Real-Time Currency Recognition on Video using AKAZE Algorithm

by Faisal Dharma Adhinata

Real-Time Currency Recognition on Video using AKAZE Algorithm

Faisal Dharma Adhinata*,1), Rifki Adhitama 2) Alon Jala Tirta Segara3)

1,2,3) *Department of Software Engineering, Faculty of Informatics, Institut Teknologi Telkom Purwokerto*
Jl. D. I. Panjaitan No. 128, Purwokerto, Jawa Tengah, Indonesia, 53147

How to Cite: F. D. Adhinata, R. Adhitama, and A. J. T. Segara, "Currency Recognition on Video using AKAZE Algorithm," *Jurnal Teknologi dan Sistem Komputer*, vol. x, no. x, pp. xx-xx, 202x. doi: 10.14710/jtsiskom.x.x.202x.xx-xx, [Online].

Abstract* – *Currency recognition is one of the essential things since everyone in any country must know money. Therefore, computer vision has been developed to recognize currency. One of the currency recognition uses the SIFT algorithm. The recognition results are very accurate, but the processing takes a considerable amount of time, making it impossible to run for real-time data such as video. AKAZE algorithm has been developed for real-time data processing because the computation time in processing video data frames is speedy. This study proposes a method that is faster than the SIFT algorithm so that the currency recognition system can run in real-time processing. The purpose of this study is to compare the SIFT and AKAZE algorithms related to the real-time video data processing to determine the value of F_1 and its speed. Based on the experimental results, the AKAZE algorithm is resulting F_1 value of 0.97 and the processing speed on each video frame is 0.251 seconds. Then at the same video resolution, the SIFT algorithm is resulting in an F_1 value of 0.65 and a speed of 0.305 seconds to process one frame. These results prove that the AKAZE algorithm is faster and more accurate to process video data.

Keywords *- currency recognition; SIFT algorithm; AKAZE algorithm; real-time video data*

I. Introduction

Object recognition is the process of identifying objects based on the characteristics of an object in a digital image or video. The characteristics of an object are often called features of the object. There is a feature extraction stage in image or video data processing. The human eye can easily recognize an object but the computer requires several features to process, such as the color, size, and shape of an object [1]. Object recognition using computers has developed in everyday life, including the recognition of aircraft and ships [2], recognition of butterflies, ants, cameras, and faces [3], and also currency recognition [4]. One of the objects that the researcher developed is currency objects. The recognition of currency objects is beneficial because everyone knows money. Even, those who are illiterate can recognize the type of money.

Some of the techniques developed in currency recognition are template matching [4] and machine learning [1]. In the template matching technique, the stage that most influences the object recognition result is feature extraction. The feature extraction algorithm greatly determines the accuracy and speed of object matching, especially in video data processing. Video data processing is done by extracting the video into frames. Object recognition is done by extracting the features contained in the object. Two types of features are extracted from the frame or image, namely local feature [5] and global feature [6]. Global features are usually used to detect objects and classify them. Instead, local features are used for object recognition or identification.

Some of the local feature extraction algorithms are SIFT [7], SURF [8], ORB [9], and AKAZE [10]. Research by Jing Xu et al. [4] introduced currency coin recognition using the SIFT algorithm. The research results are very accurate, but the matching takes 0.59 seconds. This speed makes the system unable to run in real-time processing. Furthermore, some researchers have also solved the currency recognition problem by using deep learning techniques [11][12]. The use of deep learning requires the data to be trained in advance for a long time. Therefore, another technique that does not involve training in currency recognition is needed.

Several previous studies used the AKAZE template matching algorithm technique in different case studies. Research by Kuznetsov and Savchenko [5] used the AKAZE algorithm to detect logos of sports teams. The use of the AKAZE algorithm for a matching logo is resulting in a more optimal F_1 value than other feature extraction algorithms. AKAZE algorithm only spends 0.15 seconds to process each video frame [13]. The result of using AKAZE in previous research resulted in the optimal value of F_1 and speed. In this paper, the proposed method is the AKAZE algorithm for real-time currency recognition. In this research, we will use a template matching approach with a suitable method for real-time processes with no training process as in the stages of deep learning. For comparison, we also use the SIFT algorithm to compare the F_1 value and speed with the AKAZE algorithm. In the end, we will discuss the right algorithm for the case of real-time currency recognition on video.

