Embedded Vision Systems: A Review of the Literature

Deepayan Bhowmik and Kofi Appiah

Department of Computing, Sheffield Hallam University, Sheffield, S1 1WB, United Kingdom.
{deepayan.bhowmik, k.e.appiah}@shu.ac.uk

Abstract. Over the past two decades, the use of low power Field Programmable Gate Arrays (FPGA) for the acceleration of various vision systems mainly on embedded devices have become widespread. The reconfigurable and parallel nature of the FPGA opens up new opportunities to speed-up computationally intensive vision and neural algorithms on embedded and portable devices. This paper presents a comprehensive review of embedded vision algorithms and applications over the past decade. The review will discuss vision based systems and approaches, and how they have been implemented on embedded devices. Topics covered include image acquisition, preprocessing, object detection and tracking, recognition as well as high-level classification. This is followed by an outline of the advantages and disadvantages of the various embedded implementations. Finally, an overview of the challenges in the field and future research trends are presented. This review is expected to serve as a tutorial and reference source for embedded computer vision systems.

1   Introduction

Scene understanding and prompt reaction to an event is a critical feature for any time critical computer vision system. The deployment scenarios include a range of applications such as mobile robotics, autonomous cars, mobile and wearable devices or public space surveillance (airport / railway station). Modern vision systems which play a significant role in such interaction process require higher level scene understanding with ultra-fast processing capabilities operating at extremely low power. Currently, such systems rely on traditional computer vision techniques which often follow compute intensive brute-force approaches (slower response time) and prone to fail in environments with limited power, bandwidth and computing resources. The aim of this paper is to review state-of-the-art embedded vision systems available from the literature and in the industry; and therefore to aid researchers for future development.

Research into computer vision has made steady and significant progress in the past two decades. The tremendous progress, coupled with cheap computational power has enabled many portable and embedded devices to operate with vision capabilities. Digital Signal Processing and for that matter Digital Image Processing (DIP) is an exciting area to be involved in today. Having been around for over two decades, it is typically used in application areas where cost and performance are key [7], including the entertainment industry, security surveillance systems, medical systems, automotive industry and defence. DIP systems are often implemented using the ubiquitous general purpose processors (GPPs). The increasing demand for high-speed has resulted in the use of
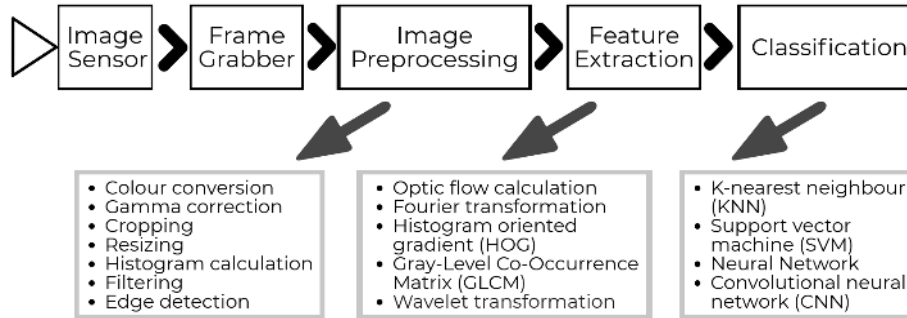
Image Sensor → Frame Grabber → Image Preprocessing → Feature Extraction → Classification

Image Preprocessing:
- Colour conversion
- Gamma correction
- Cropping
- Resizing
- Histogram calculation
- Filtering
- Edge detection

Feature Extraction:
- Optic flow calculation
- Fourier transformation
- Histogram oriented gradient (HOG)
- Gray-Level Co-Occurrence Matrix (GLCM)
- Wavelet transformation

Classification:
- K-nearest neighbour (KNN)
- Support vector machine (SVM)
- Neural Network
- Convolutional neural network (CNN)

Fig. 1. Vision system pipeline.

dedicated Digital Signal Processors (DSPs) and General Purpose Graphics Processing Units (GPGPU); special types of GPP optimised for signal processing algorithms. However, power dissipation is important in almost all DSP-based consumer electronic devices; hence the high-speed, power-hungry GPPs become unattractive. Battery-powered products are highly sensitive to energy consumption, and even line-powered products are often sensitive to power consumption [41]. For hardware acceleration and low power consumption, DIP designers have opted for alternatives like the Field Programmable Gate Array (FPGA) and Application Specific Integrated Circuits (ASIC).

