

Avoiding Type I Errors in Image Processing with SIFT/BRISK-keypoints on Android Smartphones

Dmytro Zubov^{1,*} and Andrey Kupin^{2,†}

¹ University of Central Asia, 125/1 Toktogul St., Bishkek, 720001, Kyrgyz Republic

² Kryvyi Rih National University, 11 Vitali Matusevich St., Kryvyi Rih, 50027, Ukraine

Abstract

Avoiding false-positive recognition of objects is a topical problem for specific areas, such as detecting traffic signs for visually impaired pedestrians, fire emergency signs inside buildings, and construction safety signs. Existing solutions show that the percentage of incorrectly recognized traffic signs can reach 25 % for smart vehicles. In this study, SIFT/BRISK-keypoints are employed to design the image descriptor. An experiment with ten images of crosswalk traffic signs and 90 other images (including different traffic signs) showed that the false positive rate is zero and the false negative rate equals 50 %. The implementation is based on the Java Android application with the possibility to correct the knowledge base in case of false alarms. Image analysis was performed on smartphones Doogee S96 Pro and Samsung M31 with an execution time of less than one second. The most likely prospect for further development of this study is the design of the set of image descriptors to improve the false negative rate avoiding type I errors at the same time.

Keywords

image processing, type I error, SIFT/BRISK-keypoint, Android smartphones

1. Introduction

A zero false positive rate, i.e., type I error [1, 2], and minimization of false negative rate, i.e., type II error [1, 2], is the complex criterion employed in ad-hoc image processing projects, such as detecting traffic signs for visually impaired pedestrians, fire emergency signs inside buildings, and construction safety signs. Existing solutions show that the percentage of incorrectly recognized traffic signs can reach 25 % for smart vehicles [3]. Up-to-date real-life implementations also demand autonomous and low-energy solutions since Internet connections are often unstable, and the average ChatGPT request consumes about 0.34 watt-hours and about 0.32176 ml of water [4]. The ecological impact depends on the neural network models (NNMs) – complex responses produce more CO₂ emissions than simple responses, and NNMs that provide more accurate responses result in higher emissions [5]. Reasoning-enabled models produce up to 50 times more CO₂ emissions compared to concise response models [5]. Aliya Rysbek, the research software engineer at Google DeepMind UK [6], pointed out at the KIT forum in Bishkek (Kyrgyz Republic) on 29th May 2025 that her team could recently save about 1 % of the energy consumed by some NNMs which is a huge step considering a tremendous number of requests processed by Google datacenters worldwide.

In this study, the autonomous and low-energy software was developed using Java Android mobile application and SIFT/BRISK-keypoints [7] (Scale-Invariant Feature Transform and Binary Robust Invariant Scalable Keypoints), that is the implementation of the edge computing principle [8]. Power efficacy is achieved by executing the performance-optimized code on the continuously running smartphone without transmitting the data wirelessly. The presented approach employs a unique image descriptor for every target object which is different from the previously developed method

ICST-2025: Information Control Systems & Technologies, September 24-26, 2025, Odesa, Ukraine

* Corresponding author.

† These authors contributed equally.

✉ dzubov@ieee.org (D. Zubov); kupin@knu.edu.ua (A. Kupin)

ORCID 0000-0002-5601-7827 (D. Zubov); 0000-0001-7569-1721 (A. Kupin)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[7], where a 291-point pattern is applied. Initially, a multithreaded Java Android application takes a photo via Google's CameraX library [9], and then the method *Bitmap.createScaledBitmap* generates a new bitmap scaled to a maximum resolution of 500 pixels using bilinear filtering. From up to 700 keypoints detected by the SIFT method, several (three in this paper's example) basic keypoints are selected. Next, the BRISK binary descriptor is designed considering keypoints that are unique on the target image, and the distances to basic keypoints are calculated. Experiments conducted on the Doogee S96 Pro and Samsung M31 smartphones demonstrated that the execution time is less than one second, with a false positive rate of zero and a false negative rate of 50 %.

The remaining part of the paper proceeds as follows: Section 2 reviews relevant works in the context of the most cutting-edge computer vision techniques. It also introduces the proposed soft-/hardware architecture. Section 3 outlines the problem setup and the experiment setup from image capturing to image matching. Section 4 presents a successful experiment conducted with 100 images. Results and discussion are presented in Section 5 and Section 6, respectively. Conclusions are summarized in Section 7.

2. Related works

Up-to-date image processing algorithms emphasize accuracy, interpretability, transparency, speed, and scalability while reducing computational costs [10]. Some of the most cutting-edge computer vision techniques are as follows:

1. Convolutional neural networks (CNNs) [11, 12].
2. Vision transformers (ViTs) [13].
3. Segmentation techniques [14].
4. In addition, there are numerous other image processing techniques, such as generative adversarial networks, super-resolution algorithms, adaptive histogram equalization, and denoising algorithms [10].

