FEN ve MÜHENDİSLİK BİLİMLERİ DERGİSİ

Cilt/Vol: 7 Sayı/No: 1 Yıl/Year: 2025 Araştırma Makalesi/Research Article e-ISSN: 2667-7989

https://doi.org/10.47112/neufmbd.2025.71

FPGA Donanımı için Tek Taramalı Bağlantılı Bileşen Etiketlemenin Uygulanması

İsmail YABANOVA[1*] Tarık ÜNLER [2]

[1] Manisa Celal Bayar Üniversitesi, Hasan Ferdi Turgutlu Teknoloji Fakültesi, Elektrik Mühendisliği Bölümü, Manisa, Türkiye

[2] Necmettin Erbakan Üniversitesi, Havacılık ve Uzay Bilimleri Fakültesi, Havacılık Elektrik ve Elektroniği Bölümü, Konya, Türkiye

Makale Bilgisi

Geliş Tarihi: 13.05.2024
Kabul Tarihi: 15.07.2024
Yayın Tarihi: 30.04.2025

Anahtar Kelimeler:
Bağlantılı Bileşen Etiketleme,
Blob Analizi,
FPGA.

ÖZET

Görüntü işleme dijital görüntüyü işleyip analiz ederek belirli bilgileri çıkarmayı ve nesnelerin tanımlanmasını hedefleyen bilgisayar bilimidir. Görüntüleme sistemleri günümüzde tıp, savunma sanayi, güvenlik, otomotiv ve otomasyon alanları başta olmak üzere hayatımızın her alanında yer almaktadır. Bundan dolayı görüntünün belirlenen amaçlar doğrultusunda işlenmesi gerekmektedir. Blob tespiti görüntü işlemede nesne tanıma, algılama için kullanılan bir kavramdır ve görüntüdeki bir nesnenin sınırlarını belirleyen bir grup pikseli ifade eder. Blob analizi için farklı algoritmalar geliştirilmiştir. Bu algoritmalardan birisi görüntü işlemede yaygın olarak kullanılan ve amacı bir nesneye ait tüm piksellere aynı etiketi atamak olan bağlantılı bileşen etiketleme algoritmasıdır. Bu çalışmada görüntüdeki nesnelerin tespiti için bağlantılı bileşen etiketleme algoritması FPGA yapısı üzerinde uygulanmak üzere hazırlanmış ve simülasyon ortamında test edilmiştir. Bir görüntü karesinin tamamının aynı anda işlenmesi bellek kaynakları açısından verimsiz olduğundan dolayı görüntü satır satır okunup işlenmiştir. Görüntü karesinden tek geçişte her piksel etiketlenmiş ve bağlantılı pikseller birleştirilerek her blobun alanı, sınırları ve ağırlık merkezi hesaplanmıştır. Bu şekilde PL-PS yapılarının bir arada kullanılmasına imkân sağlayacak CCL algoritmasının uygulanması için altyapı sağlanmıştır. İlerde yapılacak çalışmalarda blob tespiti işlemi ZYNQ yapısının PS ve PL kısımlarına uygulanarak verimli bir sistem geliştirilmesi hedeflenmektedir.

Implementation of Single-Scan Connected Component Labeling for FPGA Hardware

Article Info

Received: 13.05.2024
Accepted: 15.07.2024
Published: 31.12.2024

Keywords:
Blob analysis,
Connected Component Labeling,
FPGA.

ABSTRACT

Image processing is a branch of computer science that tries to extract information and identify objects from digital photographs through processing and analysis. Today, imaging systems are used in almost every aspect of our lives, particularly in medicine, the defense industry, security, automobile, and automation. As a result, the image must be processed in accordance with the intended purposes. Blob detection is a concept used in image processing to recognize and detect objects. It refers to a group of pixels that define the borders of an object in the image. Blob analysis algorithms vary. One of these techniques is the linked component labeling algorithm, which is commonly used in image processing and assigns the same label to all pixels of an object. In this study, the connected component labeling algorithm for detecting objects in the image was prepared to be implemented on the FPGA structure and tested in a simulation environment. Since processing a full image frame at once is inefficient in terms of memory resources, the image was read and processed line by line. Each pixel was labeled in a single pass through the image frame, and the area, boundaries, and centroid of each blob were determined by combining the corresponding pixels. This provides the foundation for the application of the CCL algorithm, allowing the use of PL-PS structures together. Future research will focus on constructing an efficient system that applies the blob identification procedure to the PS and PL components of the ZYNQ structure.

