented the SVM classification using systolic array architecture as well. In addition to the previous work, the author improved the kernel computation using 2- pipelined stages. 85× speed up in execution time was achieved when compared to an equivalent general purpose pure software implementation on Matlab [7]. Hussain et al. extended their work by exploiting the dynamic partially reconfiguration of the FPGA device to introduce flexibility into the system. Besides systolic array architecture and dynamic partial reconfiguration, there are other implementations that can be implemented to achieve reduced power consumption while speeding up the classification phase computations, for example in digital imaging [8].

The authors in [9] designed the hardware acceleration for SVM classification using a simplified multiplier-kernel where shift and add operations were the substitutes. Canonic Signed Digit (CSD) and Common Subexpression Elimination (CSE) vector data representations were also used in the authors' research work. This is to reduce the number of required adders that are needed for the computation process, thus leading to a lighter, less complicated design [10] that can be implemented on ASIC [11]. The multiplier-less design yielded up to an additional 4% power reductions than that of conventional vector product kernel. Research from [12] stated that the kernel of the SVM classification was realized using CORDIC algorithm. Similar to the previous literature, this multiplier-less design can achieve comparable classification performance with the software implementation. Authors from [13] introduced an alternative hardware design of the SVM classification targeted for colorectal endoscopic images. The authors used fixed-point numbers paired with a pipeline design. They implemented a two-stage pipeline similar to [14], where the inner products of the vectors were computed in the first stage and the summation in the next. According to the results reported, the design in this research demonstrates decreased hardware size with the goodness of increased throughput. The system was able to achieve 21.2 fps at 100 MHz.

Fixed point operations were also used in the work of [15] on SVM classifications to study the effects of using fixed point on the precision of the classification process. Liu et al proposed pipelined adders for the hardware implementation of the SVM classifier. By pipelining the adders, the critical path delay is reduced, thereby speeding up the entire classification computation [16]. In another paper proposed by Saurav et al, the acceleration in the execution speed of the SVM classifier used for facial recognition was achieved with the utilization of the concepts of parallelism and pipelining. To cap it off, the design also adopted a fixed-point data format. In general, pipelining can definitely help to increase the throughput, but it comes with the expense of slight area overhead. Afifi et al. proposed a hardware software co-design architecture for linear kernel SVM for melanoma image classification. The loops in the accumulation of multiplication process of the vectors were unrolled [17], pushing the research work to achieve an acceleration factor of 110× when compared to pure software implementation. The throughput in this work has been improved, but then the resource utilization used for loop unrolling is relatively higher than using only pipelining.

Based on the research works discussed in this section, hardware friendly multiplier-less kernel approach is popular among researchers. The simplification of the hardware using only shift and add operations has contributed to significant reductions in hardware utilization and lower power consumptions. Pipelining technique on the other hand was exploited in many hardware designs to increase the throughput of the classification computation. However, parallel processing translates to more resources like multipliers. This can be overcome by performing scheduling or sharing of the resources within pipeline stages so to decrease the resource utilization. However, this would come with a cost of longer latency.

Besides SVM, tongue color classification for diagnosis can also be performed using convolutional neural network (CNN) [18, 19]. These works have shown that CNN can provide very good accuracy, but at the expense of a significantly higher computational complexity than SVM.

For SVM method, it can be concluded that the main trade off that exists in hardware implementation of the SVM classification phase is between the classification accuracy and the cost, complexity, resource utilization, power consumption and speed of computations. The balance point of these parameters in design the hardware accelerators for SVM classifications has to be determined so to meet the embedded system requirements for a specific application.

3. SVM for Tongue Color Diagnosis

SVM is a supervised machine learning algorithm that aims to explore the characteristics and correlations between the gathered data to deduce the most efficient way in classifying the data into groups [17]. Linear SVM addresses the binary classification problem by finding the linear hyperplane that gives maximum margin when the separating the training samples into two classes. The hyperplane and margin that forms the class separation is shown in Figure 1.