Two-dimensional CNNs, such as those presented in [11, 12], are prevalent in image processing nowadays. They use convolution to detect patterns in images, and then classify them, detect objects, apply semantic segmentation, etc. Prior to CNN processing, images have preprocessing steps such as homogenization, normalization, and principal component analysis [12]. Basic CNN components are the convolution layer, pooling layer, activation function, batch normalization, dropout, and fully connected layer. The most common CNN models are AlexNet, ResNet, VGG, GoogleNet, Xception, Inception, DenseNet, and EfficientNet [13].

In contrast to CNNs, which depend on hierarchical feature extraction, ViTs analyze images as sequences of smaller patches, enabling them to capture contextual information and long-range dependencies, resulting in improved image recognition capacities [13].

Image segmentation divides an image into distinct regions based on certain characteristics [14, 15]. Techniques like U-Net architectures, Canny edge detection, and Mask R-CNNs provide efficient and precise solutions [10]. In this study, each image is segmented into two regions – the object O and the background B [15]. Following the application of a segmentation algorithm, pixels or other image attributes are classified into either region O or B .

The above-stated image processing techniques face common challenges in real-life implementation [10]:

1. High computational cost: values can differ several times. Solution: computational complexity reduction (the code was performance-optimized in this study).
2. Overfitting diminishes the model's generalization ability, potentially leading to lower accuracy. Solution: data augmentation and regularization (thresholding is used to separate the object and background regions in this study).

3. Noise and distortion can compromise the accuracy of image processing algorithms. Solution: images should be filtered (bilinear and Gaussian filters are employed in this study).
4. Interpretability and transparency of some AI-driven image processing methods. Solution: non-AI driven image processing methods (image processing with SIFT/BRISK-keypoints is employed in this study).
5. Real-time processing constraints. Solution: high-performance soft-/hardware (multicore smartphones were utilized in this study).
6. Ethical and privacy concerns. Solution: autonomous systems (edge computing with a mobile Java Android application was implemented in this study).

The growing demand for machine learning on mobile devices has led to the development of lightweight CNN models, such as MobileNet [11, 16], which are optimized for use with limited computational power and memory. MobileNet V1 employed depthwise separable convolutions, and MobileNet V2 improved upon this with inverted residual blocks, further enhancing efficiency. Later versions of MobileNet optimize performance for mobile CPUs. Data requirements are the key drawback of lightweight CNN models since they require a significant amount of data to be trained to achieve acceptable performance on mobile devices. Some projects, as presented in this study, lack large datasets. This limitation leads to a loss of accuracy and challenges in training and optimization.

In this study, the system requires a false positive rate of zero and a minimized false negative rate. Considering the challenges outlined and the latest image processing techniques, the proposed architecture of the project soft-/hardware is presented in Figure 1. In this prototype, the end-user interacts with the smartphone via the simple user interface based on the button element with an onClick listener [17]. The multithreaded Java Android application processes images captured by the smartphone camera using the CameraX library and SIFT/BRISK-keypoints [7]. To update the knowledge base, the mobile application should have the option to download the updated information from Internet resources, such as GitHub and Firebase. For this purpose, the study proposes the use of JSON data format [17] because of its lightweight nature and widespread use in mobile applications.

3. Methods

3.1. Problem setup

The following notations are used in the following sections of the paper. Considering a grayscale image $I(x, y)$ with n SIFT/BRISK-keypoints [7] with coordinates, i.e., pairs of real numbers (x_n, y_n) , the

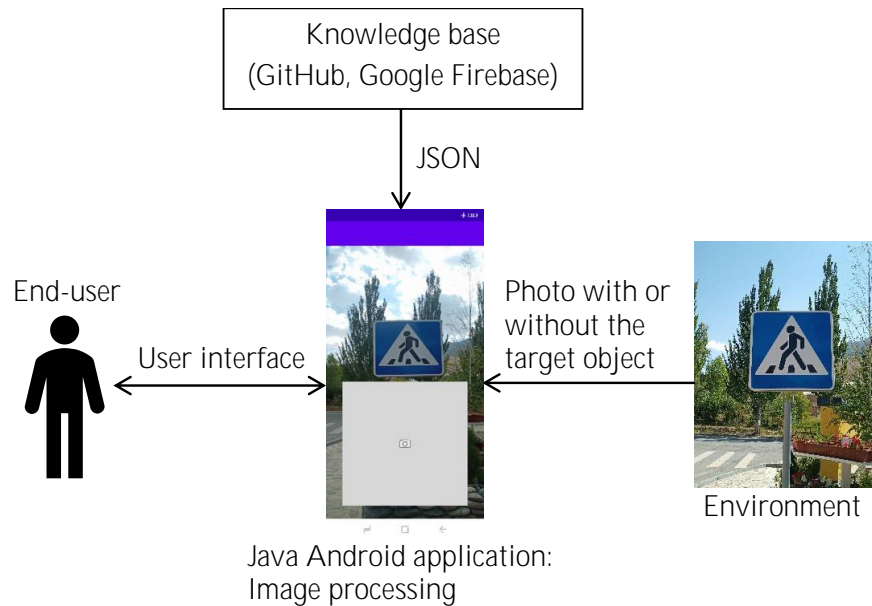


Figure 1: The proposed architecture of soft-/hardware complex.