To cite this article:
Yabanova, İ. &Ünler, T. (2025). Implementation of single-scan connected component labeling for FPGA hardware, *Necmettin Erbakan University Journal of Science and Engineering*, *7*(1), 12-21. https://doi.org/10.47112/neufmbd.2025.71

***Corresponding Author:** İsmail Yabanova, *iyabanova@gmail.com*

INTRODUCTION

Camera systems have grown commonplace in many aspects of our life as technology has advanced. The images must be processed in order to get the desired information. As a result, image processing algorithms are continually improving. The majority of this improvement is due to the development of new data processing methods, but the development of image sensors has resulted in higher refresh rates and image resolutions over time. The development of image sensors has improved not only image quality but also the efficiency of vision algorithms. However, increasing the resolution increases the amount of data to be processed [1].

Image processing has been useful in practically any field. When combined with optimization methods, they produce particularly successful results. For example, it has been reported that multi-level image thresholding, a popular image processing technique for segmenting images into highly meaningful features, can be combined with metaheuristic optimization algorithms to achieve successful results in terms of calculation time [2]. Another example of using image processing tools in a different field is to generate 3D body measurements from 2D images [3]. It will be a useful application for users, particularly if electronic purchasing becomes more widespread. Artificial intelligence is another topic that is fast evolving today. The most important factor in this progress is the development of high-performance computing equipment. Today, artificial intelligence is widely employed in a variety of industries, including health, the defense industry, image processing, security, voice and text processing, autonomous cars, and weather forecasting [4].

The connected component labeling (CCL) algorithm, which assigns a unique label to all pixels of each object (connected components) in a binary image, is critical for distinguishing different objects in the image and is required for image analysis and object recognition. As a result, CCL is a critical process in image analysis, understanding, perception, pattern recognition, segmentation, and computer vision [5-9].

Raster approaches are often preferred for the CCL algorithm, as they involve sequential processes based on label assignment and merging of equivalent labels [10]. Because the physical design and connectivity of connected components might be complicated, the CCL algorithm can be time-consuming, and the microprocessor in an embedded system frequently fails to analyze input binary images efficiently. As a result, hardware architectures tailored to the CCL algorithm have received significant attention [11]. There are three kinds of labeling algorithms for computer architectures and pixel representation images: 1) Raster scanning and label equivalence resolution methods; 2) search and label propagation algorithms; and 3) contour tracing and label propagation algorithms. Raster scanning and label equivalence resolution techniques scan an image in a raster direction (line by line, beginning in the upper-left corner and moving downward) and give a temporary label to each new pixel that is not linked to the previously scanned pixels. Temporary labels assigned to the same linked component are referred to as equivalent labels. There are three approaches for resolving label equivalence [12]. These methods can be classified into three groups based on the number of scans of the image frame in CCL algorithms [13]. Single-scan algorithms execute labeling and label merging on all picture pixels at the same time. Studies [14-18] are instances of this strategy. The two-scan approach stores the label equivalences discovered during the first scan in a 1-D or 2-D table array. Following the initial scanning, label equivalences are resolved using a search technique. The parsed results are typically kept in a 1D table. Then, during the second scan, the temporary labels are replaced with the smallest equivalent label from the table [12]. Studies [12], [19-21] provide instances of two scan approaches. Multiscan approach scans the image in both forward and reverse directions, propagating label equivalents until no labels change. The geometric complexity of the connecting components determines how many scans are

performed [12].

Some selected studies on the use of the CCL algorithm are summarized below. Selçuk et al. studied how different enzymes and additives affected bread production quality. They used the CCL algorithm to find pores in the bread [22]. Grana, et al. in their study, they compared the two most advanced CCL algorithms and evaluated their performance on an FPGA-based SoC architecture [23]. Pandey et al. proposed an FPGA platform-based strategy to implementing an efficient and improved two-scan equivalence-based linked component labeling algorithm [24]. Zhao et al. They suggested a single-scan, resource-efficient parallel CCL algorithm [25].