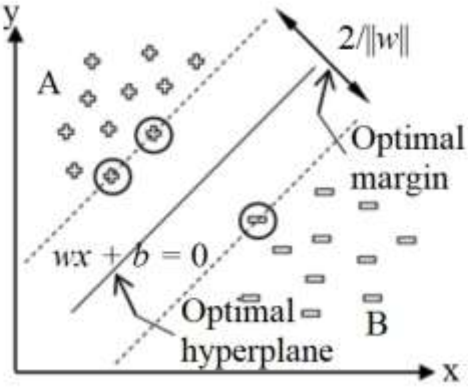


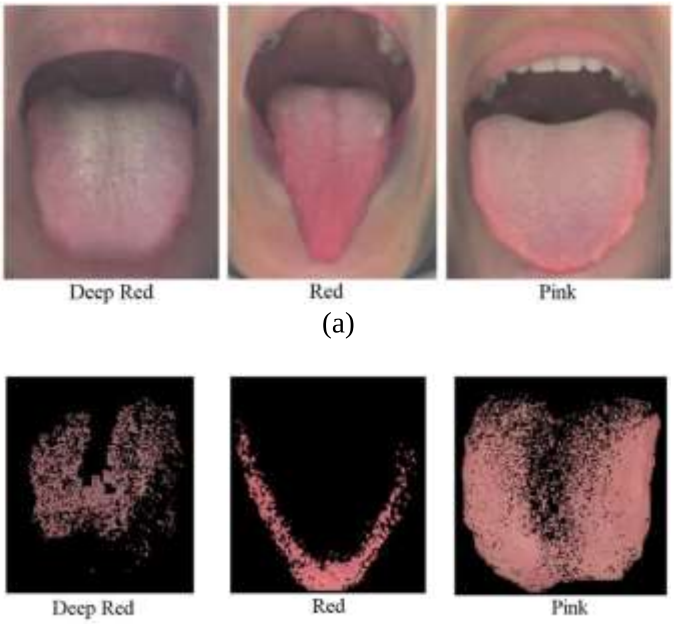
Figure 1. Linear SVM optimal hyperplane separation with maximum margin.

In the training phase, the Lagrange optimization theory and Wolfe’s dual form transformation are adopted to reach the optimal situation where the hyperplane can separate the two classes by a maximum distance. When the parameters and support vectors of the classifier model are well established, it is used in classification phase. The classifier will determine the class, ***g(X)*** of the input vector based on the output from the decision function defined by

$$g(X) = \sum_{n=1}^{N} a_n Y_n (X_n \cdot X) + b \tag{1}$$

where X_n denotes the support vectors while $a_n y_n$ the alpha values. b is the bias value that represents the offset of the hyperplane.

The detailed software-based implementation of the SVM algorithm for tongue color diagnosis used in this paper is given in [1]. The Matlab version of the algorithm has been re-implemented in our work using the same dataset without image pre-processing. Example of the dataset used for the SVM is shown in Figure 2, where backgrond and tongue coating have been removed to improve its performance.



(a)

(b)

Figure 2. Tongue images (a) before background and tongue coating removal, and (b) after background and tongue coating removal.

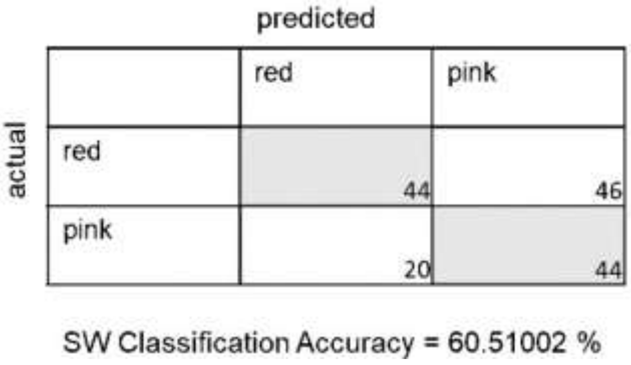
The statistics on image dataset count is given in Table 1. The training dataset tongue images consist of 39 tongue images, and is used as the input to the SVM training phase. The remaining tongue dataset with moderate tongue coating, consisting of 154 tongue images, is used as input to the classification phase. Table 2 shows the coefficients of the linear kernel SVM obtained from training phase. The coefficients were written into text files as they are to be loaded into the memory in simulation. Figure 3 shows the classification accuracy of the classifier. Out of 90 red tongue images, 44 were correctly classified. Out of 64 pink tongue images, 44 were correctly classified. In average, the accuracy of the classifier is 60.51%. The accuracy being measured here is relatively low since additional image pre-processing and augmentation are not implemented since the focus of the work is exclusively on analyzing the performance of the SVM classifier.

