

Hardware/Software Co-design for a Gender Recognition Embedded System

Andrew Tzer-Yeu Chen¹, Morteza Biglari-Abhari¹, Kevin I-Kai Wang¹,
Abdesselam Bouzerdoum², and Fok Hing Chi Tivive²

¹ Department of Electrical and Computer Engineering, University of Auckland,
Auckland, New Zealand

{andrew.chen, m.abhari, kevin.wang}@auckland.ac.nz

² School of Electrical, Computer, and Telecommunications Engineering, University of
Wollongong, Wollongong, Australia

{bouzer, tivive}@uow.edu.au

Abstract. Gender recognition has applications in human-computer interaction, biometric authentication, and targeted marketing. This paper presents an implementation of an algorithm for binary male/female gender recognition from face images based on a shunting inhibitory convolutional neural network, which has a reported accuracy on the FERET database of 97.2%. The proposed hardware/software co-design approach using an ARM processor and FPGA can be used as an embedded system for a targeted marketing application to allow real-time processing. A threefold speedup is achieved in the presented approach compared to a software implementation on the ARM processor alone.

Keywords: real-time · embedded system · computer vision · FPGA · neural network · co-design · hardware acceleration

1 Introduction

Gender recognition has important applications for developing computer systems that are better able to identify and interact with humans, from biometric authentication to targeted marketing and advertising. However, this is a non-trivial task as there are many variations in facial features to recognise, as well as other environmental conditions that can make accurate characterisation difficult and increase the computational complexity. Even for humans, accurate gender recognition can be challenging as elements of physical appearance derived from genetic makeup, such as bone structure, may not be accurate indicators of gender or gender preference, and can lead to erroneous identification.

A targeted marketing application is our focus, where we aim to determine certain demographic characteristics of the individuals at an intersection where they can see a digital billboard, so that the digital billboard can show the most appropriate advertisement for the audience. In this application, a real-time embedded system implementation is required; images must be processed in real-time so that the right advertisement is



Libraries and Learning Services

University of Auckland Research Repository, ResearchSpace

Version

This is the Accepted Manuscript version. This version is defined in the NISO recommended practice RP-8-2008 <http://www.niso.org/publications/rp/>

Suggested Reference

Chen, A. T., Biglari-Abhari, M., Wang, K. I. K., Bouzerdoun, A., & Tivive, F. H. C. (2016). Hardware-Software Co-design for a Gender Recognition Embedded System. In H. Fujita, M. Ali, A. Selamat, A. Sasaki, & M. Kurematsu (Eds.), *Lecture Notes in Computer Science: Trends in Applied Knowledge-Based Systems and Data Science* Vol. 9799 (pp. 541-552). Morioka, Japan: Springer. doi: [10.1007/978-3-319-42007-3_47](https://doi.org/10.1007/978-3-319-42007-3_47)

Copyright

Items in ResearchSpace are protected by copyright, with all rights reserved, unless otherwise indicated. Previously published items are made available in accordance with the copyright policy of the publisher.

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-42007-3_47

For more information, see [General copyright](#), [Publisher copyright](#), [SHERPA/RoMEO](#).

shown to the current audience, but because it is an embedded implementation, power consumption and system cost should also be minimised.

The detection of human faces through computer vision is well established. However, we often need more information about the audience; gender recognition is one step towards more intelligent applications. There are a number of gender recognition algorithms in the literature. One of them is an algorithm developed by Tivive and Bouzerdoum [1], which uses a shunting inhibitory convolution neural network [2] to identify faces in images and classify them as male or female. The neuron model used in the neural network is based on excitatory and inhibitory weights, which is a biologically plausible explanation for how the brain processes visual images [3]. The network is also structured to allow for a certain degree of shift and distortion invariance; critical for processing real-world images that may not be as controlled as researchers would like them to be. This algorithm achieved a reported accuracy on the Facial Recognition Technology (FERET) [13] database of 97.2%.

This paper focuses on the gender recognition part of the algorithm presented in [1], which was developed and trained in MATLAB, and investigates the feasibility and performance of implementation in an embedded system. Therefore, the first step is to implement the algorithm in C. To satisfy the real-time constraints on an embedded platform and reduce the design time, a HW/SW co-design approach has been adopted in our implementation. A Terasic DE1-SoC development board featuring an Altera Cyclone V FPGA which includes a dual-core ARM Cortex-A9 processor is used as our implementation platform.