In this study, the CCL algorithm was used in the FPGA structure to recognize objects in the image. CCL typically utilizes an 8-element neighborhood; however, a 4-element neighborhood can also be employed. Labeling in the application was done using a single scan of an 8-element neighborhood. The CCL algorithm was originally tested using the Vivado program simulation environment. For this aim, an image file was read and given to the CCL algorithm line by line, the blob values discovered were read and painted on the image file, and the CCL algorithm's performance was evaluated. Since the constructed system was intended to be used for blob identification in video pictures in future investigations, video control signals were added to the read image frame and sent to the CCL algorithm. Then it was tested on the ZYNQ SoC.

MATERIALS AND METHODS

Zynq System on Chip

The FPGA development environment consisted of a Zedboard development board with the Zynq-7000 architecture. Figure 1 depicts the appearance of the Zedboard development board. The Z-7020 family chip is used on the Zedboard development board [26]. The Zynq architecture is known as System on Chip (SoC). A SoC structure is one that integrates various hardware components required for an application onto a single chip. These structures may include programmable logic structures, processors, microcontrollers, and digital signal processing units, as needed. These structures allow a complicated set of system functions to be executed on a single chip.

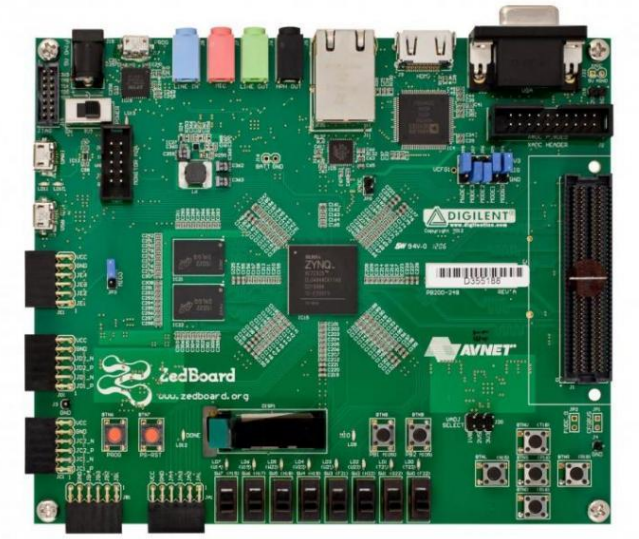

Figure 1
Zedboard development board

It has a fully programmable structure that includes an FPGA and a dual-core ARM Cortex-

A9 architecture CPU on the Zynq-7020 chip. AXI (Advanced eXtensible Interface) buses are used to transmit data between the FPGA and the ARM processor. AXI is a bus and control interface specifically intended for high-performance systems. AXI4 has several versions, including AXI4-Lite and AXI4-Stream, each designed to meet a specific requirement. The AXI interface offers a large bandwidth for high data transmission, allowing for high performance and rapid data connection between the FPGA and the processor. This ensures that these two parts collaborate in order to ensure that a complicated system runs smoothly. The programmable logic unit of the Zynq-7020 chip is referred to as PL (Programmable Logic), whereas the processor unit is known as PS (Processing System) [27]. Figure 2 shows a simplified representation of the ZYNQ 7000 architecture.