Euclidean plane R^2 is as follows:

$$R^2 = \{(x, y) | x, y \in R\}. \quad (1)$$

The well-known formula of the Euclidean distance between two arbitrary SIFT/BRISK-keypoints p_1 and p_2 is a function $d: R^2 \times R^2 \rightarrow R$ defined as follows:

$$d((x_{p_1}, y_{p_1}), (x_{p_2}, y_{p_2})) = \sqrt{(x_{p_1} - x_{p_2})^2 + (y_{p_1} - y_{p_2})^2}. \quad (2)$$

The well-known two-point form of a line on a Euclidean plane is as follows:

$$y - y_{p_2} = \frac{y_{p_2} - y_{p_1}}{x_{p_2} - x_{p_1}}(x - x_{p_2}). \quad (3)$$

After applying a segmentation algorithm, n SIFT/BRISK-keypoints are classified into one of the two regions, object O and background B . Similar to [15], a segmentation algorithm A is a function that maps a grayscale image $I(x, y)$ into two sets of SIFT/BRISK-keypoints O and B :

$$\{O, B\} = A(I(x, y)). \quad (4)$$

In this study, algorithm A employs a score Δ that quantifies the difference between the object and background regions. This score is determined based on various conditions c_j ($j \leq m$, where $m = m_1 + m_2$ is the number of conditions in algorithm A ; m_1 is the number of obligatory conditions; m_2 is the number of optional conditions), equations (2) and (3), and pixel values of a grayscale image $I(x, y)$. If $1\{c_j\}$ represents the indicator function, which returns 1 if c_j is true and 0 otherwise, the characteristic m_{ob} of the object region is calculated as follows:

$$m_{ob} = \sum_{j=m_1}^{m-1} 1(c_j) \prod_{j=0}^{m_1-1} 1(c_j), \quad (5)$$

where m_1 conditions in the product operation are obligatory, m_2 conditions in the summation operation are optional.

Similar to m_{ob} (5), the characteristic m_{bg} of the background region is calculated with given conditions c_j as follows:

$$m_{bg} = \sum_{j=m_1}^{m-1} 1(c_j) \prod_{j=0}^{m_1-1} 1(c_j). \quad (6)$$

Then, a score Δ is computed using a threshold value V :

$$\Delta = 1(m_{ob} \geq V). \quad (7)$$

The threshold value V is determined from the training data. Thus, algorithm A indicates the presence ($\Delta=1$) or absence ($\Delta=0$) of the target object in grayscale image $I(x, y)$.

3.2. Experiment setup

In this study, a group named “Crosswalk” of traffic signs “Crosswalk right”, “Crosswalk left”, and “Zebra crossing” (see Figure 2) is identified in the image using the standard representation officially accepted in the Kyrgyz Republic [18]. The “Crosswalk” group is a subset of the traffic signs intended for pedestrians. Avoiding type I errors in this object recognition is a crucial point in the spatial cognition of visually impaired people [7].

The core steps of the above-stated algorithm A are as follows:

1. Image capturing by the smartphone’s camera and CameraX Android API.
2. Downsampling the image with bilinear filtering in a Java Android application.

3. Localization of SIFT/BRISK-keypoints.
4. Selection of SIFT/BRISK-keypoints that have stable positions across different SIFT octaves.
5. Designing the image descriptor based on selected SIFT/BRISK-keypoints and algorithm A (7).
6. Image matching.



Figure 2: A group of traffic signs “Crosswalk”: “Crosswalk left” (A), “Crosswalk right” (B), and “Zebra crossing” (C).

4. Experiment

4.1. Selection of SIFT/BRISK-keypoints with stable positions across different SIFT octaves

The image capturing by the smartphone’s camera and CameraX Android API, downsampling the image with bilinear filtering in Java Android application, and localization of SIFT/BRISK-keypoints are the algorithm steps, which are similar to those described in [7]. An example of the selection of SIFT/BRISK-keypoints on traffic sign “Crosswalk left” with stable position across different SIFT octaves is shown in Figure 3, where the first octave of size 340×340 pixels contains 49 SIFT/BRISK-keypoints, the second octave – 680×680 pixels, 109 SIFT/BRISK-keypoints, and the third octave – 1360×1360 pixels, 124 SIFT/BRISK-keypoints (fuchsia and turquoise colors are used for third/second/first and fourth/third/second DoG (Difference of Gaussians) functions, respectively). The values of the population standard deviations in the Gaussian blur operator are consistent with those presented in [7]. Only 700 keypoints that are closest to the center of the image, based on Euclidean distance (2), are considered.