This paper is organized into 4 sections. The first section is an introduction, while section 2 describes the methods used in this study. Next, section 3 contains a discussion of the results and evaluation of this system.

The last section or section 4 contains conclusions and suggestions for further work.

II. RESEARCH METHODS

The currency recognition system starts with the acquisition of video data and the query image. Video data is extracted into frames for the next stage of processing. The large query image is resized so that it can be matched with the currency object contained in the video. Then, both query image and video frames are carried out by feature extraction using the SIFT or AKAZE algorithm. The results of feature extraction are keypoints and descriptors of features.

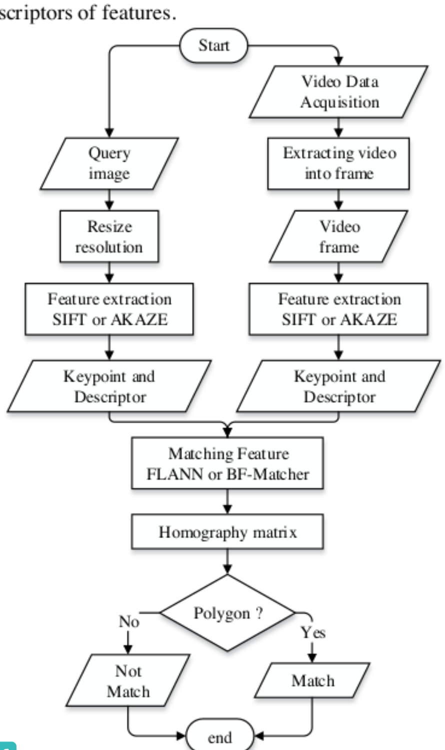

Figure 1. The architecture of the currency recognition system

Keypoints are unique coordinate points as object features, while descriptors are numbers that define keypoints. The next stage is matching the descriptor in the query image and the video frame. The matching features of the SIFT algorithm use the FLANN method, while the AKAZE algorithm uses the Brute-Force Matcher (BF-Matcher) method. The result of feature matching is done by forming a polygon using a homography matrix. If the polygons are formed and the query images with video frames match, this indicates that there is a currency object corresponding to the input query image. The currency recognition system architecture is shown in Figure 1.

A. Data Acquisition

Data acquisition is divided into two parts, namely query image acquisition and video data acquisition. The query image uses Indonesian paper currency with a nominal value of 1000, 2000, 5000, 20000, and 50000 rupiahs. The query image data is shown in Figure 2. The video data acquisition, in this case, uses a 2 MP HiLook camera with Full HD resolution and ten fps. The number of video frames used for the experiment is 600 video frames. For each nominal amount, we use three videos with a distance of 10 cm and 30 cm. Meanwhile, the video recording applied two resolutions, namely Full HD and HD. Table 1 shows the number of videos used in this research. The total number of videos is 20 videos.

Table 1. Details of the data in this research

No	Recorded Distance	Resolution	Video Currency
1	10	Full HD	1000, 2000, 5000, 20000, and 50000
2	10	HD	1000, 2000, 5000, 20000, and 50000
3	30	Full HD	1000, 2000, 5000, 20000, and 50000
4	30	HD	1000, 2000, 5000, 20000, and 50000

Figure 2. Indonesian paper currency for query image input

B. SIFT Algorithm

The SIFT algorithm consists of four stages, namely searching extreme values on scale-space, detecting keypoints, determining orientation, and creating keypoint descriptors [7]. The flowchart of SIFT algorithm is shown in Figure 3. The first stage is constructing a scale-space (octave) using Gaussian blur with equation (1).

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (1)$$

L is a blurred image. Then, G is the Gaussian Blur operator. I is an image where x,y is the location coordinates. σ is the "scale" parameter. Think of it as the amount of blur. More significant the value, the greater the blur. The * is the convolution operation in x and y. It "applies" gaussian blur G onto the image I. The SIFT algorithm on each detection requires 4 octaves and five blur scales.