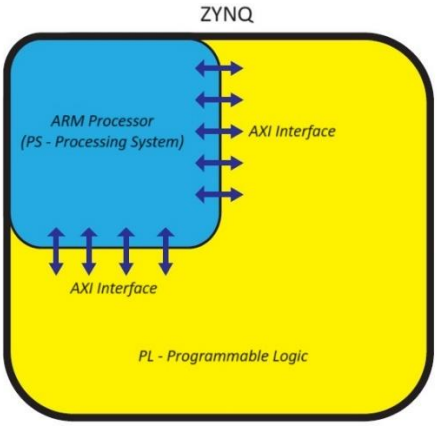

Figure 2
Zynq-7000 simplified architecture

Connected component labeling

The CCL method attempts to recognize objects in images by assigning the same label to adjacent pixels in black-and-white images. The algorithm's primary goal is to identify various items or sections of the image and assign them unique labels. The CCL algorithm can employ either four or eight neighborhoods. CCL algorithms are often designed as single-scan, double-scan, or multi-scan. These scan types specify how pixels in a picture are processed and related components are identified. The single-scan CCL algorithm analyzes the image in a single scan and discovers connected components after passing through it once. Single-scan algorithms are generally simple and fast, but they might be difficult to label precisely when coupled components are too close together. Two-scan algorithms label images using two independent scans. The first scan detects related components, and the second scan performs additional analysis and accurately labels connected components. Although it produces more accurate findings, it may incur a greater computational load. Multi-scan CCL techniques are used to scan more than two image frames. Multi-scan algorithms are employed in situations that need handling of exceptional instances or the reduction of greater detail. This approach may achieve great accuracy in complicated image frames, however each scan may result in increased computational load and delay. Which scan type to utilize in the CCL algorithm is governed by the application's requirements and the hardware resources to be used in the calculation.

The CCL algorithm labels pixels with a value of 1 on binary pictures. A pixel can get a label value of top-left, top-right, top-right, or one of the label values of the pixel to its left. If none of these pixels have a label, the label counter is increased and a new label is applied to the current pixel. Figure 3 illustrates the labeling process for the current pixel.

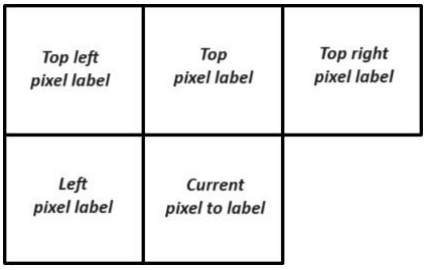

Figure 3
Labeling process of the current pixel

While labeling, you may come across more than one pixel with a different label than the adjacent pixels. Although the connected components labeled 1 and 2 appear to be two different components, when it comes to the current pixel to be labeled, the upper left neighbor pixel has label 1, and the upper right neighbor pixel has label 2. While the current pixel is being labeled, the algorithm determines that labels 1 and 2 need to be merged and marks them for correction. Components with labels 1 and 2 are actually part of the same blob, and after correction, all pixels with label 2 have label value 1. This ensures that all connected components are properly labeled in a single scan.

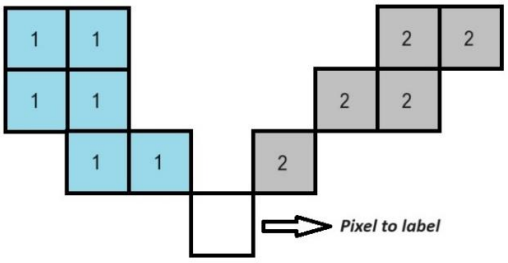

Figure 4
The process of combining labels

Simulation and Testing

The CCL algorithm was tested using a testbench file created using the Vivado program. The appropriate signals were generated by reading the image file that would be tested with this file. Because image processing is typically performed on a pixel, line, or neighboring pixels rather than a whole image frame, and to make better use of resources, the lines of the image file received by another module were taken one at a time and supplied to the CCL algorithm with video control signals. Figure 5 depicts the simulation screen.