Table 1. Tongue image dataset count

Dataset	**Red**	**Pink**	**Total**
Training (original)	10	29	39
Training (modearate coating)	90	64	154
Total	100	93	193

Table 2. SVM parameters and coefficients from the training phase

Kernel Function: Linear	
Coefficient	**Size**
Support vectors, sv	24 x 10000
Alpha, α	24 x 1
Bias, б	0.8269



		predicted	
		red	pink
actual	red	44	46
	pink	20	44

SW Classification Accuracy = 60.51002 %

Figure 3. Software implementation classification accuracy based on the confusion matrix.

4. Proposed Hardware Accelerator Architecture

Block level description of the proposed architecture of the linear SVM classifier is illustrated in Figure 4. The design consists of a controller to provide control signals for memory content initialization, memory read address counter, register content clearing and classification computation initialization. The design also contains the memory blocks where the test data, support vectors, alphas and bias values are stored. Then in the classification computation block, it can be divided into two main sections. The first section performs the inner product computation between the input image pixels and the support vectors. The second section performs dot product between the previously computed results and the stored alpha values. In this segment of the computation unit, the floating-point values were represented in 18-bit IEEE 754 format.

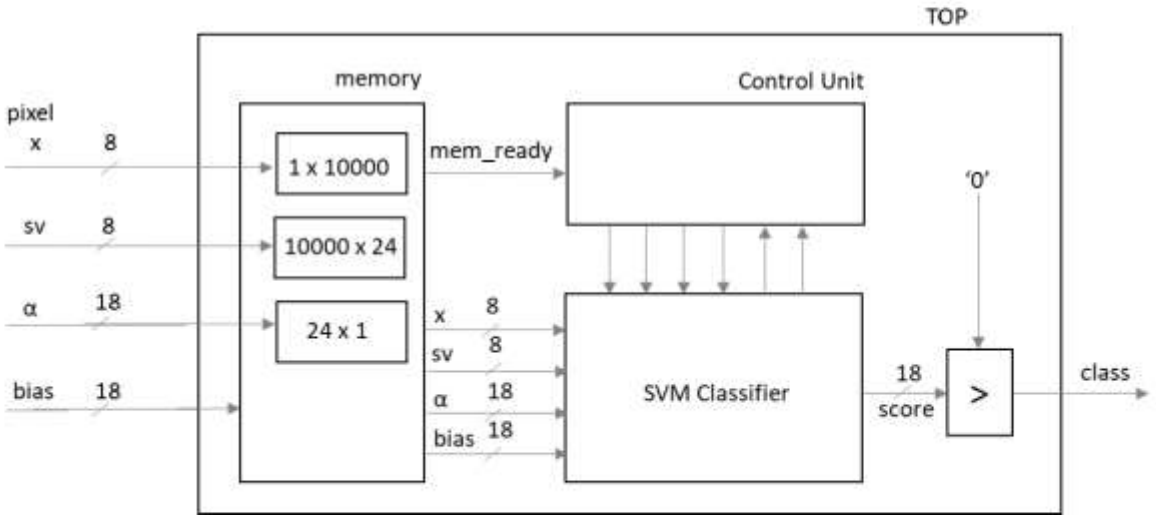


Figure 4. Proposed top-level block diagram of the SVM hardware architecture

The details for each of the block: memory, control unit, and the SVM classifier are explained as follows.

4.1 Memory Architecture

The memory block holding the input image value is called "x". It has a depth of 10000 and a width of 8 bits, with only one memory read port. Support vector, SV memory block contains 24 individual smaller memory blocks (this project has 24 support vectors) with depth of 10000 and width of 8 bits. Each of the individual blocks carry one complete support vector. As a whole, the support vector memory block has 24 memory read ports. Alpha memory block, α has a depth of 24 and width of 18 bits. The memory block has only one read port. Figure 5 illustrates the memory block sizes and the number of memory-read ports