The second stage is detecting keypoints. Keypoint determination takes a sample point that is compared with 26 pixels neighboring. If the point has the smallest (local minima) or largest (local maxima) value, the point will become a candidate keypoint. Candidate key points that chosen are then filtered to eliminate low-contrast keypoints and keypoints located near the edge. Keypoints are also calculated on magnitude and angle. This stage makes SIFT invariant orientation.

In creating descriptors on the keypoint, SIFT algorithm creates an area of 16x16 pixel size around the keypoint and 4x4 sub-areas with eight orientation directions. The final result is 128 descriptors.

C. AKAZE Algorithm

The AKAZE algorithm consists of 4 parts: computing the contrast factor, constructing non-linear scale-space, detecting features, and creating descriptors [10]. The flowchart of the AKAZE algorithm is shown in Figure 4. The first stage is computing the contrast factor [14]. A Gaussian filter smoothes the query image or frame video. The next step is calculating the maximum absolute gradient value (h_{max}). Index i is looping on the histogram. Afterward, the gradient value is divided by a histogram of 300 bins. Equation (2) is a formula for computing the contrast factor.

$$k = \frac{h_{max}.i}{300} \quad (2)$$

The second stage is constructing a non-linear scale-space [15]. The scale-space approach is as same as the SIFT algorithm, which discretizes the scale-space in logarithmic steps arranged in octaves and scales. The scale-space in the AKAZE algorithm is a pyramid which is shown in Figure 5. It consists of sub-levels that each octave is quarter size than the previous octave.

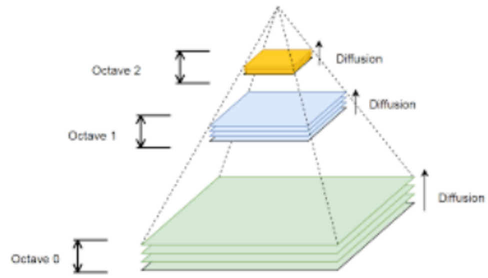

Figure 5. Scale-space representation [16]

The third step is the feature detector. The AKAZE algorithm uses the determinant of Hessian (DoH) blob-detector. After constructing the non-linear scale-space, DoH query image or video frame is computing at sub-levels increase. The keypoints or features in the query image or frame video are extracted by comparing the DoH image with the neighboring window of size 3x3. The pixel point is compared with eight neighbors. If it is greater than eight neighbors, then it becomes a keypoint.

The next step is creating a descriptor [17]. AKAZE algorithm generates a descriptor on each keypoint that scales and rotates invariant. Each key point is made by sampling 16 × 16 pixels around the key point and dividing it into 4 × 4 blocks. The histogram is then calculated by eight bins. The final result is 128 descriptors of the AKAZE algorithm.

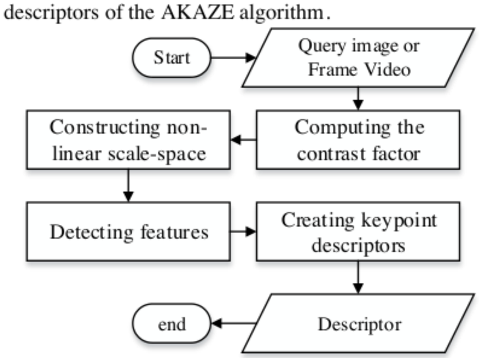

Figure 4. The flowchart of the AKAZE algorithm

D. Matching Feature Query Image with Frame Video

In the matching feature stage, the SIFT algorithm uses the FLANN method while the AKAZE algorithm uses BF-Matcher. The requirement of matching features is at least four keypoints having good matches on query image with frame video. If good matches are more than or equal to four, a Homography matrix search of the query image and frame video is performed [18]. The object on an image will have geometrical transformations such as translation, rotation, scaling, shear. The next stage is checking whether the Homography matrix is formed or not. The process will be terminated if the Homography matrix is not formed, which indicates a mismatch.

1. *FLANN Method*

The Fast Library Approximated Nearest Neighbor (FLANN) method is used to find the nearest neighbor's value [19]. The SIFT algorithm produces 128 dimensions of descriptor for each keypoint. Therefore, matching features with K-NN is considered inefficient so the FLANN method for matching multi-dimensional data is needed. The FLANN method uses the K-Dimensional Tree (KD-Tree) representing multi-dimensional binary tree data to separate certain data areas based on their value position [20].