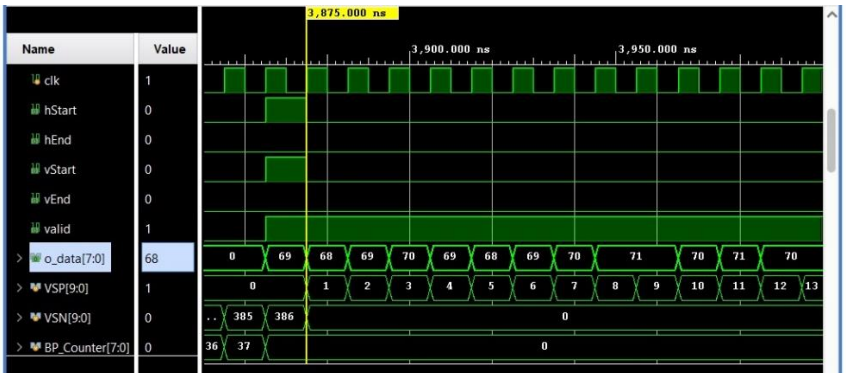

Figure 5
Realized simulation image

Video images are made up of two signals: data and control signals. Pixel values are supplied serially via data signals, while the position of these pixels in the image is defined by control signals. The video frame is sent from left to right and top to bottom. Gap signals are used to modify the timing of visual frames. The horizontal space range generated between the end of one image line and the beginning of the next line is known as the back porch and the front porch. The vertical gap signal [28] refers to the space between the image square's final active line and the beginning of the next active line. Figure 6 depicts the framework utilized to transport video images.

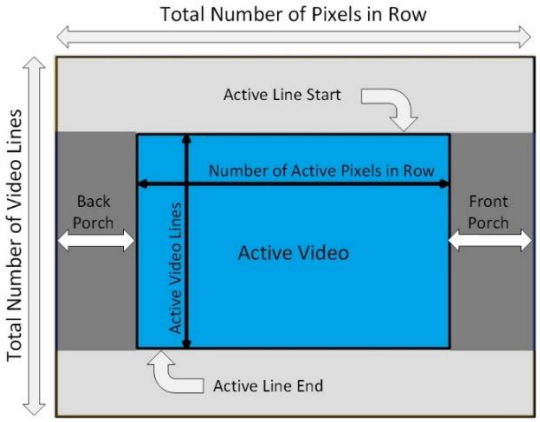

Figure 6
Transmission structure of the video frame

Figure 7 depicts an example of sending a 2x3-pixel image frame with video control signals. The zero-value pixels surrounding the image frame generate horizontal and vertical gaps. In the example, one line space precedes the first active line and one line space follows the last active line. Similarly, there is one pixel of backspace and two pixels of front space. Video control signals are made up of five signals: hStart, hEnd, vStart, vEnd, and valid. With these control signals, the position of each pixel within the image frame is determined.

0	0	0	0	0	0
0	45	60	75	0	0
0	100	120	140	0	0
0	0	0	0	0	0

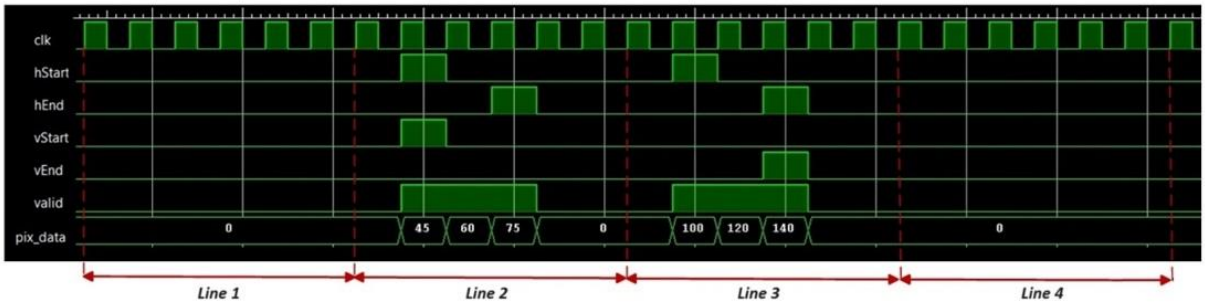

Figure 7
Transmission signals of an example video frame

RESULTS

The CCL algorithm was tested by creating several image files. With the written testbench, image files were read and passed to the CCL algorithm, which read the frame and center of gravity data that constituted the borders of the connected components and drawn on the corresponding image file. Figure 8 depicts an example image frame with the boundaries of the related components discovered as a consequence of the simulation marked in green and the center of gravity noted in red. As seen in the image, the CCL algorithm successfully identified all of the items in the image frame.