The use of FPGAs in application areas like communication, image processing and control engineering has increased significantly over the past decade [54]. Computer vision and image processing algorithms often perform a large number of inherently parallel operations, and are not good candidates for implementation on machines designed around the von Neumann architecture. Some image processing algorithms have successfully been implemented on embedded system architectures running in real-time on portable devices [35] [45], and relatively small literature has been dedicated to the development of high-level algorithms for embedded hardware [39] [63]. The demand for real-time processing in the design of any practical imaging system has led to the development of the Intel Open source Computer Vision library (OpenCV) for the acceleration of various image processing tasks on GPPs [46]. Many imaging systems rely heavily on the increasing processing speed of today's GPPs to run in real-time.

2 Application specific vision systems

Every embedded vision systems follows a common pipeline of image processing functional blocks as depicted in Fig. 1. The image sensor or camera is the starting point of this pipeline followed by a frame grabber that controls the frame synchronization and frame rate. The raw pixels are then passed for further processing which includes image pre-processing, feature extraction and classification. Within this higher level abstraction various vision systems implemented required functionalities as shown in the figure. Image preprocessing functions are often pixel processing and offer stream computations. However features extraction and classification tasks are complex in nature and usually

Table 1. Embedded vision application areas. UAV: unmanned aerial vehicle; AUV: autonomous underwater vehicle.

| | Robotics | | | Face | Media | Autonomous | Assisted |
	UAV	Mobile Robot	AUV	Detection	Compression	Driving	Living
Cesetti *et al.* [15]	X						
Humenberger *et al.* [31]							X
Yang *et al.* [70]			X				
Chen *et al.* [17]						X	
Velez *et al.* [65]						X	
Yang *et al.* [69]				X			
Lin *et al.* [42]	X						
Oleynikova *et al.* [50]	X						
Flores *et al.* [22]	X			X			
Xu *et al.* [68]		X					
Wang *et al.* [67]			X				
Abeydeera *et al.* [1]					X		
He *et al.* [28]					X		
Basha *et al.* [10]					X		

involves non-deterministic loop conditions. Analysis and optimisations [59] of such complexity with respect to performance and power [13] is an emerging topic of interest and often seen as a trade-off including the choice of the hardware.

Embedded vision systems are usually developed either to accelerate complex algorithms that handles large stream of image data, *e.g.*, stereo matching, video compression etc.; or to minimize power at resource constraint systems including unmanned aerial vehicle (UAV) or autonomous driver assistant systems. While a large number of applications of embedded vision systems can be found in the literature, they can be grouped to major application areas including robotics, face detection applications, multimedia compression, autonomous driving and assisted living as shown in Table 1. Various implementation techniques are proposed in the literature that considers a range of image processing algorithms. Efforts were made either to parallelize the algorithms, or to approximate computing to reduce computational complexities.

While the first approach has implications in performance improvement, the latter ones are more suitable for low power applications. Popular higher level complex image processing algorithms that are used in embedded computer vision literature includes stereo vision, feature extraction and tracking, motion estimation, object detection, scene segmentation and more recent convolutional neural network (CNN). These categories and corresponding literature are captured in Table 2.

3 Embedded vision systems

3.1 Central Processing Unit (CPU)

The widespread adoption of imaging and vision applications in industrial automation, robotics and surveillance calls for a better way of implementing such techniques for

Table 2. Common high level algorithms used in embedded vision systems.

	Feature point extraction	Stereo vision	Motion estimation	Object detection	Scene segmentation	CNN
Park *et al.* [51]	X					
Jin *et al.* [36]		X				
Chen *et al.* [17]	X				X	
Belbachir *et al.* [11]				X		
Banz *et al.* [8]		X				
Cesetti *et al.* [15]	X					
Humenberger *et al.* [32]		X				
Lin *et al.* [42]	X			X	X	
Oleynikova *et al.* [50]		X				
Flores *et al.* [22]	X					
Ttofis *et al.* [64]		X				
He *et al.* [28]			X			
Basha *et al.* [10]			X			
Liu *et al.* [43]				X		X
Zhao *et al.* [74]				X		X

real-time purposes. The need to address the gap in knowledge for students who have either studied computer vision or microelectronics to fill positions in the industry requiring both expertise has been address with the introduction of various CPU based platforms like Beagleboard [47] and Raspberry-Pi [48]. Hashmi *et al.* [27] used a beagleboard-xM low-power open-source hardware to prototype a real-time copyright protection algorithm. A human tracking system which reliably detect and track human motion has been implemented on a beagleboard-xM [24]. In [5], a LeopardBoard has been used to implement an efficient edge-detection algorithm for tracking activity level in an indoor environment. Similarly, Sharma and Kumar [56] presented an image enhancement algorithm on a beagleboard, mainly for monitoring the health condition of an individual. To demonstrate the efficiency of embedded image processing Sahani and Mohanty [55] showcased various computer vision applications developed on Raspberry-Pi. The system uses a camera powered by the raspberry-pi with a resolution of 1280×720 to detect text and images in real-time. Various other computer vision algorithm have been implemented on small dedicated platforms using Raspberry-Pi. In [30], a robot with on-board camera for carrying lightweight objects is presented and uses raspberry-pi to process the camera data in aid of navigation. Other robotic systems like [37] [44] have all implemented some vision based algorithms on a Raspberry-Pi because of its portability and ease of programmability.

