VGG16-based Feature Fusion For Image Kyepoint Description

Javid Norouzi
Department of Electrical Engineering
Shiraz University of Technology
Shiraz, Iran
j.norouzi@sutech.ac.ir

Alireza Liaghat
Department of Electrical Engineering
Shiraz University of Technology
Shiraz, Iran
a.liaghat@sutech.ac.ir

Mohammad Sadegh Helfroush
Department of Electrical Engineering
Shiraz University of Technology
Shiraz, Iran
ms_helfroush@sutech.ac.ir

Habibollah Danyali
Department of Electrical Engineering
Shiraz University of Technology
Shiraz, Iran
danyali@sutech.ac.ir

***Abstract*—The detection and description of feature points or keypoints play a vital role in numerous computer vision tasks. While deep learning-based methods have achieved remarkable results in various fields, they still require extensive training and resources to develop and optimize the network for specific tasks. In this work, we propose a novel method for extracting keypoint descriptors from natural images using a pre-trained VGG16 network that is specifically trained for classification tasks. Unlike previous works, our proposed approach does not require further training or modifications of the pre-trained network, which reduces the overall computational complexity and increases the efficiency of the method. The proposed method is divided into two stages: keypoint detection and description. In the keypoint detection stage, a traditional detector is employed to locate the image keypoints. In the keypoint description stage, a downscaled version of the input image is fed into the VGG16 network, and the first five convolutional layers are used to generate a rich descriptor by combining the descriptors at keypoint positions from various layers. By utilizing a sparse random projection technique, we decrease the size of the introduced descriptor, making it more efficient for keypoint matching. Our evaluation results demonstrate that our proposed method is highly competitive with traditional hand-crafted descriptors and outperforms some of them in terms of geometric transformation estimation. Additionally, our method is versatile and can be applied to different image domains by using task-specific pre-trained networks. Overall, our proposed approach offers an efficient and effective method for keypoint description with competitive performance to traditional methods and without requiring extensive training or modifications to the pre-trained network.**

***Index Terms*—VGG16, Deep Learning, Computer Vision, Key-Point, KeyPoint Description**

I. Introduction

Feature point or keypoint detection and description plays a crucial role in various complex computer vision tasks, including image registration, change detection, 3D reconstruction, object recognition, data fusion, and more. Due to its significance and practical applications, it is an active area of research. Keypoints, or feature points, are structures in the image that are precisely detected and located despite changes in viewpoint, intensity, color, or noise. Descriptors are used to establish relevance between corresponding feature points and need to be robust against different variations. In practical scenarios, keypoints and their descriptors are used to estimate the geometric transformation between a pair of images. This is possible through the process of matching keypoints across the images. This matching is established through the utilization of different similarity measures such as cosine distance, l2-norm, or hamming distance between feature point's descriptors. Keypoint detection and description methods are mainly divided into two categories; classical approaches and deep-learning-based approaches. The classical approaches use carefully hand-crafted features to detect and describe keypoints. Different structures such as blobs, corners, edges, etc are used as feature points in classical methods. HARRIS [1] is a feature point detector capable of detecting corners in the image. FAST [2] defines keypoints as the location of a pixel surrounded by a ring of brighter ones. MSER [3] uses a threshold to make a binary image and then chooses the most stable regions as keypoints by varying the threshold. Feature points are described using different information encoding schemes e.g BRIEF [4] chooses random pairs of pixels around a keypoints and compares their intensity value to create a binary descriptor. Some of the algorithms come with both feature point detector and descriptor. SIFT [5] is one of the earliest and most well-known algorithms for feature point detection and description. It detects blobs at different scales and describes them using a histogram of gradients around each feature point. SURF [6] is a faster alternative to SIFT that uses the image integral and second-order derivative of the Gaussian filter. ORB [7] combines the FAST detector and BRIEF descriptors to create a fast and efficient method similar to SIFT and SURF. BRISK [8] uses a keypoint detector based on FAST and describes them using pixel intensity comparisons around the keypoints. KAZE [9]

978-1-6654-8087-1/22/$31.00 © 2022 IEEE

and its computationally more efficient version, AKAZE [10], use the Hessian Matrix to detect scale-invariant features in a nonlinear diffusion scale space.

In recent years deep-learning approaches have been the trending topic in many different areas of research such as computer vision, natural language processing, medical science, robotics, etc. These data-driven approaches have challenged and outperformed many of the classical methods in many different fields and especially computer vision. Deep-learning-based methods are advantageous compared to classical approaches for many reasons. Hand-crafted features designed by the researchers mostly extract low-level and semantically poor features. However high-level features extracted by deep-learning-based approaches can improve the process of feature point detection and description. Deep-learning-based approaches are capable of extracting efficient feature points and descriptors through optimization and finding the best features to detect and describe keypoints. Deep learning for feature detection and description uses a single forward pass operation to detect both keypoint and their descriptors. TILDE [11] is a regressor that predicts SIFT keypoints and descriptors. SuperPoint [12] is trained to detect corners as a feature point in a supervised fashion using a synthetic dataset and then extracts a descriptor for each keypoint. QuadNet [13] uses an unsupervised training fashion with ranking loss to detect keypoints and extract descriptors. D2-Net [14] trains a network to detect reliable and robust keypoints and descriptors under extreme intensity changes. R2D2 [15] uses the same approach as D2-Net but decreases the number of model parameters and introduces a reliability heat map to specify reliable and unreliable regions for keypoint detection. ALIKE [16] tries to increase feature point localization accuracy by introducing a differentiable method for unsupervised keypoint detection.