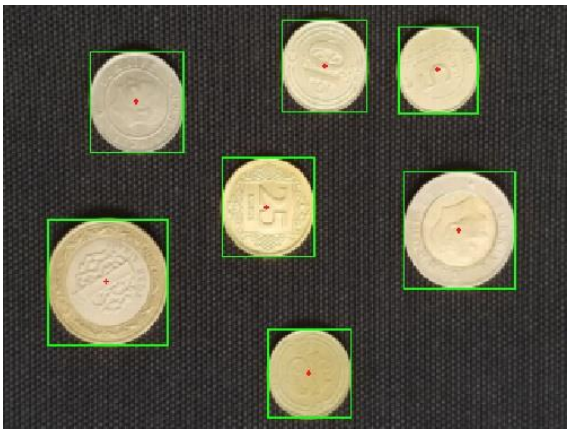

Figure 8
Finding connected components in a sample image frame

To ensure that the CCL algorithm successfully classifies items in a single scan, an image frame identical to Figure 4 was used. The simulation was used to conduct the test, which involved merging the three coins in the bottom left corner of Figure 9. There is no pixel connecting the coins in the upper left and upper right before reaching the image pixels of the coin in the lower middle, hence these two coins are labeled differently. However, as the pixels on the bottom coin began to be labeled, the labels were updated, and the same label was assigned to all three coins as the pixels on the top coins were merged. Drawing the boundaries of the components in Figure 9 showed that the related components were accurately identified. Figure 10 depicts a binary (black-white) image demonstrating that the labeling was correct.

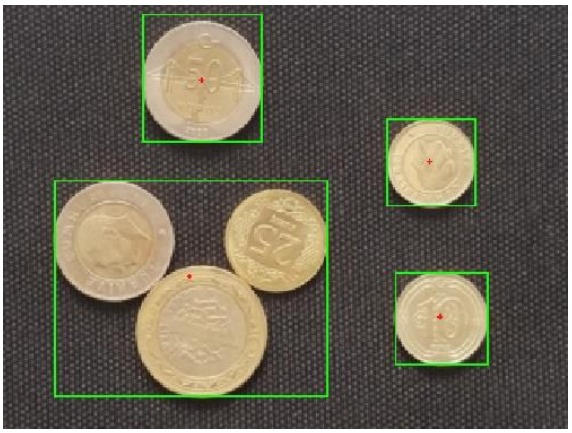

Figure 9
Example of correct labeling of connected components

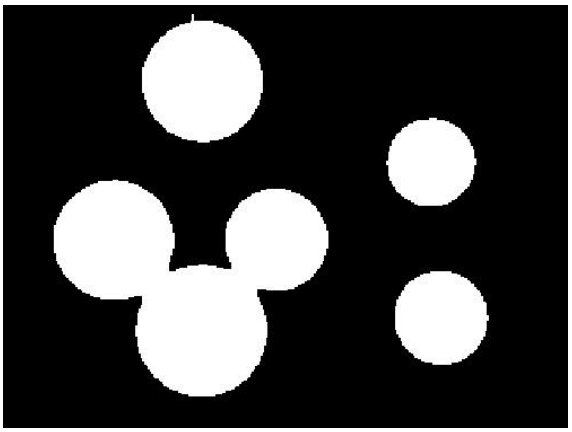

Figure 10
Binary (black-white) image of found connected components

DISCUSSION AND CONCLUSIONS

In this study, the single-scan CCL algorithm was developed, implemented to the FPGA structure, and tested in a simulation environment. The prepared images were read and delivered as lines to the CCL algorithm, which then read and drew the values of the related components. Data for connected components includes a frame and a center of gravity, which indicate the component's limits. Its performance was tested by running the CCL algorithm on various image files. The simulation results showed that the BBE algorithm correctly identified and labeled all components in the images. At a clock frequency of 100 MHz, all connected components in a 240x320 pixel image are detected in around 980 microseconds. Future research will focus on testing the CCL algorithm on real-time video images utilizing the Zynq SoC.

Ethical Statement

This article is the authors' own original work, which has not been previously published elsewhere. The article reflects the authors' own research and analysis in a truthful and complete manner. The results are appropriately placed in the context of prior and existing research.