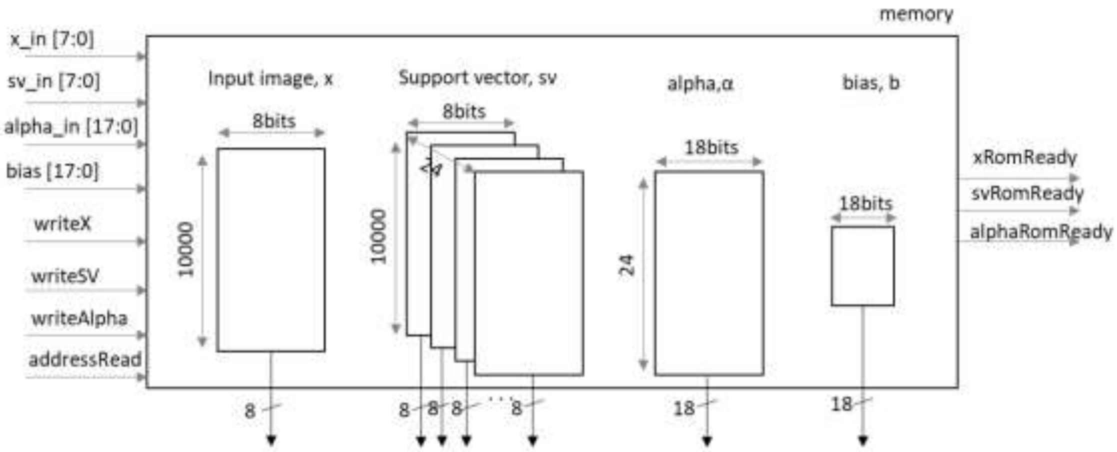


Figure 5. Memory architecture block diagram

4.2 Control Unit

In order to obtain a proper sequence of operations through precise coordination between different modules, a controller has been developed as part of the proposed hardware architecture. The control unit consists of a finite state machine that generates control signals to clear register contents, load memory block, read from memory blocks and enabling accumulators at proper instant of time. The control unit design for the SVM hardware implementation consists of five states: S0, S1, S2, S3, S4. The description of the states is listed in Table 3.

Table 3. Proposed control unit states and description

State	Description
S0	1. Clear register content upon reset 2. initialize memory address to zero 3. clear memory ready flags.
S1	1. Assert write memory signals to input memory with image, sv, alpha and beta 2. If romReady, stop writing to memory 3. Read memory values for calculation 4. Start loop1 calculation
S2	1. Increment memory read address
S3	1. Assert the enable signal to start loop2 calculation

S4	1. Clear content in registers if there is a next input image 2. Assert write memory signal to start loading the next input image

4.3 SVM Classifier

Figure 6 shows the architecture of the classification computation unit. Generally, the hardware optimization techniques used in the development of this hardware architecture are, fully unrolling the loop performing the inner product between the input image pixel and support vectors and pipelining the loop that performs the dot product between the resultant matrix from the inner product and the alpha values. Loop1 in Figure 6 performs the inner product computation while loop2 performs the dot product computation. To accommodate for fully unrolling loop1, the array "sv" was completely partitioned with respect to the second dimension, the original matrix (10000×24) was broken down into 24 smaller memory blocks with depth of 10000 with corresponding width of 8 bits. The input tongue image, x is loaded into one memory block with depth of 10000 and corresponding bit width of 8.