3.2 Graphic Processing Unit (GPU)

The parallel nature of GPUs have made them a choice for the acceleration of many computer vision algorithms [66]. Coupled with the emerging heterogeneous programming models like OpenCL, GPGPU has been enabled on mobile devices. To explore the capabilities of mobile GPU for the acceleration of computer vision algorithms, Wang *et*

al. [66] presented and exemplar-based inpainting algorithm for object removal. Rister *et al.* [53] presented an implementation of the Scale-Invariant Feature Transform feature detection algorithm on a mobile based GPU to achieved 7× speed-up over optimised GPP implementation. A face detection and recognition system implementation on two GPU architectures are presented in [71] with reported speed-up of approximately 3.7×. A mobile GPU based object detection algorithm with twofold speed-up compared to a similar implementation on a mobile GPP is presented in [3]. The implementation also reported energy savings of up to 84% compared to a smartphone GPP. A GPU enabled architecture for scaling up convolutional networks have been presented in [62]. The explored networks [62] are trained with stochastic gradient distributed machine learning system using 50 replicas on a NVidia Kepler GPU. Deep learning or Convolutional Neural Network (CNN) has become popular in the fields of machine learning and computer vision, because of it's high performance in object detection [33]. Using only GPP, a complex CNN may require more than one month to train [19]. GPUs offer approximately ten fold speed-up compared to GPP, which is demonstrated in [33] for faster training and testing. A number of other computer vision and image processing algorithms [57] [34] [9] have been implemented on GPU mainly to accelerate them for real time needs.

3.3 Field Programmable Gate Array (FPGA)

FPGAs are successfully used in many application areas, including embedded computer vision and image processing. The key advantage of FPGAs over conventional CPUs or GPUs is configurability. Resource allocation and memory hierarchy on general purpose processors must perform well across a range of applications, whereas FPGA designs leave many of those decisions to the application designer to optimally use logic gates to implement one specific application. Moreover, they can be significantly faster as their nature supports fine-grained, massively parallel and pipelined execution. FPGAs allows stream processing from camera input and offers parallel execution of processing blocks that resembles the vision system pipeline as depicted in Fig. 1. Various forms of parallelism, *e.g.*, pipeline, task or data parallelism were exploited in FPGA based vision systems [59]. Additionally FPGAs are known for low power execution and vision system designers often exploit this characteristics by using multi-clock domain design paradigm [13]. However on the downside, FPGAs are blamed on programmability aspect as FPGAs are most often specified directly in low level less expressive hardware description languages such as Verilog or VHDL.

The intrinsic parallel architecture of FPGAs have also been exploited in a number of application areas including high level feature classification with conventional neural networks [60] [29], convolutional neural networks [52] [18] [12] and architecture specific neural networks [49] [4]. A variant of self-oganising map designed specifically for FPGA is presented in [4] and tested on two computer vision applications; character recognition and appearance-based object identification. The implementation in [4] was achieved using Xilinx Virtex-4 XC4VLX160 and capable being trained with approximately 25,000 patterns every second. Embedded vision systems, implemented on FPGAs are usually evaluated on a few objective measurements including 1) performance measured in throughput (*e.g.*, frames per second or fps); 2) clock frequency; 3)

Table 3. A comparative overview of the FPGA metrics used in embedded computer vision.

	Frame size	Frame Rate	Max. Clock Frequency	Target Devices
Jin *et al.* [36]	640×480	230 fps	93 MHz	Xilinx Virtex-4
Appiah *et al.* [6]	640×480	35 fps	65 MHz	Xilinx Virtex-4
Banz *et al.* [8]	640×480	30 fps	39 MHz	Xilinx Virtex-5
Oleynikova *et al.* [50]	640×480	60 fps	-	Xilinx Artix-7
Ttofis *et al.* [64]	1280×720	60 fps	103 MHz	Xilinx Kintex-7
He *et al.* [28]	7680×4320	30 fps	188 MHz	Altera Stratix II
Abeydeera *et al.* [1]	4096×2160	30 fps	150 MHz	Xilinx Zynq 7045
Tanabe *et al.* [63]	640×480	349 fps	228 MHz	Xilinx Virtex-6
Albo-Canals *et al.* [2]	177×144	1562 fps	30 MHz	Actel IGLOO
Bhowmik *et al.* [13]	320×240	52 fps	85 MHz	Xilinx Zynq 7020