Analysis of three SIFT octaves shows that 47 SIFT/BRISK-keypoints have stable positions on these SIFT octaves (see Figure 4). Some coordinates of SIFT/BRISK-keypoints are as follows (octave of size 1360×1360 pixels is used; the origin of coordinates is at top left; the axis x is horizontal; the axis y is vertical): $(x_0, y_0)=(283, 1064)$, $(x_1, y_1)=(353, 961)$, $(x_{19}, y_{19})=(116, 1086)$, $(x_{31}, y_{31})=(679, 111)$, $(x_{46}, y_{46})=(1243, 1086)$.



Figure 3: SIFT octaves with SIFT/BRISK-keypoints on the traffic sign “Crosswalk left” (images are not scaled according to the SIFT octaves’ sizes): octave 340×340 pixels (A), octave 680×680 pixels (B), octave 1360×1360 pixels (C).

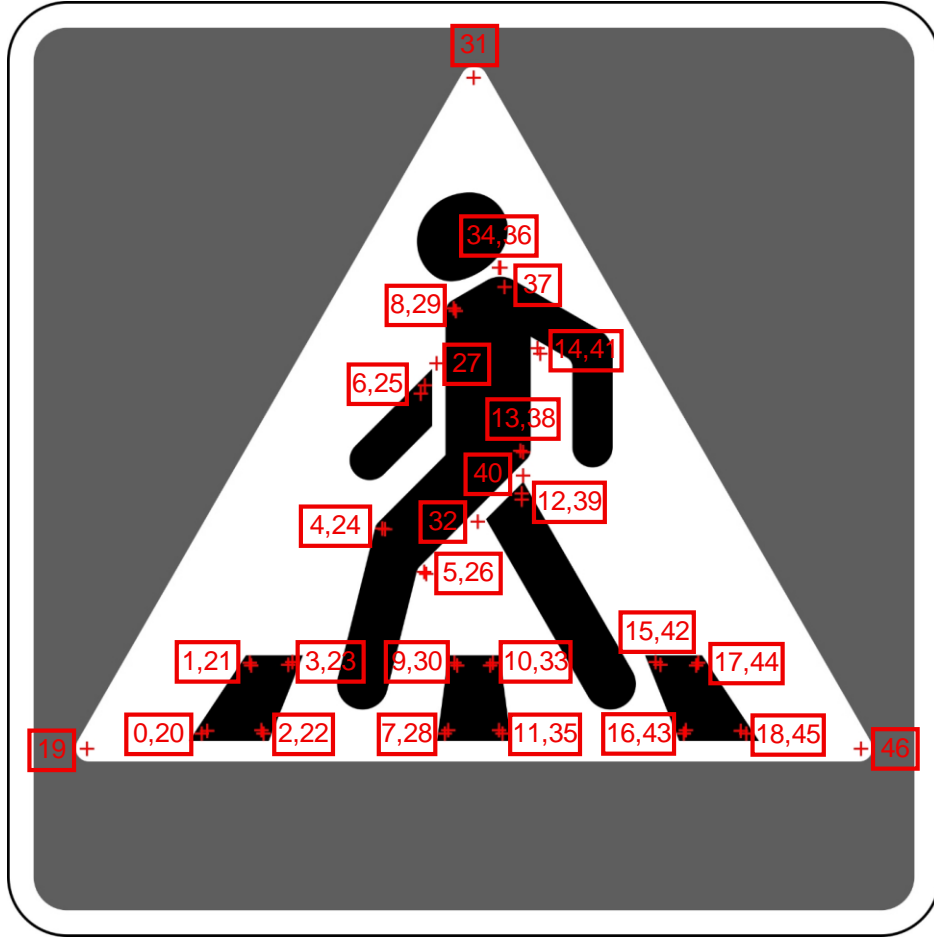


Figure 4: 47 SIFT/BRISK-keypoints on the traffic sign “Crosswalk left” with stable positions across three SIFT octaves.

4.2. Designing the image descriptor based on the selected SIFT/BRISK-keypoints and algorithm A (7)

In this study, the following descriptive visual attributes [19] are employed to design indicator functions $1\{c_j\}$ using a few human-friendly text descriptions:

1. SIFT/BRISK-keypoints 19, 31, and 46 are the basic components that form a triangle with other SIFT/BRISK-keypoints located inside.
2. The grayscale image has the average pixel value denoted as I_{av} .
3. Euclidean distances (2) between SIFT/BRISK-keypoints.
4. Lines (3) connect various SIFT/BRISK-keypoints.