The remainder of the paper is organised as follows; Section II discusses the motivating application in more depth and Section III investigates some of the related work in this area. Section IV describes the steps taken to implement the algorithm and the portions of the algorithm targeted for hardware acceleration on the FPGA fabric, Section V presents the experimental results, and Section VI discusses areas for future work.

2 Motivating Application

At a busy intersection, a digital billboard is mounted on one of the buildings so that it is visible to approaching vehicles. The advertisement on the billboard changes once every ten seconds; however, at this stage, the billboard cycles through a predefined set of advertisements. Decades of research have shown that there are differences in consumer behaviour and preferences between the two main gender types, male and female [4]. Marketers are therefore interested in targeting their advertisements towards specific genders; showing the wrong advertisement is a waste of time and money, and results in an inefficient advertising spend.

A camera could be placed above the billboard, scanning the faces of front-seat occupants of approaching vehicles and pedestrians. Using a real-time gender recognition algorithm, the percentage of males and females (the gender distribution) currently looking at the billboard can be determined. An appropriate advertisement can be selected that better targets that particular audience, turning the passive billboard into a more

active advertising medium. This notion of a “smart billboard” is an example of intelligent systems; adding computing capabilities to an otherwise static system. Additionally, by counting the number of faces, client companies could be billed more accurately for the number of actual impressions made.

In order to achieve this, the gender distribution of the audience must be identified in real-time. This is not difficult for a single face, but more challenging when there are many faces, changing at high speed as vehicles move through the intersection. As the faces are moving, there may also be blurring effects that make some images unusable, so gender recognition may need to be performed on the same face multiple times in different positions and orientations in order to achieve accurate identification.

During peak times, assuming the vehicles are not currently stopped at a traffic light, a busy intersection could have as many as a hundred unique individuals travelling through the intersection, or in marketing terms, a hundred impressions, every ten seconds. This means that, assuming that the extraction of faces is dealt with by another processor, a gender recognition embedded system in this application should achieve at least ten successful identifications per second in order to provide an accurate gender distribution to inform advertisement selection.

3 Related Works

Ng et al. [5] presented a comprehensive survey of vision-based human gender recognition, which shows a large amount of activity in this area. They report that a human can achieve roughly 95% accuracy in male/female binary gender recognition. In computer vision, 99.8% accuracy has been achieved in controlled environments [6], while in uncontrolled environments the accuracy is up to 95% [7]. Ng et al. also describe potential applications, in particular demographic classifiers for customer relationship and marketing systems, which require the ability to process 15-20 images per second.

In the embedded context, there have been a few implementations of gender recognition algorithms. Perhaps the most significant is Azarmehr et al. [8], which uses a Support Vector Machine (SVM) and Radial Basis Function (RBF) Classifier on a 1.7GHz quad-core Snapdragon 600 SoC to characterise gender in 2.3ms per image with 95% accuracy. Their algorithm also detects faces and characterises age, with an average performance of 15 to 20 frames per second.

Irick et al. briefly report in [9] an SVM algorithm implemented purely on an FPGA, achieving only 88% accuracy but processing a massive 1,100 images per second at 100MHz. Irick et al. also reported in a separate paper [10] an artificial neural network (ANN) based system implemented on an FPGA that achieves an accuracy of 83.3%, processing roughly 30 images per second. Ratnakar and More [11] report an FPGA-based system that achieves 78% accuracy with a “propagation delay” of 1.9 seconds.

However, there are few implementations in the existing literature bringing these two paradigms together – utilising a hard processor core to better implement floating point arithmetic and maintain precision and accuracy while leveraging the FPGA fabric to improve throughput in order to meet real-time requirements. An important example is

Gudis et al. [12], but in general there are gaps in the literature. There are also few implementations of gender classification using ANNs in an embedded context. While many gender recognition algorithms use SVMs for higher accuracy and modelling flexibility, the ANN can characterise multiple outputs based on multiple input factors or features, and is more suited for fixed hardware implementations that seek to avoid unused capacity or reliance on dynamic reconfiguration. To our knowledge, this paper describes the first implementation of an ANN-based gender recognition algorithm in an embedded system using both a hard core processor and FPGA fabric.