input image frame size; 4) FPGA resource usage (*e.g.*, DSP, BRAM, FF/LUTs) and 5) power consumption. Power consumption on FPGAs consists of *a) static power*, which is directly proportional to the amount of used logic; and *b) dynamic power*, which is a weighted sum of several components (these include clock signal propagation power, proportional to clock frequency; signals power, proportional to signal switching rates, among others). The implementation relies on available programmable logic gates available on different FPGA boards from handful of manufacturers, including Xilinx and Altera (now Intel). Table 3 provides a comparative overview of these measurements metrics reported in the literature that are referred earlier in Section 2.

3.4 ASIC

Vision based applications and systems are typically associated with high computational cost, slow when implemented on general purpose processors and not very useful in real-time applications. To address some of theses problems, mainly the real-time requirements, most researchers have resulted to the use of dedicated and application specific systems. In [61], Sugiura *et al* used an application specific instruction-set processor to execute a lossless data compression method as part of a visual prosthesis systems. Deep networks, models for understanding the content of images, videos and audio have been used successfully in various application [40] with relatively high computational cost. Gokhale *et al* [26] presented a scalable, low-power co-processor for enabling real-time execution of deep neural networks on mobile devices. This was implemented using a large number of parallel operators, optimised to process multiple streams of information. The implementation presented in [26] shows that image understanding with deep networks can be accelerated on custom hardware to achieve better performance per watt. Chen *et al.* [16] presented an application specific integrated circuit accelerator on a 65nm scale technology, for large-scale convolutional and deep neural networks capable of performing 452 GOP/s of key neural network operations in a small footprint. A convolution chip built on 0.35 μm CMOS technology for event-driven vision sensing and processing is presented in [14].

4 Future trends and conclusions

In this paper we made a modest effort to review embedded computer vision systems that satisfy application specific constraints *e.g.*, performance or power. The literature is scattered and covers a range of application areas, vision algorithms and target hardware. This paper made an effort to categorize them in an orderly fashion. We identified two emerging trends (described below) in this domain namely, heterogeneous computing and bi-inspired computing for efficient vision systems.

4.1 Heterogeneous computing for vision systems

Current computer vision algorithms are highly complex and consist of different functional blocks that are suitable for a variety of targets *i.e.*, CPUs, GPUs or FPGAs. Therefore, designing computer vision systems for single target hardware platform is inefficient and does not necessarily meet performance and power budgets especially for embedded and remote operations. A heterogeneous architecture is a natural alternative but manifests new challenges:

- design choices to dissect the algorithm according to their suitability for the target hardware,
- interoperability and data flow synchronisations between functional units as different blocks may have different timing constraints.
- programmability and coordination between different hardware platforms. There is a need for unified programming environment.

Although recently hardware manufacturers launched new heterogeneous products, *e.g.*, Xilinx Zynq Ultrascale+ MPSoC[1] (CPU, GPU and FPGA) and Altera SoC products[2] (CPU and FPGA), these are not fully exploited in computer vision domain (except handful of recent work, *e.g.*, Zhang *et al.* [73]) as majority of the existing algorithms are not designed to target heterogeneous platforms. Consideration of target hardware during the algorithmic development cycle is not always necessary and the domain experts often prototype new algorithms using library-rich languages such as MATLAB. However, efficient deployment of these prototypes on a heterogeneous hardware is challenging. Asynchronous data process network [20] may provide a plausible solution to this problem, however requires further research.

4.2 Biologically inspired vision systems

The ability to detect moving objects in a scene is a fundamental problem in computer vision. This is a baseline problem that requires detection accuracy as well as computational efficiency to guarantee a successful high level processing in behavioural or event analysis [72]. Various background subtraction methods [25] have been proposed and

---

[1] https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html
[2] https://www.altera.com/products/soc/overview.html

proven to be successful for detecting moving objects with the use of stationary cameras. These methods build statistical background models and extract moving objects by finding regions which do not have similar characteristics to the background model. Human visual systems processes a very high volume of data and hence it is often selective and activity driven (responsive to the scene event).