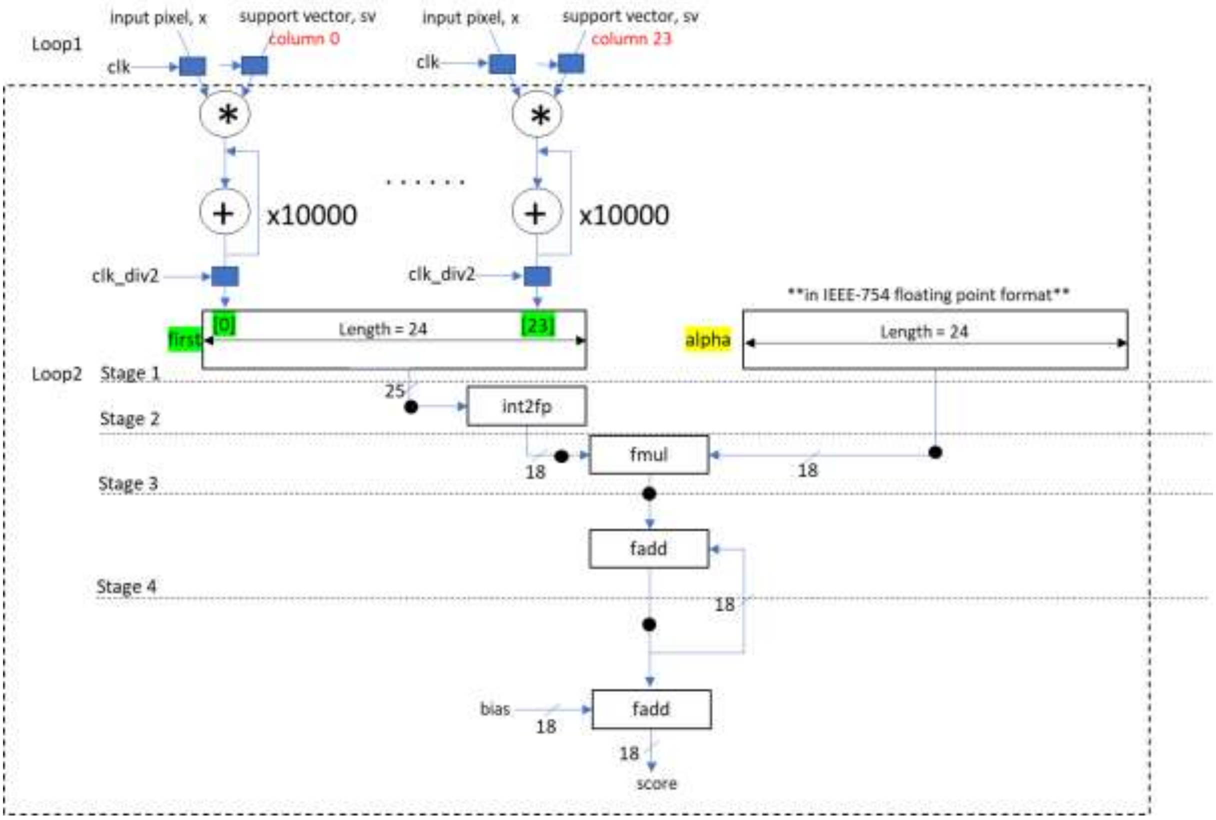


Figure 6. Proposed SVM classifier computation unit architecture block diagram

5. Experimental Results

The proposed top-level SVM classifier system in Figure 4 has been designed in System Verilog and synthesized to Xilinx Virtex-7 FPGA for performance analysis and benchmark. To demonstrate the effectiveness of our method, we have compared with the original software Matlab implementation in [20]. The result is summarized in Table 4.

The proposed architecture is equipped with 18 bits floating point computations that achieved almost similar accuracy with one misclassification across 153 tongue images, showing only ~1% of accuracy degradation. The maximum achievable frequency of the proposed hardware is 128 MHz while for software, a PC with Intel Core i5-8350 @ 1.7 GHz CPU and 8 GB of RAM was used. The execution time to classify in both platforms were measured, and results show that hardware implementation requires only 0.157 ms, while 22 ms was required for software for a complete SVM classification. This translates to roughly 140x faster execution time on our proposed implementation and platform as compared to the reference software implementation.

Table 4. Comparison between the proposed hardware and existing software implementation of the SVM classifier system.

	System Verilog (Proposed)		Matlab [20]	
	Class1 (red)	Class2 (pink)	Class1(red)	Class2(pink)
Class 1 (red)	44	46	44	46
Class2 (pink)	21	43	20	44
SVM classification accuracy (%)	59.69		60.51	
Number of misclassification	1		0	
Platform	Xilinx Virtex-7 xc7z020clg484-1		Intel Core i5-8350 @ 1.7 GHz	
Maximum Frequency (MHz)	128		1700	
Total execution time (ms)	0.157		22	

6. Conclusion

In this paper, a hardware implementation of the linear SVM classification algorithm for tongue color diagnosis has been presented. The designed architecture has been coded in System Verilog and simulations of the hardware were performed in Vivado Design Suite targeting FPGA implementations, more specifically, xc7z020clg484-1 from Xilinx Virtex-7 family. The implemented architecture can perform tongue color classification at a clock frequency of 128 MHz with a complete execution time of only 0.157 ms. The proposed architecture outperforms the software implementation of the classification algorithm in terms of overall execution time by rougly 140x. The reduction of the floating-point representation from 32 bits to 18 bits also demonstrated a significant improvement in resource utilization while only reducing the classification accuracy by only 1%. Therefore, the proposed architecture is suitable for real-time tongue color diagnosis on a FPGA based embedded platform. Future work can be done to implement a real-time tongue color diagnosis system on FPGA.

7. References

[1] Qi, Z., Tu, L.-P., Chen, J.-B., Hu, X.-J., Xu, J.-T. and Zhang, Z.-F. The Classification of Tongue Colors with Standardized Acquisition and ICC Profile Correction in Traditional Chinese Medicine. *BioMed Research International*, 2016.

[2] Kawanabe, T., Kamarudin, N. D., Ooi, C. Y., Kobayashi, F., Mi, X., Sekine, M., Wakasugi, A., Odaguchi, H. and Hanawa, T. Quantification of tongue colour using machine learning in Kampo medicine. *European Journal of Integrative Medicine*, vol 8, no. 6, 2016.