4 Algorithm Implementation

To improve the real-time performance of the software implementation of the gender recognition algorithm, one option is a hardware-only implementation using a hardware description language. However, this requires a large amount of development time and may use a lot of hardware resources for certain operations, such as floating point calculations. Considering the availability of FPGA chips which have hard core processors as well as configurable FPGA resources, HW/SW co-design can be a better approach.

The DE1-SoC development board is used as the target platform with a Cyclone V 5CSEMA5F31C6 device which has a dual-core ARM Cortex A9 (as hard processor system or HPS) and FPGA logic cells, DSP blocks, and memory resources. This allows a developer to easily segment an application, leveraging the flexibility of higher-level programming of the hard processor system as well as the reconfigurability and parallelism provided through the FPGA resources.

To compare the performance of the algorithm in an embedded context, two versions of the code are developed; one that executes purely on the ARM processor (i.e. in software), and one that executes on the ARM processor with some parts offloaded to the FPGA (i.e. software-hardware co-design). The original algorithm uses a number of built-in MATLAB functions, such as *imfilter*, which have to be rewritten from first principles in C. After the software-only implementation is complete, execution profiling is used to identify the bottlenecks, which are suitable candidates for hardware acceleration on the FPGA fabric.

The implemented algorithm has three main stages, as depicted in Figure 1, where each stage implements one layer of the ANN, depicted in Figure 2. The first layer (filtering) uses Gabor filters for multi-scale oriented feature extraction (the circular and regular Gabor filters have the same steps and structure but different coefficients), the NAKA-Rushton equation for contrast enhancement, and local averaging for smoothing. The hidden layer (feature detection) uses adaptive masks and activation functions based on the shunting inhibition neuron model [2], whose weights are learned from training data. The output layer (gender classification) filters the outputs of the previous stage by a set of trained weights to determine the likelihood that the face is male and the likelihood that the face is female. The outputs of the classifier are scaled to the range -1 to 1, where -1 to 0 indicates female and 0 to 1 indicates male.

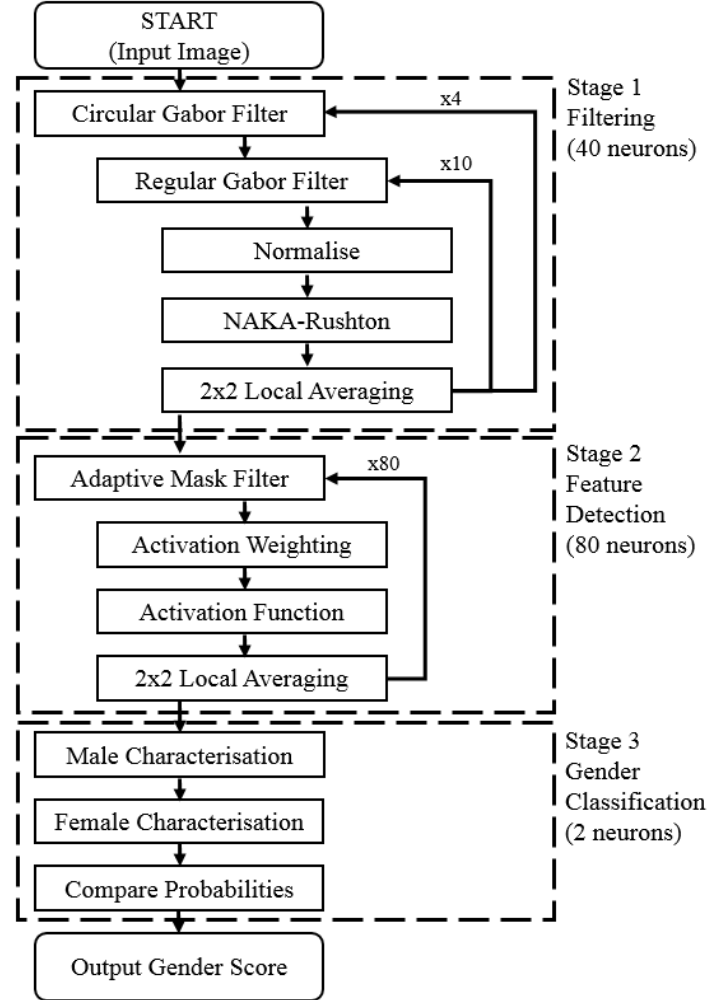


Fig. 1. Flowchart of the gender recognition algorithm

As shown in Table 1, execution profiling using GNU gprof revealed that the primary bottleneck is the Gabor filters. Figure 3 shows how the Gabor filter uses a 5x5 window with real and imaginary components and therefore has a computational complexity of $\Theta(50N)$ for each pass (plus sum and absolute value operations), where N is the number of pixels in the image.