2. *BF-Matcher Method*

The AKAZE algorithm generates key points and binary descriptors in the query image and frame video. The BF-Matcher work compares each descriptor in the query image with all descriptors on the frame video to find the smallest result [21].

III. RESULT AND DISCUSSION

A. The Experiment of Recording Resolution

Video resolution needs to be tested to see which resolution produces the best F_1 value. Resolution experiments are done by using Full HD (1920x1080) and HD (1280x720) resolutions. In this experiment, the distance between the currency object and the camera is 20 cm. Figures 6 and 7 present the resulting graph of the F_1 value at Full HD and HD resolutions.

In Figure 6, the AKAZE algorithm gives better results than the SIFT algorithm on all currency experiments. Moreover, on currencies 1000 and 20000, the AKAZE algorithm gets the maximum value of F_1. In currencies 2000, 5000, and 50000, many false positive were found which were negative data but were recognized as positive by the system. For example, testing the 50000 currency on the video often matches the query image of 5000. This is because there is the same leading number, which is 5. In the previous study [22], there was also an error in detecting the same nominal using the template matching technique. It is because the numbers 5000 and 50000 only add 0 on the last digit of the nominal.

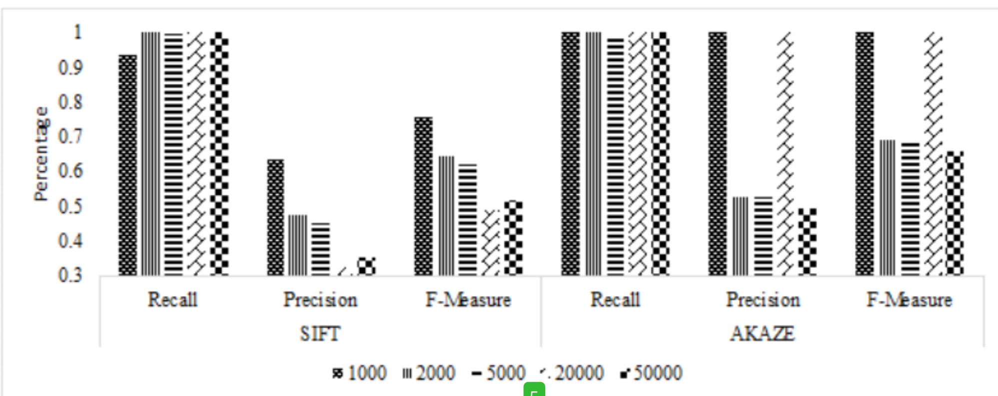

Figure 6. The experiment result of Full HD resolution, a) recall and precision values, b) F-measure values

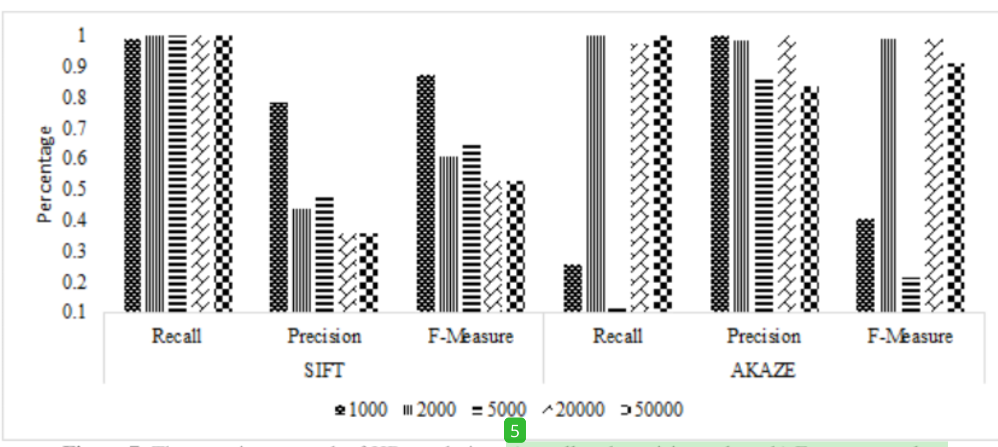

Figure 7. The experiment result of HD resolution, a) recall and precision values, b) F-measure values

The recall-precision curves in Figure 6 show that the SIFT algorithm has a lower precision value than the AKAZE algorithm. This low precision value indicates that the level of accuracy between the information requested by the user and the answer by the system is often wrong. Both Full HD and HD resolutions produce good recall values. In various experiments, it is often found that the recall and precision values are inversely related. If the recall value is high, the precision value is likely low [23].