[3] Afifi, S., Gholamhosseini, H. and Sinha, R. Hardware Implementations of SVM on FPGA: AState-of-the-Art Review of Current Practice. *International Journal of Innovative Science, Engineering Technology (IJISET)*, pp.733–752, 2015.

[4] Wang, J.-F., Peng, J.-S., Wang, J.-C., Lin, P.-C. and Kuan, T.-W. Hardware/software co-design for fast-trainable speaker identification system based on SMO. *2011 IEEE International Conference on Systems, Man and Cybernetics*, 2011.

[5] Patil, R. A., Gupta, G., Sahula, V. and Mandal, A. Power Aware Hardware Prototyping of Multiclass SVM Classifier Through Reconfiguration. *2012 25th International Conference on VLSI Design*, 2012.

[6] Hussain, H., Benkrid, K. and Seker, H. Novel dynamic partial reconfiguration implementations of the support vector machine classifier on FPGA. *Turkish Journal of Electrical Engineering and Computer Sciences*, 2014.

[7] Hussain, H. M., Benkrid, K. and Seker, H. Reconfiguration-based implementation of SVM classifier on FPGA for Classifying Microarray data. *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2013.

[8] Y. Y. Yeap, U. U. Sheikh, A. A. H. Ab Rahman, Image forensic for digital image copy move forgery detection, *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)*, pp.239-244, 2018.

[9] Mandal, B., Sarma, M. P. and Sarma, K. K. Design of a systolic array based multiplierless support vector machine classifier. *2014 International Conference on Signal Processing and Integrated Networks (SPIN)*, 2014.

[10] Mandal, B., Sarma, M., Sarma, K. and Mastorakis, N. Implementation of systolic array based SVM classifier using multiplierless kernel. *Proceedings of the 16th International Conference on Automatic Control, Modelling Simulation (ACMOS '14)*, pp.288-294, 2014.

[11] A. A. H. Ab Rahman, I. Kamisian, A. Z. Sha'ameri, ASIC Modeling and Simulation of Adaptive FIR Filter, *2008 Interantional Conference on Electronic Design,* pp.1-4, 2008.

[12] Vranjkovic, V. and Struharik, R. New architecture for SVM classifier and its application to telecommunication problems. *2011 19thTelecommunications Forum (TELFOR) Proceedings of Papers*, 2011.

[13] Koide, T., Hoang, A.-T., Okamoto, T., Shigemi, S., Mishima, T., Tamaki, T.,Raytchev, B., Kaneda, K., Kominami, Y., Miyaki, R., Matsuo, T., Yoshida, S. and Tanaka, S. FPGA implementation of type identifier for colorectal endoscopie images with NBI magnification. *IEEE Asia-Pacific Conference on Circuits and Systems, Proceedings, APCCAS*, 2015.

[14] N. M. Zabidi, A. A. H. Ab Rahman, VLSI Design of a fast pipelined 8x8 discrete cosine transform, *International Journal of Electrical and Computer Engineering*, vol 7, no. 3, 2017.

[15] Lesser, B., MÃŒcke, M. and Gansterer, W. Effects of Reduced Precision on Floating-Point SVM Classification Accuracy. Procedia Computer Science, 2011.

[16] Afifi, S., Gholamhosseini, H. and Sinha, R. Hardware Acceleration of SVMBased Classifier for Melanoma Images. *Lecture Notes in Computer Science*, 2016.

[17] Liu, C., Qiao, F., Yang, X. and Yang, H. Hardware acceleration with pipelined adder for Support Vector Machine classifier. *Fourth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*, 2014.

[18] Guangyu N., Caiqun W., Bo Y., Yong P., Tongue Color Classification Based on Convolutional Neural Network. *Advances in Intelligent Systems and Computing*, vol 1364. Pp 649-662, 2021.

[19] Hou, J., Su, H., Yan, B., Zheng, H., Sun, Z., & Cai, X., Classification of tongue color based on CNN. *2017 IEEE 2nd International Conference on Big Data Analysis*, pp 725-729, 2017.

[20] Kamarudin, N. D., Ooi, C. Y., Kawanabe, T., Odaguchi, H. and Kobayashi, F. A Fast SVM-Based Tongues Colour Classification Aided by k-Means Clustering Identifiers and Colour Attributes as Computer-Assisted Tool for Tongue Diagnosis. *Journal of Healthcare Engineering*. vol. 2017. pp.1–13, 2017.