The obligatory conditions c_j ($j < m_1$; $m_1=3$ in this study) were formulated by the human expert as follows:

1. c_0 : the distances between SIFT/BRISK-keypoints 19-31, 19-46, and 31-46 should be greater than 100 pixels and equal to one another, with a relative standard deviation (RSD) of 0.05.
2. c_1 : the pixel values at SIFT/BRISK-keypoints 19, 31, and 46 should be greater than $(I_{av}-20)$, indicating that the intensity must be light.
3. c_2 : the pixel values on the lines (3) connecting SIFT/BRISK-keypoints 19-31, 19-46, and 31-46 should be greater than $(I_{av}-20)$. 5 % is the mistake allowed in c_2 .

The optional conditions c_j ($m_1 \leq j < (m_1+m_2)$; $m_2=44$ in this study) were formulated by a human expert as follows:

1. c_3 : the distances between SIFT/BRISK-keypoints 0-19, 0-31, and 0-46 should match the calculated distances on the template image (see Figure 3) with RSD=0.05. Additionally, the pixel value at the SIFT/BRISK-keypoint 0 should correspond to the relevant SIFT/BRISK-keypoint on the template image (it must be less than $(I_{av}-30)$ in this study).
2. c_4 : the distances between SIFT/BRISK-keypoints 1-19, 1-31, and 1-46 should match the calculated distances on the template image (see Figure 3) with RSD=0.05. Additionally, the pixel value at the SIFT/BRISK-keypoint 1 should correspond to the relevant SIFT/BRISK-keypoint on the template image (it must be less than $(I_{av}-30)$ in this study).
- ...
6. c_8 : the distances between SIFT/BRISK-keypoints 5-19, 5-31, and 5-46 should match the calculated distances on the template image (see Figure 3) with RSD=0.05. Additionally, the pixel value at the SIFT/BRISK-keypoint 5 should correspond to the relevant SIFT/BRISK-keypoint on the template image (it must be greater than $(I_{av}-20)$ in this study).
- ...
44. c_{46} : the distances between SIFT/BRISK-keypoints 45-19, 45-31, and 45-46 should match the calculated distances on the template image (see Figure 3) with RSD=0.05. Additionally, the pixel value at the SIFT/BRISK-keypoint 45 should correspond to the relevant SIFT/BRISK-keypoint on the template image (it must be less than $(I_{av}-30)$ in this study).

The conditions c_0 - c_{46} were designed to specifically target the features of crosswalk signs (see Figure 2).

5. Results

In this study, a Java Android mobile application implements the proposed image processing method to detect a specific group of Kyrgyz traffic signs “Crosswalk”. An experiment with ten images of crosswalk traffic signs and 90 other images (including different traffic signs) showed a false positive rate of zero and a false negative rate of 50 %. Image analysis was performed on smartphones Doogee S96 Pro and Samsung M31 with an execution time of less than one second. All original color pictures/photos and their grayscale versions with keypoints generated by SIFT algorithm were uploaded to the Google Drive folder “https://drive.google.com/drive/folders/15Dk27s8_2mIZnBsLNcq8j11WJbeFAFVP”. All photos were taken by co-author Dr. Dmytro Zubov. Figure 5 shows nine examples of pictures and photos with average pixel values used in the experiment. Figure 6 presents the grayscale images with SIFT/BRISK-keypoints for the initial data presented in Figure 5.

The analysis of the experimental results shows that the target image can be identified using two distinct rules:

1. Rule 1 (for pictures of traffic signs [18]): A threshold value of V equals 20 (the closest value of m_{ob} equals 36 for the traffic sign “Crosswalk right”; the closest value of m_{bg} equals 14 for the traffic sign “Bike crossing” uploaded in subfolder 12 on Google Drive).
2. Rule 2 (for photos taken by the smartphone): A threshold value of V equals 9 (the closest value of m_{ob} equals 9 for the traffic sign “Zebra crossing” presented in Figure 5(G); the closest value of m_{bg} equals 6 for the photo uploaded in subfolder 91 on Google Drive).

6. Discussion

During the discussion, the project was presented at the Department of Computer Science of the University of Central Asia and the Department of Computer Systems and Networks of the Kryvyi Rih National University, two questions were raised:



Figure 5: Examples of pictures and photos with average pixel values used in the experiment.

1. The segmentation technique was employed to process images. However, it was noted that CNNs are widely used nowadays. An explanation was provided regarding the interpretability and transparency of the applied approach, especially in safety-critical projects where understanding the decision-making process is essential.
2. The smartphone application was developed using Java programming language. Although this is a common choice for Android mobile applications, it was suggested to consider using the Kotlin programming language and the Jetpack Compose declarative paradigm to enhance code readability for other developers.