In the HD resolution experiment, as shown in Figure 7, the average matching result of the AKAZE algorithm is better than the SIFT algorithm. The F_1 value on the 1000 and 5000 currencies is low because the resolution of the recording video is very influential. In previous research [24], video resolution also significantly affects the object detection results and speed. Decreasing the resolution results in fewer features being detected, so that false negatives, which means positive data are recognized as negative by the system, often occur. For example, video data that use the currency of 5000 and is matched against a query image of 5000, the results do not match. However, when compared to the average resolution of Full HD and HD, Full HD resolution produces an average value of F_1 0.81, which is better than HD 0.70 for the AKAZE algorithm. The average SIFT algorithm tends to be the same in Full HD and HD resolutions, namely 0.60 and 0.63. Therefore, in the currency distance experiment with the camera using Full HD resolution.

In previous research[11], the use of deep learning was quite accurate, reaching F_1 of 0.918. Meanwhile, this research uses five currencies, each with a currency of 1700, so that the total training data is 8500 images. This training process is of course takes a long time, which is 48 hours. On the other hand, the research that we propose does not go through a training process and uses five different nominal currencies. The experimental results in this study are also completely accurate, where the F_1 value is 0.81 on the use of Full HD video data.

B. The Experiment of Object Distance

The experiment on the distance of the currency object with a camera aims to determine the optimal matching distance. This experiment uses Full HD resolution considering that in the previous experiment, Full HD resolution resulted in an optimal value of F_1. The distance variation in this experiment uses a distance of 10 cm and 30 cm. Figures 8 and 9 show the effect of a distance of 10 cm and 30 cm on the matching results.