Author Contributions

Research Design (CRediT 1) İ.Y. (%60) - T.Ü. (%40)
Data Collection (CRediT 2) İ.Y. (%70) - T.Ü. (%30)
Research - Data Analysis - Validation (CRediT 3-4-6-11) İ.Y. (%60) - T.Ü. (%40)
Writing the Article (CRediT 12-13) İ.Y. (%60) - T.Ü. (%40)
Revision and Improvement of the Text (CRediT 14) İ.Y. (%50) - T.Ü. (%50)

Financing

This research was not supported by any public, commercial, or non-profit organization.

Conflict of Interest

The authors have no conflict of interest to disclose for this study.

REFERENCES

[1] M. Kowalczyk, T. Kryjak, A Connected Component Labelling algorithm for a multi-pixel per clock cycle video stream, içinde: *24th Euromicro Conference on Digital System Design (DSD)*, Palermo, Italy, 2021, 43-50.

[2] A. Ünlü, İ. İlhan, A novel hybrid gray wolf optimization algorithm with harmony search to solve multi-level image thresholding problem, *Necmettin Erbakan University Journal of Science and Engineering.* 5(2) (2023), 230- 245. doi:10.47112/neufmbd.2023.21

[3] N.G. Şengöz, F. Zeybek, Sharp Silhouettes for Obtaining 3D Body Measurements from 2D Images, *Necmettin Erbakan University Journal of Science and Engineering.* 4(2) (2022), 8-25.

[4] F. Özen, R. Ortaç Kabaoğlu, T.V. Mumcu, Deep learning based temperature and humidity prediction, *Necmettin Erbakan University Journal of Science and Engineering*, 5(2) (2023), 219-229. doi:10.47112/neufmbd.2023.20

[5] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, Y. Chao, The connected-component labeling problem: A review of state-of-the-art algorithms, *Pattern Recognition.* 70 (2017), 25-43. doi:10.1016/j.patcog.2017.04.018

[6] B. Aissou, A. Aissa, An adapted connected component labeling for clustering non-planar objects from airborne lidar point cloud, *Volume XLIII-B2-2020: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS)*, 2020, pp:191–195.

[7] D. Jaiswal, P. Kumar, Real-time implementation of moving object detection in UAV videos using GPUs, *Journal of Real-Time Image Process.* 17 (2019), 1301–1317, doi:10.1007/s11554-019-00888-5

[8] A. A. Kalinin, V. I. Iglovikov, A. Rakhlin, A. A. Shvets, Medical Image Segmentation Using Deep Neural Networks with Pre-trained Encoders. In: M. Wani, M. Kantardzic, M. Sayed-Mouchaweh, (eds) *Deep Learning Applications. Advances in Intelligent Systems and Computing*, *Springer Singapore*, 2020: pp. 39–52 doi: 10.1007/978-981-15-1816-4_3

[9] V. S. N. Virothi, M. S. Janapareddy, Signature Extraction Using Connected Component Labeling. In: S.C. Satapathy, V. Bhateja, M. Ramakrishna Murty, N. Gia Nhu, Jayasri Kotti (eds) *Communication Software and Networks. Lecture Notes in Networks and Systems, Springer, Singapore*, 2021: pp. 619-629 doi: 10.1007/978-981-15-5397-4_62.

[10] C. Zhao, G. Duan, N. Zheng, A Hardware-Efficient method for extracting statistic ınformation of connected component, *Journal of Signal Processing Systems*. 88 (2016), 55–65. doi:10.1007/s11265-016-1126-5

[11] C. Zhao, W. Gao, F. Nie, A memory-efficient hardware architecture for connected component labeling in embedded system, *IEEE Transactions on Circuits and Systems for Video Technology*. 30 (2019), 3238–3252. doi:10.1109/tcsvt.2019.2937189

[12] L. He, Y. Chao, K. Suzuki, A run-based two-scan labeling algorithm, *IEEE Transactions on Image Processing.* 17(5) (2008), 749-756. doi: 10.1109/TIP.2008.919369

[13] H. Zhou, R. Dou, L. Cheng, J. Liu, N. Wu, A provisional labels-reduced, real-time connected component labeling algorithm for edge hardware, *IEEE Transactions on Circuits and Systems II: Express Briefs.* 69(6) (2022), 2997-3001. doi: 10.1109/TCSII.2022.3152783