Additionally, the Gemini generative artificial intelligence chatbot [20] was requested to provide recommendations on what should be improved in the research paper (see Figure 7) on June 28, 2025. The response included several suggestions for improvement, categorized by different sections of the manuscript. Three general recommendations are as follows:

1. Consistency in Terminology: While you use "SIFT/BRISK-keypoints" throughout, sometimes "SIFT keypoints" are mentioned in isolation (e.g., in the experiment section, "Localization of SIFT/BRISK-keypoints" is followed by "Selection of SIFT/BRISK-keypoints"). Ensure consistency or clarify if there's a specific reason for the distinction at certain points.
2. Flow and Transitions: Some transitions between paragraphs and sections could be smoother. Consider using more explicit linking phrases or sentences to guide the reader.
3. Conciseness: While the paper is generally well-structured, some sentences or phrases could be more concise. For example, instead of "The most likely prospect for further development of this study is the design of the set of image descriptors to improve the false negative rate

avoiding type I errors at the same time", you could simplify it to "Future work will focus on designing a set of image descriptors to improve the false negative rate while maintaining zero Type I errors."

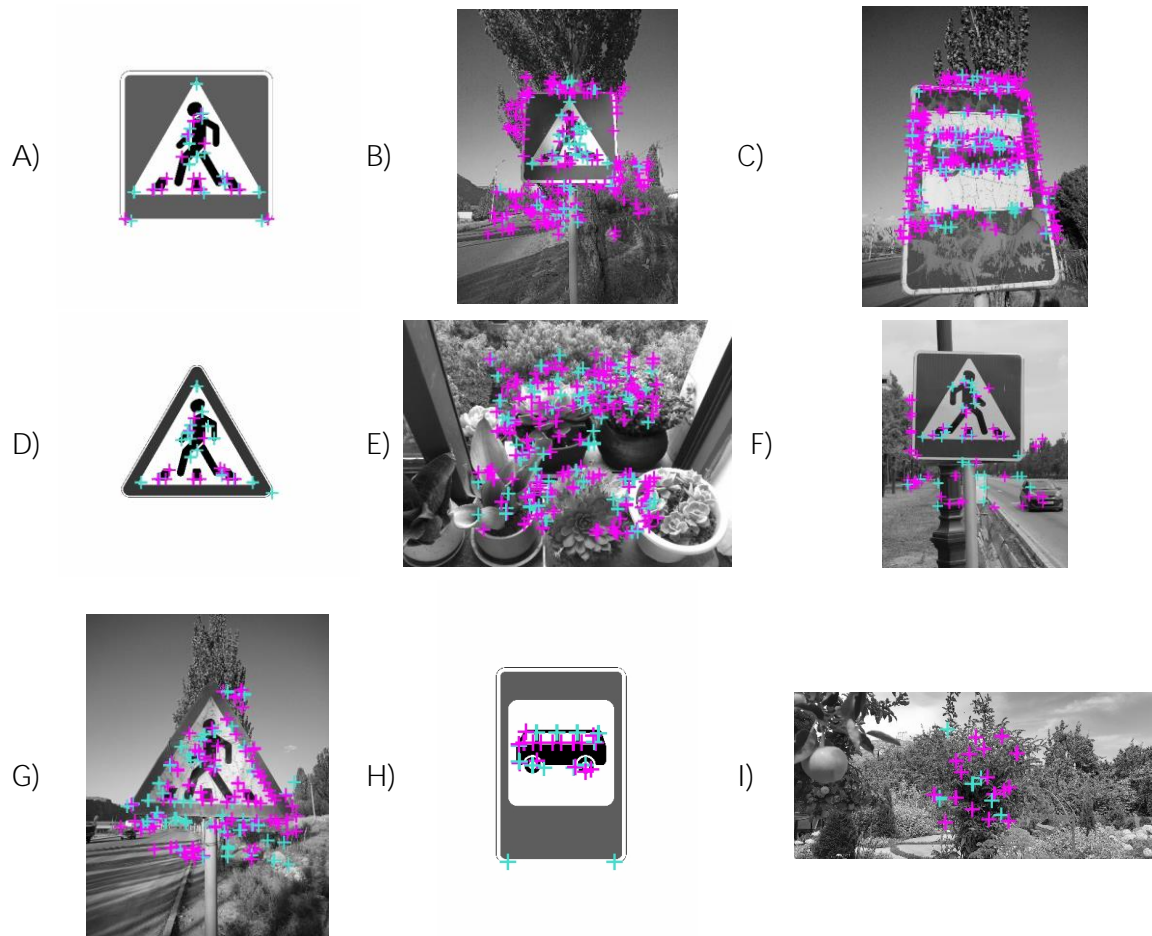


Figure 6: Grayscale images with SIFT/BRISK-keypoints for the initial data presented in Figure 5.