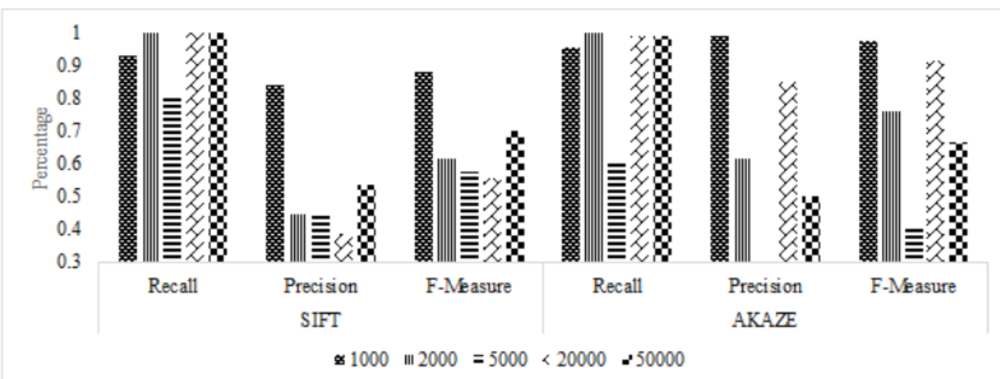

Figure 8. The experiment result of 10 cm distance

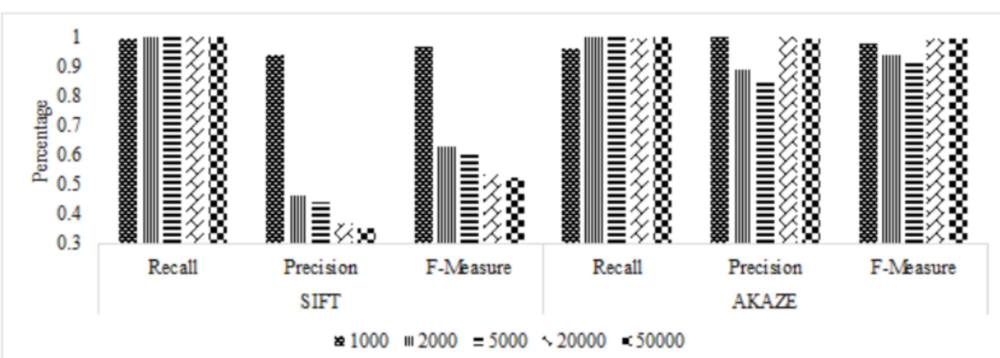

Figure 9. The experiment result of 30 cm distance

Based on Figures 8 and 9, the AKAZE algorithm optimally results in an average F_1 value of 0.97 at 30 cm. However, in the SIFT algorithm, the experimental results showed no significant changes in the 10 cm or 30 cm distance with an average F_1 value of 0.66 and 0.63. At 10 cm, the F_1 value is not optimal because the object is too close to the camera, which causes the feature size to be too large compared to the features in the query image.

The use of this template matching technique dramatically affects the distance. Features that are too large cause the currency object to go undetected. It is because the template matching technique uses feature similarity in the query image. Previous research [25] analyzed the object's distance with the camera which significantly affected accuracy results. Video data that is too large or small makes it incompatible with the query image because its features have a low level of similarity. Therefore, in studies that use distance variations, the optimal results obtained are at a distance of 30 cm with an F_1 value of 0.97.

C. Discussion

Based on the experiment of resolution and distance, the AKAZE algorithm produces a better F_1 value than the SIFT algorithm. In terms of processing speed on each frame, the AKAZE algorithm was found to be faster than the SIFT algorithm as shown in Table 1. This enables the real-time processing of video data. In a previous study [4], the speed of the SIFT algorithm was 0.59 seconds. This speed is also influenced by the computer hardware used.

Table 1. The experiment result of time processing on matching per frame

Resolution	Speed of processing frame (second)	
	SIFT	AKAZE
Full HD	0.305	0.251
HD	0.154	0.113

At Full HD resolution, the AKAZE algorithm takes 0.25 seconds to process a single frame. It is faster than the SIFT algorithm. Processing real-time video data can be done by selecting keyframes, such as processing only a sequence of frames. The data processing speed is highly dependent on the resolution of the video data. The use of full HD data at a distance of 30 cm produces an optimal F_1 value in all currencies, which is more than 0.9, with an average of 0.97. However, the Full HD resolution makes processing only four fps in real-time video data processing. In contrast, HD resolution video data can produce nine fps but produces an F_1 value of 0.7. Overall, between the SIFT and AKAZE algorithms, the processing speed of AKAZE video frames is faster, both at Full HD and HD resolutions.

The weakness of this research is mainly on the value of F_1currency objects which have the same nominal value on the front number. Our future work will modify the recognition of nominal currency numbers to improve accuracy in HD resolution. The results of matching the query image and video data are shown in Figure 10.

IV. Conclusion

This study provides an overview of video data processing, which takes in optimal accuracy and the speed of processing on each frame video. Currency recognition using the AKAZE algorithm results in an F_1 value of 0.97 and a speed of 0.251 at Full HD resolution. Future research can modify the feature extraction method section to make it more accurate when using HD resolution. In this research, experiments at HD resolution resulted in a processing speed of 0.113, but the matching results were not quite accurate.

Figure 10. The result of matching query image and video

References

[1] G. Farooque, A. B. Sargano, I. Shafi, and W. Ali, “Coin Recognition with Reduced Feature Set SIFT Algorithm Using Neural Network,” *Proceedings - 14th International Conference on Frontiers of Information Technology, FIT 2016*, Islamabad-Pakistan, December 2016, pp. 93–98, doi: 10.1109/FIT.2016.025.

[2] B. Jiang, X. Li, L. Yin, W. Yue, and S. Wang,

"Object recognition in remote sensing images using combined deep features," *Proceedings of 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2019*, Chengdu-China, March 2019, pp. 606–610, doi: 10.1109/ITNEC.2019.8729392.

[3] Y. Zhang and J. Liang, "A Vision Based Method for Object Recognition," *Proceedings - 2016 3rd International Conference on Information Science and Control Engineering, ICISCE 2016*, Beijing-China, July 2016, pp. 139–142, doi: 10.1109/ICISCE.2016.40.