The high volume data problem is also faced by many modern technical systems like computer vision systems which need to deal with a multitude of image pixels at any point in time. Physiological research has illustrated that biological vision systems use neuronal circuits to extract movement in the visual scenes [38]. Biological visual systems are intrinsically complex hierarchical processing systems with diverse specialised neurons, displaying very powerful specific biological processing functionalities that traditional computer vision techniques have not yet fully emulated [38]. Another important finding during the last decades, that most neuromorphic designers may overlook is the fact that processing of the visual information is not serial but rather highly parallel [23] and hence such implementations should target parallel architectures.

A concept proposed and implemented in [21], shows that motion information can be capture with the use of one retina sheet and two LGN sheets (one ON and one OFF). Orientation preference has successfully been modelled using a Gain Control, Adaptation, Laterally (GCAL) model consisting of four two-dimensional sheets. Solari *et al.* [58], presented a feed-forward model based on the biological visual system to solve motion estimation problem. The model integrates media temporal (MT) neurons for estimation of optical flow by extending it into a scalable framework. What is missing from their model is the feedback capabilities as perceived in the visual pathway, but the results are very promising and acts as a good starting point for building bio-inspired scalable computer vision algorithms.

Acknowledgement

References

1. Abeydeera, M., Karunaratne, M., Karunaratne, G., De Silva, K., Pasqual, A.: 4K real-time HEVC decoder on an FPGA. IEEE Trans. on Circuit and Systems for Video Technology 26(1), 236–249 (2016)
2. Albo-Canals, J., Ortega, S., Perdices, S., Badalov, A., Vilasis-Cardona, X.: Embedded low-power low-cost camera sensor based on FPGA and its applications in mobile robots. In: 19th IEEE International Conference on Electronics, Circuits, and Systems. pp. 336–339 (2012)
3. Andargie, F.A., Rose, J., Austin, T., Bertacco, V.: Energy efficient object detection on the mobile gp-gpu. In: 2017 IEEE AFRICON. pp. 945–950 (Sept 2017)
4. Appiah, K., Hunter, A., Dickinson, P., Meng, H.: Implementation and applications of tri-state self-organizing maps on fpga. IEEE Transactions on Circuits and Systems for Video Technology 22(8), 1150–1160 (Aug 2012)
5. Appiah, K., Hunter, A., Lotfi, A., Waltham, C., Dickinson, P.: Human behavioural analysis with self-organizing map for ambient assisted living. In: 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). pp. 2430–2437 (July 2014)

6. Appiah, K., Hunter, A., Dickinson, P., Meng, H.: Accelerated hardware video object segmentation: From foreground detection to connected components labelling. Comput. Vis. Image Underst. 114(11), 1282–1291 (Nov 2010)
7. Athi, M.V., Zekavat, S.R., Struthers, A.A.: Real-time signal processing of massive sensor arrays via a parallel fast converging svd algorithm: Latency, throughput, and resource analysis. IEEE Sensors Journal 16(8), 2519–2526 (April 2016)
8. Banz, C., Hesselbarth, S., Flatt, H., Blume, H., Pirsch, P.: Real-time stereo vision system using semi-global matching disparity estimation: Architecture and FPGA-implementation. In: Int'l Conf. on Embedded Computer Systems (SAMOS). pp. 93–101 (2010)
9. Barbu, A., She, Y., Ding, L., Gramajo, G.: Feature selection with annealing for computer vision and big data learning. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(2), 272–286 (Feb 2017)
10. Basha, S.M., Kannan, M.: Design and implementation of low-power motion estimation based on modified full-search block motion estimation. Journal of Computational Science (2016)
11. Belbachir, A.N., Hofstatter, M., Litzenberger, M., Schon, P.: High-speed embedded-object analysis using a dual-line timed-address-event temporal-contrast vision sensor. IEEE Transactions on Industrial Electronics 58(3), 770–783 (2011)
12. Bettoni, M., Urgese, G., Kobayashi, Y., Macii, E., Acquaviva, A.: A convolutional neural network fully implemented on fpga for embedded platforms. In: 2017 New Generation of CAS (NGCAS). pp. 49–52 (Sept 2017)
13. Bhowmik, D., Garcia, P., Wallace, A., Stewart, R., Michaelson, G.: Power efficient dataflow design for a heterogeneous smart camera architecture. In: Conference on Design and Architectures for Signal and Image Processing (DASIP 2017) (August 2017)
14. C.-Mesa, L., A.-Jimenez, A., Z.-Ramos, C., S.-Gotarredona, T., L.-Barranco, B.: A 32x32 pixel convolution processor chip for address event vision sensors with 155 ns event latency and 20 meps throughput. IEEE Trans. on Circuits and Sys. I: Regular Papers 58(4) (2011)
15. Cesetti, A., Frontoni, E., Mancini, A., Zingaretti, P., Longhi, S.: A vision-based guidance system for uav navigation and safe landing using natural landmarks. Journal of intelligent and robotic systems 57(1-4), 233 (2010)
16. Chen, T., Du, Z., Sun, N., Wang, J., Wu, C., Chen, Y., Temam, O.: Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning. In: Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems. pp. 269–284. ASPLOS '14, ACM, New York, NY, USA (2014)
17. Chen, Y.L., Wu, B.F., Huang, H.Y., Fan, C.J.: A real-time vision system for nighttime vehicle detection and traffic surveillance. IEEE Trans. on Industrial Electronics 58(5) (2011)
18. Colangelo, P., Luebbers, E., Huang, R., Margala, M., Nealis, K.: Application of convolutional neural networks on intel; xeon; processor with integrated fpga. In: 2017 IEEE High Performance Extreme Computing Conference (HPEC). pp. 1–7 (Sept 2017)
19. Courbariaux, M., Bengio, Y.: Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1. CoRR abs/1602.02830 (2016)
20. Eker, J., Janneck, J.: Cal language report: Specification of the cal actor language (2003)
21. Fischer, T.: Model of all known spatial maps in primary visual cortex. Master's thesis, University of Edinburghs (2014)
22. Flores-Delgado, J., Martínez-Santos, L., Lozano, R., Gonzalez-Hernandez, I., Mercado, D.: Embedded control using monocular vision: Face tracking. In: Unmanned Aircraft Systems (ICUAS), 2017 International Conference on. pp. 1285–1291. IEEE (2017)
23. Frintrop, S., Rome, E., Christensen, H.I.: Computational visual attention systems and their cognitive foundations: A survey. ACM Trans. Appl. Percept. 7(1), 6:1–6:39 (Jan 2010)
24. Gantala, A., Nehru, K., Telagam, N., Anjaneyulu, P., Swathi, D.: Human tracking system using beagle board -xm. Int'l Journal of Applied Engg. Research 12(16), 5665–5669 (2017)