Some of the above-stated recommendations, such as paraphrasing sentences in conclusions, have already been considered. Other suggestions are discussed in references or are not critical in the presented study, and hence they can be taken into account in future work.

7. Conclusions

This study presents a new method of image processing with SIFT/BRISK-keypoints and descriptive visual attributes, implemented in the developed prototype of Java Android mobile application. The interpretability, transparency, and zero false positive rate of the applied approach are the key advantages.

The core steps of the image processing algorithm are as follows:

1. Image capturing by the smartphone's camera and CameraX Android API.
2. Downsampling the image with bilinear filtering.
3. Localization of SIFT/BRISK-keypoints.
4. Selection of SIFT/BRISK-keypoints with stable positions across different SIFT octaves.
5. Designing the image descriptor based on the selected SIFT/BRISK-keypoints and descriptive visual attributes.
6. Image matching.

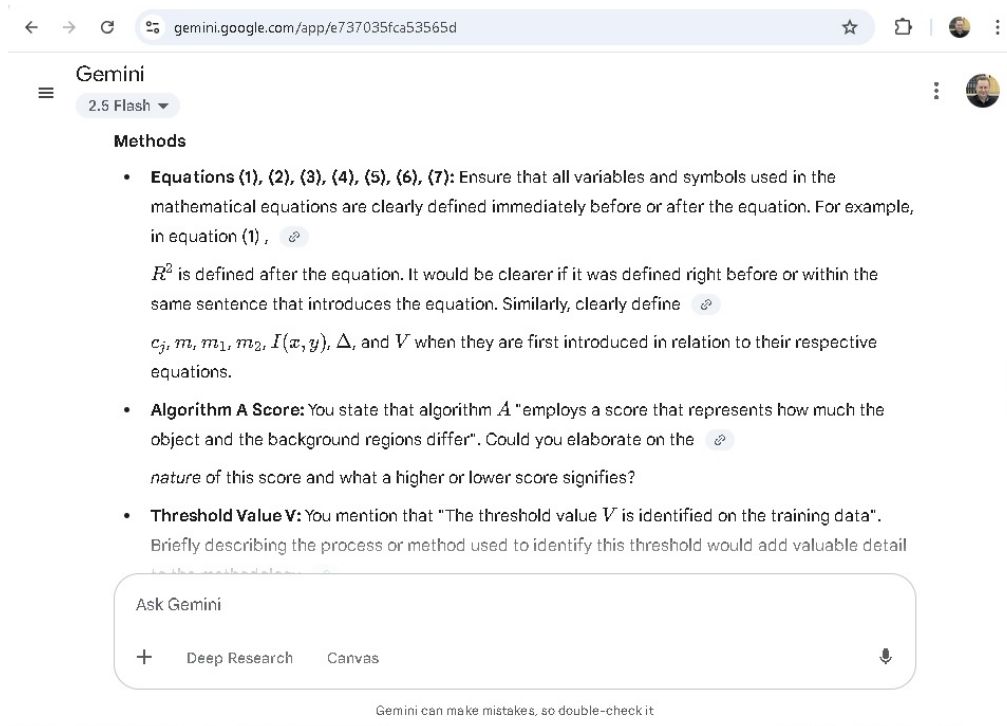


Figure 7: The screenshot of the response (a part) from the Gemini generative artificial intelligence chatbot regarding recommendations for improving the research paper.

Analysis of the experimental results indicates that the target image, i.e., the group of traffic signs “Crosswalk”, can be identified on the photo using a distinct rule: a threshold value V should be equal to 9 in the proposed classification algorithm. The experiment of the image analysis on smartphones Doogee S96 Pro and Samsung M31 showed that the execution time is less than a second.

Future work will focus on designing a set of image descriptors to improve the false negative rate while maintaining zero Type I errors.

Acknowledgements

This work and the research behind it received support from the universities where the authors conducted the study. The authors express their sincere gratitude to colleagues at the University of Central Asia and Kryvyi Rih National University who contributed to this project.

Declaration on Generative AI

In preparing this work, the authors employed the Grammarly writing assistant [21] for grammar and spelling errors, as well as the Gemini generative AI chatbot to discuss the results of the study. Following the use of these tools, the authors reviewed and edited the content. The authors take full responsibility for the content of this publication.

References

- [1] J. Shreffler; M. R. Huecker, Type I and Type II Errors and Statistical Power, 2023. URL: <https://www.ncbi.nlm.nih.gov/books/NBK557530/>.
- [2] M. D. Lieberman, W. A. Cunningham, Type I and Type II Error Concerns in fMRI Research: Re-balancing the Scale. *Social Cognitive and Affective Neuroscience* 4.4 (2009) 423-428. doi:10.1093/scan/nsp052.
- [3] D. Babić, D. Babić, M. Fiolić, Ž. Šarić, Analysis of Market-Ready Traffic Sign Recognition Systems in Cars: A Test Field Study, *Energies* 14.12 (2021) 3697. doi:10.3390/en14123697.