[4] J. Xu, G. Yang, Y. Liu, and J. Zhong, "Coin recognition method based on SIFT algorithm," *Proceedings - 2017 4th International Conference on Information Science and Control Engineering, ICISCE 2017*, Changsha-China, July 2017, pp. 229–233, doi: 10.1109/ICISCE.2017.57.

[5] A. Kuznetsov and A. Savchenko, "Sport teams logo detection based on deep local features," *SIBIRCON 2019 - International Multi-Conference on Engineering, Computer and Information Sciences, Proceedings*, Novosibirsk-Russia, October 2019, pp. 548–552, doi: 10.1109/SIBIRCON48586.2019.8958301.

[6] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2017-Octob, no. Nips, pp. 1–15, 2018, doi: 10.1007/978-3-319-23528-8_31.

[7] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[8] H. Bay, A. Ess, T. Tuytelaars, and L. Vangool, "Speeded-Up Robust Features (SURF) (Cited by: 2272)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2006, doi: 10.1016/j.cviu.2007.09.014.

[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *Proceedings of the IEEE International Conference on Computer Vision*, Barcelona-Spain, November 2011, pp. 2564–2571, doi: 10.1109/ICCV.2011.6126544.

[10] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *BMVC 2013 - Electronic Proceedings of the British Machine Vision Conference 2013*, Bristol-UK, September 2013, pp. 1-9, doi: 10.5244/C.27.13.

[11] M. L. Meharu and H. S. Worku, "Real-Time Ethiopian Currency Recognition for Visually Disabled Peoples Using Convolutional Neural Network." Research Square, 2020, doi: 10.21203/rs.3.rs-125061/v1.

[12] Q. Zhang, W. Q. Yan, and M. Kankanhalli, "Overview of currency recognition using deep learning," *Journal of Banking and Financial Technology*, vol. 3, no. 1, pp. 59–69, 2019, doi: 10.1007/s42786-018-00007-1.

[13] D. Henry, Y. Yao, R. Fulton, and A. Kyme, "An Optimized Feature Detector for Markerless Motion Tracking in Motion-Compensated Neuroimaging," *2017 IEEE Nuclear Science Symposium and Medical Imaging Conference, NSS/MIC 2017 - Conference Proceedings*, Atlanta-USA, October 2017, pp. 0–3, doi: 10.1109/NSSMIC.2017.8532865.

[14] P. Soleimani, D. W. Capson, and K. F. Li, "Real-time FPGA-based implementation of the AKAZE algorithm with nonlinear scale space generation using image partitioning," *Journal of Real-Time Image Processing*, no. 0123456789, 2021, doi: 10.1007/s11554-021-01089-9.

[15] H. Seong, H. Choi, H. Son, and C. Kim, "Image-based 3D building reconstruction using A-KAZE feature extraction algorithm," *ISARC 2018 - 35th International Symposium on Automation and Robotics in Construction and International AEC/FM Hackathon: The Future of Building Things*, Berlin-Germany, July 2018, doi: 10.22260/isarc2018/0127.

[16] L. Kalms, K. Mohamed, and D. Göhringer, "Accelerated Embedded AKAZE Feature Detection Algorithm on FPGA," *ACM International Conference Proceeding Series*, Bochum-Germany, June 2017, pp. 3–8, doi: 10.1145/3120895.3120898.

[17] B. Soni, V. Anji Reddy, N. B. Muppalaneni, and C. Lalrempuii, "Image forgery detection using AKAZE keypoint feature extraction and trie matching," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 1, pp. 2208–2213, 2019, doi: 10.35940/ijitee.A4784.119119.

[18] D. K. Iakovidis, E. Spyrou, and D. Diamantis, "Efficient homography-based video visualization for wireless capsule endoscopy," *13th IEEE International Conference on BioInformatics and BioEngineering, IEEE BIBE 2013*, Chania-Greece , November 2013, pp. 1–4, doi: 10.1109/BIBE.2013.6701598.

[19] M. Muja and D. Lowe, "FLANN - Fast Library for Approximate Nearest Neighbors User Manual," *Visapp*. pp. 1–28, 2013.