25. Ge, W., Guo, Z., Dong, Y., Chen, Y.: Dynamic background estimation and complementary learning for pixel-wise foreground/background segmentation. Pattern Recognition 59(Supplement C), 112 – 125 (2016)
26. Gokhale, V., Jin, J., Dundar, A., Martini, B., Culurciello, E.: A 240 g-ops/s mobile coprocessor for deep neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2014)
27. Hashmi, M.F., Shukla, R.J., Keskar, A.G.: Platform independent real time copyright protection embedding and extraction algorithms on android and embedded framework. In: Int'l Symp. on Signal Processing and Info. Technology (ISSPIT). pp. 000189–000194 (2014)
28. He, G., Zhou, D., Li, Y., Chen, Z., Zhang, T., Goto, S.: High-throughput power-efficient vlsi architecture of fractional motion estimation for ultra-hd hevc video encoding. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 23(12), 3138–3142 (2015)
29. Ho, S.M.H., Hung, C.H.D., Ng, H.C., Wang, M., So, H.K.H.: A parameterizable activation function generator for fpga-based neural network applications. In: IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM) (2017)
30. Horak, K., Zalud, L.: Image processing on raspberry pi for mobile robotics. International Journal of Signal Processing Systems 4(2), 1–5 (2016)
31. Humenberger, M., Schraml, S., Sulzbachner, C., Belbachir, A.N., Srp, A., Vajda, F.: Embedded fall detection with a neural network and bio-inspired stereo vision. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, (2012)
32. Humenberger, M., Zinner, C., Weber, M., Kubinger, W., Vincze, M.: A fast stereo matching algorithm suitable for embedded real-time systems. Computer Vision and Image Understanding 114(11), 1180–1202 (2010)
33. Islam, S.M.S., Rahman, S., Rahman, M.M., Dey, E.K., Shoyaib, M.: Application of deep learning to computer vision: A comprehensive study. In: 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV). pp. 592–597 (May 2016)
34. Jain, V., Patel, D.: A gpu based implementation of robust face detection system. Procedia Computer Science 87(Supplement C), 156 – 163 (2016), fourth International Conference on Recent Trends in Computer Science & Engineering (ICRTCSE 2016)
35. Jasani, B.A., Lam, S.K., Meher, P.K., Wu, M.: Threshold-guided design and optimization for harris corner detector architecture. IEEE TCSVT PP(99), 1–1 (2017)
36. Jin, S., Cho, J., Dai Pham, X., Lee, K.M., Park, S.K., Kim, M., Jeon, J.W.: FPGA design and implementation of a real-time stereo vision system. IEEE transactions on circuits and systems for video technology 20(1), 15–26 (2010)
37. Jing, X., Gong, C., Wang, Z., Li, X., Ma, Z., Gong, L.: Remote Live-Video Security Surveillance via Mobile Robot with Raspberry Pi IP Camera. In: International Conference on Intelligent Robotics and Applications. pp. 776–788. Springer (2017)
38. Kerr, D., McGinnity, T., Coleman, S., Clogenson, M.: A biologically inspired spiking model of visual processing for image feature detection. Neurocomput. 158(C), 268–280 (Jun 2015)
39. Khan, M.U.K., Khan, A., Kyung, C.M.: Ebscam: Background subtraction for ubiquitous computing. IEEE Transactions on Very Large Scale Integration Systems 25(1), 35–47 (Jan 2017)
40. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM 60(6), 84–90 (May 2017)
41. Li, Y., Chen, L., Benson, B., Kastner, R.: Determining the Suitability of FPGAs for a Low-Cost, Low-Power Underwater Acoustic Modem. In: Proceedings of Future Control and Automation. pp. 509–517. Springer-Verlag Berlin Heidelberg (2012)
42. Lin, F., Dong, X., Chen, B.M., Lum, K.Y., Lee, T.H.: A robust real-time embedded vision system on an unmanned rotorcraft for ground target following. IEEE Trans. on Industrial Electronics 59(2), 1038–1049 (2012)