This is especially significant as the first stage of the algorithm requires 44 passes of the filter with various sets of coefficients for each image. This operation became the primary target for hardware acceleration, as the operations on the individual pixels can be executed in parallel in a single logical cycle using combinational circuits, reducing the computation time to $\Theta(N)$ (plus sum and absolute value operations). However, it is important to note that this introduces data transmission overheads between the hard core processor and the FPGA fabric, so the cost must be weighed against the benefits.

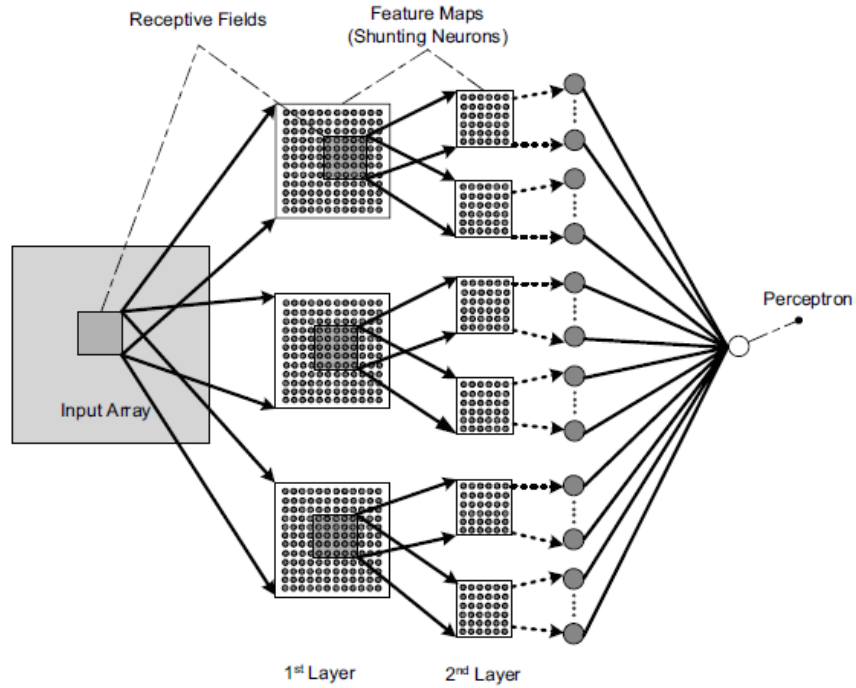


Fig. 2. A three layer binary-connected Shunting Inhibitory Convolutional Neural Network (SICoNNet), from [1]

Table 1. Execution profiling of the algorithm on the ARM Processor, executed over 62 iterations/images

Function	Time per call (ms)	# of calls	Total Time (s)	Time (%)
Circular Gabor and Gabor Filters	2.4	2728	6.56	73.87
Adaptive Mask Filter	0.33	4960	1.66	18.69
2x2 Local Averaging	0.04	7440	0.29	3.27
NAKA-Rushton Equation	0.06	2480	0.14	1.58
Normalisation	0.05	2480	0.12	1.35
Activation Function	0.01	4960	0.06	0.68
Activation Weighting	0.01	4960	0.04	0.45