[20] J. Jo, J. Seo, and J. D. Fekete, "PANENE: A Progressive Algorithm for Indexing and Querying Approximate k-Nearest Neighbors," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 2, pp. 1347–1360, 2020, doi: 10.1109/TVCG.2018.2869149.

[21] I. W. A. Suryawibawa, I. K. G. D. Putra, and N. K. A. Wirdiani, "Herbs Recognition Based on Android using OpenCV," *International Journal of Image, Graphics and Signal Processing*, vol. 7, no. 2, pp. 1–7, 2015, doi: 10.5815/ijigsp.2015.02.01.

[22] M. Naharul, H. Najihul, and S. Adinugroho, "Implementasi Metode Template Matching untuk

Mengenali Nilai Angka pada Citra Uang Kertas yang Dipindai," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 2, pp. 1550–1556, 2019.

[23] N. P. Lestari, "Uji Recall and Precision Sistem Temu Kembali Informasi OPAC Perpustakaan ITS Surabaya," *Universitas Airlangga*, vol. 5, no. 3, pp. 45–46, 2016, [Online]. Available: http://journal.unair.ac.id/LN@uji-recall-and-precision-sistem-temu-kembali-informasi-opac-perpustakaan-its-surabaya-article-10825-media-136-category-8.html.

[24] F. D. Adhinata, M. Ikhsan, and W. Wahyono, "People counter on CCTV video using histogram of oriented gradient and Kalman filter methods," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 3, pp. 222–227, 2020, doi: 10.14710/jtsiskom.2020.13660.

[25] F. D. Adhinata, A. Harjoko, and Wahyono, "Object Searching on Video Using ORB Descriptor and Support Vector Machine," in *Advances in Computational Collective Intelligence*, Da Nang-Vietnam, November 2020, pp. 239–251, doi: 10.1007/978-3-030-63119-2_20

Real-Time Currency Recognition on Video using AKAZE Algorithm

ORIGINALITY REPORT

9%	5%	7%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Lester Kalms, Khaled Mohamed, Diana Göhringer. "Accelerated Embedded AKAZE Feature Detection Algorithm on FPGA", Proceedings of the 8th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies, 2017 Publication	2%
2	theonly1.tistory.com Internet Source	2%
3	journals.usm.ac.id Internet Source	1%
4	E. Basafa, M. Sheikholeslami, A. Mirbagheri, F. Farahmand, G.R. Vossoughi. "Design and implementation of series elastic actuators for a haptic laparoscopic device", 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2009 Publication	1%

5	Rim Messaoudi, Faouzi Jaziri, Achraf Mtibaa, Manuel Grand-Brochier et al. "Ontology-Based Approach for Liver Cancer Diagnosis and Treatment", Journal of Digital Imaging, 2018 Publication	1 %
6	Ukrit Watchareeruetai, Akisato Kimura, Robert Cheng Bao, Takahito Kawanishi, Kunio Kashino. "Interest Point Detection Based on Stochastically Derived Stability", IPSJ Transactions on Computer Vision and Applications, 2011 Publication	<1 %
7	sciencepubco.com Internet Source	<1 %
8	repozitorij.etfos.hr Internet Source	<1 %
9	"Advances in Computational Collective Intelligence", Springer Science and Business Media LLC, 2020 Publication	<1 %
10	ijarcet.org Internet Source	<1 %
11	jurnal.ugm.ac.id Internet Source	<1 %
12	silo.pub Internet Source	

<1%

13	Abolghasem Sadeghi-Niaraki, Soo-Mi Choi. "A Survey of Marker-Less Tracking and Registration Techniques for Health & Environmental Applications to Augmented Reality and Ubiquitous Geospatial Information Systems", Sensors, 2020 Publication	<1%
14	aaltodoc.aalto.fi Internet Source	<1%
15	web-tools.uts.edu.au Internet Source	<1%
16	www.mdpi.com Internet Source	<1%
17	"Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB)", Springer Science and Business Media LLC, 2019 Publication	<1%
18	Md. Atikur Rahman, Muhammad Sheikh Sadi. "IoT Enabled Automated Object Recognition for the Visually Impaired", Computer Methods and Programs in Biomedicine Update, 2021 Publication	<1%