43. Liu, Z., Dou, Y., Jiang, J., Xu, J., Li, S., Zhou, Y., Xu, Y.: Throughput-Optimized FPGA Accelerator for Deep Convolutional Neural Networks. ACM Transactions on Reconfigurable Technology and Systems (TRETS) 10(3), 17 (2017)

44. Loureiro, R., Lopes, A., Carona, C., Almeida, D., Faria, F., Garrote, L., Premebida, C., Nunes, U.J.: Isr-robothead: Robotic head with lcd-based emotional expressiveness. In: 2017 IEEE 5th Portuguese Meeting on Bioengineering (ENBENG). pp. 1–4 (Feb 2017)

45. Ma, X., Borbon, J.R., Najjar, W., Roy-Chowdhury, A.K.: Optimizing hardware design for human action recognition. In: 2016 26th International Conference on Field Programmable Logic and Applications (FPL). pp. 1–11 (Aug 2016)

46. Mazumdar, A., Moreau, T., Kim, S., Cowan, M., Alaghi, A., Ceze, L., Oskin, M., Sathe, V.: Exploring computation-communication tradeoffs in camera systems. 2017 IEEE International Symposium on Workload Characterization (IISWC) (2017)

47. Morison, G., Jenkins, M.D., Buggy, T., Barrie, P.: An implementation focused approach to teaching image processing and machine vision - from theory to beagleboard. In: European Embedded Design in Education and Research Conference (EDERC). pp. 274–277 (2014)

48. Nguyen, H.Q., Loan, T.T.K., Mao, B.D., Huh, E.N.: Low cost real-time system monitoring using raspberry pi. In: 2015 Seventh International Conference on Ubiquitous and Future Networks. pp. 857–859 (July 2015)

49. Nurvitadhi, E., Sheffield, D., Sim, J., Mishra, A., Venkatesh, G., Marr, D.: Accelerating binarized neural networks: Comparison of fpga, cpu, gpu, and asic. In: 2016 International Conference on Field-Programmable Technology (FPT). pp. 77–84 (Dec 2016)

50. Oleynikova, H., Honegger, D., Pollefeys, M.: Reactive avoidance using embedded stereo vision for mav flight. In: Int'l Conf. on Robotics and Automation (ICRA). pp. 50–56 (2015)

51. Park, J.S., Kim, H.E., Kim, L.S.: A 182 mW 94.3 f/s in Full HD Pattern-Matching Based Image Recognition Accelerator for an Embedded Vision System in 0.13-mm CMOS Technology. IEEE Trans. on Circuit and Systems for Video Technology 23(5), 832–845 (2013)

52. Qiu, J., Wang, J., Yao, S., Guo, K., Li, B., Zhou, E., Yu, J., Tang, T., Xu, N., Song, S., Wang, Y., Yang, H.: Going Deeper with Embedded FPGA Platform for Convolutional Neural Network. In: Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. pp. 26–35. FPGA '16, ACM, New York, NY, USA (2016)

53. Rister, B., Wang, G., Wu, M., Cavallaro, J.R.: A fast and efficient sift detector using the mobile GPU. In: Proc. IEEE ICASSP. pp. 2674–2678 (2013)