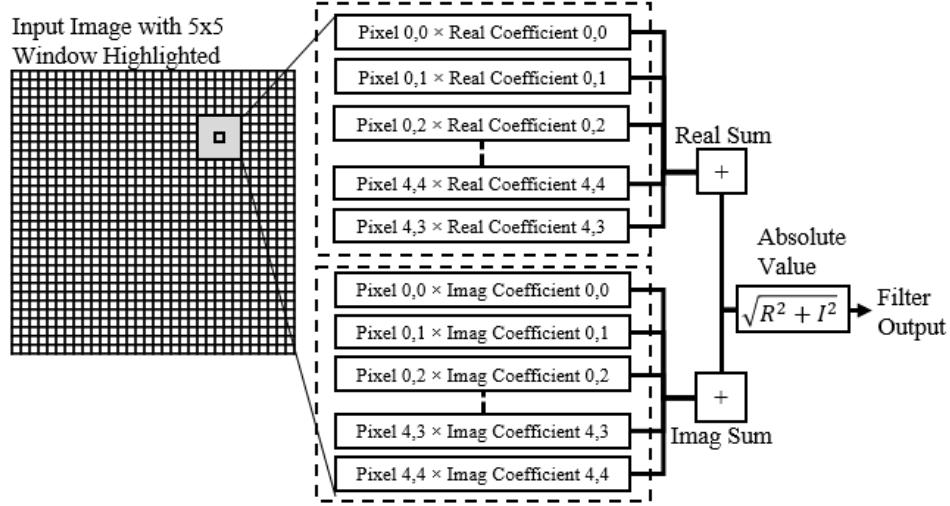


Fig. 3. Diagram of Gabor filter operation

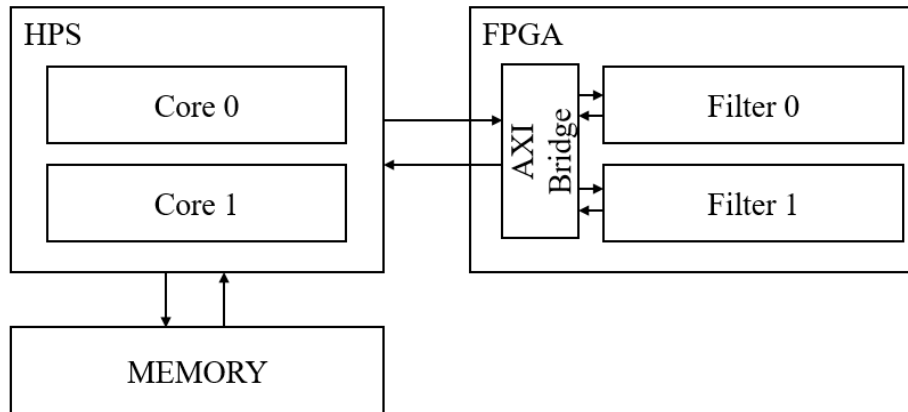


Fig. 4. Computer architecture of dual-core HPS-FPGA system

The filter described in VHDL is a kernelised correlation filter designed to complete part of the *imfilter* function from MATLAB. When passed a set of imaginary and real coefficients (which are stored in memory as fixed point numbers), the filter does the required multiplication operations, sums the products, and then calculates the absolute value by determining the magnitude of the imaginary and real sums. The filter is simulated in Modelsim and tested.

After implementing the filter in VHDL, the challenge becomes passing the data between the hard processor and the FPGA in an efficient manner. Iterating through the image is controlled by the HPS (i.e. ARM Cortex A9), with pixel and coefficient values passed to the FPGA. The HPS-FPGA bridge, which uses the AMBA AXI bus protocol, can at times be the bottleneck, as the handshaking required to retrieve data from the HPS

memory and pass it to the FPGA is non-negligible. A number of steps are taken to mitigate this issue; firstly, the coefficient values and pixel values are concatenated as much as possible to use the full 32-bit bus (also known as data packing), and a shifting window (or sliding window) is used on the filter to minimise the number of data transfers required between the HPS and the FPGA fabric.

Finally, the dual-core nature of the HPS is leveraged by dividing the algorithm into two threads, each responsible for the computations of half of the processing units, running independently to ensure no race conditions. This also utilises two identical filters on the FPGA to calculate output values independently. The overall architecture is presented in Figure 4. Using this approach, the computation time of the algorithm can be significantly reduced by using a processor with a larger number of cores in order to run more threads in parallel.

5 Results

Two test systems are set up: a desktop PC running Cygwin in Windows 7 on a 3.60GHz i7 processor, and a DE1-SoC development board with Cyclone V FPGA (5CSEMA5F31C6) from Altera, which has a dual-core ARM Cortex-A9 processor running Linux at 400MHz and FPGA logic running at 100 MHz.

A test set of 62 cropped images from the FERET database [13], with 30 male images and 32 female images, is used on each platform. This test set was provided in the demonstration code for [1], allowing a fair comparison of performance. The images are resized to 32x32 pixels and converted to greyscale before the processing begins, then pre-loaded into memory for the purposes of testing the algorithm speed. As shown in Table 2, the final implementation using two threads with a shifting window filter achieves a threefold speedup in comparison to the implementation that uses a single core on the HPS only. The performance of each iteration of the system is also included to show how each optimisation improves the performance. In each case the software optimisation from the gcc compiler is left at the default -O0.