- [4] S. Altman, The Gentle Singularity, 2025. URL: <https://blog.samaltman.com/the-gentle-singularity>.
- [5] D. Pirchner, Some of Your AI Prompts Could Cause 50 Times More CO2 Emissions Than Others, 2025. URL: <https://www.frontiersin.org/news/2025/06/19/ai-prompts-50-times-more-co2-emissions>.
- [6] I. Jurenka, M. Kunesch, K. R. McKee, et al., Towards Responsible Development of Generative AI for Education: An Evaluation-Driven Approach, 2024. URL: <https://arxiv.org/abs/2407.12687>. doi:10.48550/arXiv.2407.12687.
- [7] A. Aljarbouh, D. Zubov, A. Kupin, N. Shaidullaev, Traffic-sign Recognition for Visually Impaired Pedestrians in Kyrgyzstan: Two-keypoint SIFT/BRISK Descriptor with CameraX, in: Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Systems: Intelligent Systems Workshop, Lviv Ukraine, 2024, pp. 145-156. URL: <https://ceur-ws.org/Vol-3688/paper11.pdf>.
- [8] H. Yuan, Z. Niu, J. Liu, Optimization Techniques for Real-time Optical Image Processing Using Edge Computing, in: SPIE Proceedings of Second International Conference on Big Data, Computational Intelligence, and Applications, volume 13550, Huanggang China, 2025, pp. 1-8. doi:10.1117/12.3059702.
- [9] R. Iyengar, Scaling Up Wearable Cognitive Assistance for Assembly Tasks, PhD's thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2023. UMI order number: CMU-CS-23-112. doi:10.1184/R1/23302121.v1.
- [10] leylinepro.Com, Latest Image Processing Algorithms in 2025, 2025. URL: <https://www.leylinepro.ai/blog/latest-image-processing-algorithms>.
- [11] R. Raj, A. Kos, An Extensive Study of Convolutional Neural Networks: Applications in Computer Vision for Improved Robotics Perceptions, Sensors 25.4 (2025) 1033. doi:10.3390/s25041033.
- [12] X. Zhao, L. Wang, Y. Zhang, et al., A Review of Convolutional Neural Networks in Computer Vision, Artificial Intelligence Review 57.99 (2024) pp. 1-43. doi:10.1007/s10462-024-10721-6.
- [13] S. Jamil, M. J. Piran, O-J. Kwon, A Comprehensive Survey of Transformers for Computer Vision, Drones 7.5 (2023) 287. doi:10.3390/drones7050287.
- [14] D. S. Bhuvaneshwaran, K. Veeraperumal, M. Rajagopal, S. Santharaman, P. Ramasamy, Edge detection in image segmentation, in: Proceedings of 6th International Conference on Intelligent Computing (ICONIC2K23), Chennai, Tamil Nadu, India, 2023, 020061. doi:10.1063/5.0259758.
- [15] K. Tanizaki, N. Hashimoto, Y. Inatsu, H. Hontani, I. Takeuchi, Computing Valid P-Values for Image Segmentation by Selective Inference, in: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 9550-9559. doi:10.1109/CVPR42600.2020.00957.
- [16] M. Rani, M. Kumar, MobileNet for Human Activity Recognition in Smart Surveillance Using Transfer Learning, Neural Computing and Applications 37 (2025) 3907-3924. doi:10.1007/s00521-024-10882-z.
- [17] A.-R. Mawlood-Yunis, Android for Java Programmers, Springer Cham, Cham, Switzerland, 2022. doi:10.1007/978-3-030-87459-9.
- [18] Wikipedia, Road signs in Kyrgyzstan, 2025. URL: https://en.wikipedia.org/wiki/Road_signs_in_Kyrgyzstan.
- [19] A. Yan, Y. Wang, Y. Zhong, et al., Learning Concise and Descriptive Attributes for Visual Recognition, in: Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2023, pp. 3067-3077. doi:10.1109/ICCV51070.2023.00287.
- [20] M. Imran, N. Almusharraf, Google Gemini as a Next Generation AI Educational Tool: A Review of Emerging Educational Technology, Smart Learning Environments 11.22 (2024) pp. 1-8. doi:10.1186/s40561-024-00310-z.
- [21] P. Phuangsuwan, S. Siripipatthanakul, P. Limna, S. Siripipattanakul, Grammar and Plagiarism Checking for English Learners: The Case of Grammarly Users, Journal of Wisdom Political Science and Multidisciplinary Sciences, 7.3 (2024) 1-20. doi:10.2139/ssrn.4861177.