54. Romoth, J., Porrmann, M., Ruckertr, U.: Survey of FPGA applications in the period 2000 2015. Tech. rep., Bielefeld University, Germany (03 2017)

55. Sahani, M., Mohanty, M.N.: Realization of different algorithms using raspberry pi for realtime image processing application. In: Intelligent Computing, Communication and Devices, pp. 473–479. Springer (2015)

56. Sharma, G., Kumar, K.: Prototyping of image enhancement algorithms using beagle board for rural health monitoring. In: International Conference on Recent innovations in Science, Management, Education and Technology. pp. 346–358 (August 2016)

57. Singh, R., Ranasinghe, L.: Accelerating computer vision on mobile embedded platforms. In: 2016 IEEE Region 10 Conference (TENCON). pp. 3131–3134 (Nov 2016)

58. Solari, F., Chessa, M., Medathati, K., Kornprobst, P.: What can we expect from a classical V1-MT feedforward architecture for optical flow estimation? Signal Processing: Image Communication 49(1), 250–257 (Oct 2015)

59. Stewart, R.J., Bhowmik, D., Wallace, A.M., Michaelson, G.: Profile Guided Dataflow Transformation for FPGAs and CPUs. Signal Processing Systems 87(1), 3–20 (2017)

60. Su, J., Liu, J., Thomas, D.B., Cheung, P.Y.: Neural network based reinforcement learning acceleration on fpga platforms. SIGARCH Comput. Archit. News 44(4), 68–73 (Jan 2017)

61. Sugiura, T., Yu, J., Takeuchi, Y., Imai, M.: A low-energy asip with flexible exponential golomb codec for lossless data compression toward artificial vision systems. In: 2015 IEEE Biomedical Circuits and Systems Conference (BioCAS). pp. 1–4 (Oct 2015)

62. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)

63. Tanabe, Y., Maruyama, T.: Fast and Accurate Optical Flow Estimation Using FPGA. SIGARCH Comput. Archit. News 42(4), 27–32 (Dec 2014)

64. Ttofis, C., Kyrkou, C., Theocharides, T.: A low-cost real-time embedded stereo vision system for accurate disparity estimation based on guided image filtering. IEEE Transactions on Computers 65(9), 2678–2693 (2016)

65. Velez, G., Cortés, A., Nieto, M., Vélez, I., Otaegui, O.: A reconfigurable embedded vision system for advanced driver assistance. J. Real-Time Image Processing 10(4), 725–739 (2015)

66. Wang, G., Xiong, Y., Yun, J., Cavallaro, J.R.: Accelerating computer vision algorithms using opencl framework on the mobile gpu - a case study. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 2629–2633 (May 2013)

67. Wang, K., Yu, J.: An embedded vision system for robotic fish navigation. In: Int'l Conf. on Computer Application and System Modeling (ICCASM). vol. 4, pp. V4–333. IEEE (2010)

68. Xu, H., Shen, Y.p.: Target tracking control of mobile robot in diversified manoeuvre modes with a low cost embedded vision system. Journal of Industrial Robot 40(3), 275–287 (2013)

69. Yang, M., Crenshaw, J., Augustine, B., Mareachen, R., Wu, Y.: Adaboost-based face detection for embedded systems. Computer Vision and Image Understanding 114(11) (2010)

70. Yang, X., Wu, Z., Yu, J.: Design and implementation of a robotic shark with a novel embedded vision system. In: Robotics and Biomimetics (ROBIO), 2016 IEEE International Conference on. pp. 841–846. IEEE (2016)

71. Yi, S., Yoon, I., Oh, C., Yi, Y.: Real-time integrated face detection and recognition on embedded gpgpus. In: 2014 IEEE 12th Symposium on Embedded Systems for Real-time Multimedia (ESTIMedia). pp. 98–107 (Oct 2014)

72. Yun, K., Choi, J.Y.: Robust and fast moving object detection in a non-stationary camera via foreground probability based sampling. In: 2015 IEEE International Conference on Image Processing (ICIP). pp. 4897–4901 (Sept 2015)

73. Zhang, B., Zhao, C., Mei, K., Zheng, N., et al.: Hierarchical and parallel pipelined heterogeneous soc for embedded vision processing. IEEE Trans. on Circuit and Systems for Video Technology (2017)

74. Zhao, R., Niu, X., Wu, Y., Luk, W., Liu, Q.: Optimizing CNN-Based Object Detection Algorithms on Embedded FPGA Platforms. In: International Symposium on Applied Reconfigurable Computing. pp. 255–267. Springer (2017)