[14] A. AbuBaker, R. Qahwaji, S. Ipson and M. Saleh, One Scan Connected Component Labeling Technique, *IEEE International Conference on Signal Processing and Communications*, Dubai, United Arab Emirates, 2007, 1283-1286

[15] J.-W. Jeong, G.-B. Lee, M.-J. Lee, J.-G. Kim, A single-pass connected component labeler without label merging period, *Journal of Signal Processing Systems*. 84 (2015), 211–223. doi:10.1007/s11265-015-1048-7

[16] C. T. Johnston, D. G. Bailey, FPGA implementation of a Single Pass Connected Components Algorithm, *4th IEEE International Symposium on Electronic Design, Test and Applications,* Hong Kong, China, 2008, 228-231

[17] N. Ma, D. G. Bailey and C. T. Johnston, Optimised single pass connected components analysis, *International Conference on Field-Programmable Technology*, Taipei, Taiwan, 2008, 185-192.

[18] J. Trein, A. Th. Schwarzbacher, and B. Hoppe, FPGA implementation of a single pass real-time blob analysis using run length encoding, *MPC-Workshop,* Ravensburg-Weingarten, Germany, (2008).

[19] K. Wu, E. Otoo, K. Suzuki, Optimizing two-pass connected-component labeling algorithms, *Pattern Analysis and Applications*. 12 (2008) 117–135. doi:10.1007/s10044-008-0109-y.

[20] L. He, Y. Chao, K. Suzuki, K. Wu, Fast connected-component labeling, *Pattern Recognition.* 42(9) (2009), 1977-1987. doi:10.1016/j.patcog.2008.10.013

[21] L. He, X. Zhao, Y. Chao, K. Suzuki, Configuration-transition-based connected-component labeling, *IEEE Transactions on Image Processing*. 23(2) (2014), 943-951. doi:10.1109/TIP.2013.2289968

[22] A. Alkan, T. Selcuk, A. S. Çolakoğlu, Görüntü işleme teknikleri kullanılarak ekmek doku analizi ve arayüz programının oluşturulması, *Journal of the Faculty of Engineering and Architecture of Gazi University*. 33(1) (2018), 31-41. doi: 10.17341/gazimmfd.406777

[23] C. Grana, D. Borghesani, P. Santinelli, R. Cucchiara, High Performance Connected Components Labeling on FPGA, *Workshops on Database and Expert Systems Applications,* Bilbao, Spain, 2010, 221-225.

[24] J. G. Pandey, A. Karmakar, A. K. Mishra, C. Shekhar and S. Gurunarayanan, Implementation of an İmproved Connected Component Labeling Algorithm Using FPGA-based Platform, *International Conference on Signal Processing and Communications (SPCOM),* Bangalore, India, 2014, 1-6.

[25] C. Zhao, W. Gao and F. Nie, A resource-efficient parallel connected component labeling algorithm and its hardware implementation, *IEEE Transactions on Multimedia*. 23 (2021), 4184-4197. doi: 10.1109/TMM.2020.3037511.

[26] Avnet Inc., Zedboard Hardware User's Guide, (2014), https://www.avnet.com/wps/wcm/connect/onesite/922900e3-3d57-4cc7-883f-a8b9fbea0cd0/ZedBoard_HW_UG_v2_2.pdf?MOD=AJPERES&CACHEID=ROOTWORKSPACE.Z18_NA5A1I41L0ICD0ABNDMDDG0000-922900e3-3d57-4cc7-883f-a8b9fbea0cd0-nxyWMFS (Access 10 March 2024).

[27] Xilinx Inc., Zynq-7000 All Programmable SoC Data Sheet: Overview (DS190), (2018), https://docs.amd.com/v/u/en-US/ds190-Zynq-7000-Overview (Access 15 March 2024).

[28] MathWorks, Inc., Streaming Pixel Interface, https://www.mathworks.com/help/visionhdl/ug/streaming-pixel-interface.html (Access 18 March 2024)