Table 2. Execution times for all 62 test images, and per image

Implementation	Total Execution Time (s)	Execution Time per Image (ms)
Desktop PC – MATLAB	8.50	137.10
Desktop PC – C	0.58	9.35
ARM Processor (HPS) Only (single core)	8.88	143.23
Unoptimised HPS-FPGA Implementation	27.03	435.97
Shifting Window HPS-FPGA Implementation	6.08	98.06
Dual-core Shifting Window HPS-FPGA Implementation	2.99	48.23

The actual speed of the image processing is less important than the speed-up; images can be processed faster with higher clock frequencies or more cores, but it is important that a significant speed-up can be achieved by leveraging intelligently implemented hardware acceleration on an FPGA with relatively low development time and cost (compared to pure hardware implementation of the algorithm). With an execution time of 3 seconds per image, using this implementation we can process 20 faces per second, which is double the rate required for the motivating application.

As shown in Table 3, the speed-up is largely attributable to the fact that the Gabor Filter calculations are now performed in hardware. As the filter on the FPGA fabric is implemented combinatorially, results are available one logical clock cycle after all inputs are provided, i.e. 10ns in a 100MHz system. The bottleneck becomes the data transmission between the HPS and FPGA rather than the computation itself. As mentioned previously, the overhead of the HPS-FPGA bridge is significant. This is a good place to start for future optimisations. However, it is important to note that the proportion of the total execution time that is spent on the filter operation is similar for both the Desktop PC (65.52%) and final embedded implementation (65.89%).

Importantly, the loss of precision when moving from the MATLAB algorithm to the HPS+FPGA implementation is small and in many cases negligible. In order to save on computation resources and memory on the FPGA, some of the floating point operations were converted to fixed point.

Table 3. Execution times for the Gabor Filters in Stage 1 of the algorithm

Implementation	Filter Total Time (s)	Filter Execution Time (%)
Desktop PC – C	0.38	65.52
ARM Processor (HPS) Only (Single core)	6.56	73.87
Unoptimised HPS-FPGA Implementation	24.68	91.31
Shifting Window HPS-FPGA Implementation	3.77	62.01
Dual-core Shifting Window HPS-FPGA Implementation	1.97	65.89

As shown in Figure 5, for cases where the gender is more certain (i.e. the absolute value for the score of the detected gender is larger than 0.9), the loss of precision is in the order of 1% or less. As the gender becomes less certain, the error increases and can be as high as 10%. Overall, the accuracy of the algorithm remains 96-97%. Also as shown in Figure 5, the algorithm is capable of working on a variety of image conditions, with different face orientations/poses and lighting, as well as artefacts such as glasses and beards. However, when the image is not cropped properly, and contains either part of a single face or part of more than one face, then the ability of the algorithm to make a robust characterisation of gender decreases (e.g. the last example).







	MATLAB	HPS+FPGA
	0.99973	0.99964
	-0.99524	-0.99623
	0.92512	0.89070
	-0.85450	-0.86704
	-0.70115	-0.75620
	0.58971	0.52091

Fig. 5. A sample of face images and MATLAB / Embedded gender scores – scores larger than 0 are male, scores less than 0 are female. Note that the model has low confidence where the absolute value of the gender score is lower than 0.8, such as in the last two examples. This is usually when part of the face has been obscured, or there are in fact multiple faces in the image.

6 Future Work

Future development can consider new applications as well as further improving the performance and energy efficiency of the algorithms. Further investigation should be done into improving the data transmission rate between the HPS and FPGA fabric, as this has become a significant bottleneck in the system. For example, the FPGA could

be given access to the main memory as done in [12], and simply passed an address from the HPS, so that the FPGA can then retrieve 25 or 50 contiguous values directly from memory. Alternatively, a point-to-point connection could be used between the HPS and FPGA fabric.

An important application of this work may be in facial recognition systems; if an algorithm is attempting to match a face found in an image to a database of known faces, then determining the gender first as a top-level characteristic can greatly reduce the search space which may result in saving time and energy. This can be combined with other facial characteristics such as age category and hair or skin colour or tone to greatly speed up facial recognition in large databases. However, there is an important limitation; since the original neural network was trained with mostly up-right frontal face images, the gender detection algorithm may fail in situations where the camera has an oblique or side view of the face. Since faces in the real world cannot be constrained to always be facing the camera, the algorithm should potentially be retrained to include faces in different orientations. Alternatively, multiple networks can be trained depending on the view of the face, with the weights of the neurons stored in the HPS memory. If a face detector can also determine the orientation of the face, then we can follow the same procedure as described in this paper, but loading different weights as required.

As discovered through implementing the algorithm in two threads, the algorithm is highly parallelisable as each individual neuron could be computed independently, i.e. there are no dependencies between neurons. In this paper we have not considered mapping the ANN structure directly to the FPGA fabric. More parts of the algorithm could be shifted to the FPGA or more cores could be used to leverage more parallelism. However, this should be done only if the performance gain of hardware acceleration is larger than the overhead loss of transmitting data between the HPS and FPGA.

7 Conclusion

In this paper, we have presented an embedded implementation of real-time gender recognition for a targeted marketing application, where the efficacy of a billboard can be improved by determining the gender distribution of the audience. A software-hardware co-design approach is taken to optimise the throughput of a convolutional neural network-based gender recognition algorithm while maintaining a high level of accuracy so that it can operate in real-time. After porting the algorithm from MATLAB to C, the main bottleneck is identified using execution profiling. By moving the Gabor filter into hardware on the FPGA and performing further optimisations such as data packing and using a shifting window, a threefold speedup is achieved compared to a software implementation on an ARM processor alone. This allows 20 faces to be processed per second on an embedded platform, double the throughput required in the motivating application. This implementation satisfies the embedded requirements of the target application.

References

1. Tivive, F. H. C., Bouzerdoum, A.: A Gender Recognition System using Shunting Inhibitory Convolutional Neural Networks. In: International Joint Conference on Neural Networks, pp. 5336-5341. IEEE Press, New York (2006)
2. Tivive, F. H. C., Bouzerdoum, A.: Efficient training algorithms for a class of shunting inhibitory convolutional neural networks. IEEE Transactions on Neural Networks, vol. 16, no. 3, pp. 541-556. IEEE Press, New York (2005)
3. Fregnac, Y., Monier, C., Chavane, F., Baudot P., Graham, L.: Shunting inhibition, a silent step in visual computation. J. Physiology. 97, pp. 441-451 (2003)
4. Wolin, L.: Gender Issues in Advertising—An Oversight Synthesis of Research: 1970–2002. J. Advertising Research. 43, pp. 111-129 (2003)
5. Ng, C. B., Tay, Y. H., Goi, B. M.: Recognizing Human Gender in Computer Vision: A Survey. In: 12th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence, pp. 335-346 (2012)
6. Zheng, J., Lu, B.: A support vector machine classifier with automatic confidence. Neurocomputing, vol. 74, no. 11, pp. 1926-1935 (2011)
7. Shan, C.: Learning local binary patterns for gender classification on real-world face images. Pattern Recognition Letters, vol. 4, no. 33, pp. 431-437 (2012)
8. Azarmehr, R., Laganieri, R., Lee, W. S., Xu, C., Laroche, D.: Real-time Embedded Age and Gender Classification in Unconstrained Video. In: Conference on Computer Vision and Pattern Recognition Workshops, pp. 56-64. IEEE Press, New York (2015)
9. Irick, K. M., DeBole, M., Narayanan V., Gayasen, A.: A Hardware Efficient Support Vector Machine Architecture for FPGA. In: International Symposium on Field-Programmable Custom Computing Machines, pp. 304-305. IEEE Press, New York (2008)
10. Irick, K., DeBole, M., Narayanan, V., Sharma, R., Moon, H., Mummareddy, S.: A Unified Streaming Architecture for Real Time Face Detection and Gender Classification. In: International Conference on Field Programmable Logic and Applications, pp. 267-272. IEEE Press, New York (2007)
11. Ratnakar, A., More, G.: Real time gender recognition on FPGA. International Journal of Scientific & Engineering Research, vol. 6, no. 2, pp. 19-22 (2015)
12. Gudis, E., Lu, P., Berends, D., Kaighn, K., van der Wal, G., Buchanan, G., Chai S., Piacentino, M.: An embedded vision services framework for heterogeneous accelerators. In: Conference on Computer Vision and Pattern Recognition, pp. 598-603. IEEE Press, New York (2013)
13. Phillips, P. J., Moon, H., Rauss, P.J., Rizvi, S.: The FERET evaluation methodology for face recognition algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 10, pp. 1090-1104 (2000)