Many studies have compared classical and deep-learning-based methods for keypoint detection and description. According to [17], although deep-learning methods can introduce intensity invariance, classical methods remain competitive for keypoint detection. The study suggests that deep-learning-based approaches do not significantly outperform classical methods in terms of keypoint detection. Therefore, we have decided to use classical keypoint extraction methods for our proposed approach.

VGG16 [18] is a CNN that was specifically trained for the classification task of the ImageNet dataset [19]. It is a popular neural network architecture that is simple to understand and modify. It has achieved state-of-the-art performance on many benchmark datasets, and its simple and uniform design makes it easy to modify and experiment with. VGG16's popularity is due to its effectiveness, simplicity, and ease of modification, which is why many researchers choose to use and modify it in their work. Other CNN architectures have also achieved state-of-the-art performance, but VGG16 remains a popular choice in the field of computer vision. Since it's creation, VGG16 has been widely used for various tasks. In [20] VGG16 has been used through transfer learning for kiwi fruit detection. In [21] VGG16 has been used for brain tumor segmentation.

Keypoint descriptors need to be distinctive. In normal circumstances, features extracted from different layers of CNNs are not necessarily distinctive enough to be used as descriptors for keypoints. Here in this work, we argue that a combination of features extracted from different layers of VGG16 are distinctive enough to be used as keypoint descriptors. To this end, we use a down-scaled version of the input image and extract features from the first 5 convolution layers of VGG16 and then we concatenate them. Using a dimensionality reduction method we decrease the final descriptor's length which decreases the computational time necessary for comparing them. The rest of this paper is structured as follows: We introduce our methodology in Section II, present the experimental setup and results in Section III, and propose our conclusions in Section VI.

II. The Proposed Method

The simplest way to extract features from a pre-trained network is to isolate a patch from the image surrounding a keypoint and feed it to a CNN. The outermost layer of the network where the extracted features have dimensions of $1 \times 1 \times feaure\ depth$ provides the most accurate description of the input image. However, this approach has a time complexity that is not feasible for practical use.

For our work, we utilized a pre-trained VGG16 network. Typically, CNNs begin by extracting high-dimensional, low-depth features from the input image and end with low-dimensional, high-depth features. The initial layer's features have a low level of abstraction but a higher level of localization, which makes them unsuitable as keypoint descriptors. These features describe primitive structures in the input image such as corners and edges. On the other hand, the features extracted from the higher layers have a higher level of abstraction but lower localization, making them more suitable as keypoint descriptors but lacking precise localization. To extract both highly localized and abstract features from the input image, we propose the method shown in figure 1, which consists of two stages: keypoint detection and descriptor extraction. In the keypoint detection stage, we use a classical keypoint detection approach to locate keypoints in the input image. These keypoints are then used for feature extraction in the second stage, where we downscale the input image and feed it to the first five layers of the VGG16 network.

For the feature extraction stage, a downscaled version of the input image is fed into the first 5 convolution layers of the VGG16 network, which include convolution layers with 64, 64, 128, 128, and 256 filters, followed by ReLu activation functions and Max Pooling layers. To reduce the time complexity of feature extraction, the input image is downscaled before feeding it into the network. The Descriptor Sampler block samples the features from normalized locations for each keypoint. Bilinear interpolation is used for sampling the descriptors. Descriptors that belong to the same location are concatenated to create a vector of size 640 for each keypoint, which includes features from different levels of abstraction and sizes.

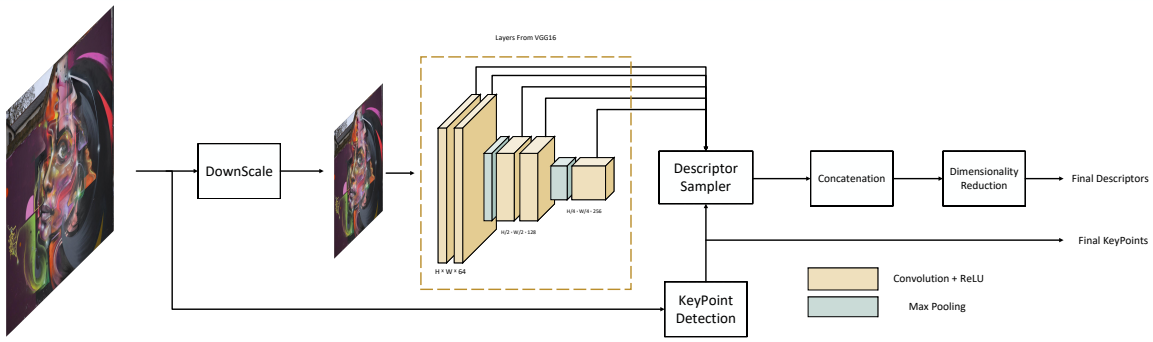


Fig. 1: Scheme of the proposed method

High-dimensional vectors take more time to compare and consume a lot of resources. For this reason, we have used dimensionality reduction blocks to reduce the descriptor size. We have used orthogonal bases obtained using a random sparse projection method. If $A \in \mathbb{R}^{n \times D}$ is a matrix consisting of n vectors of size D, then dimensionality reduction is possible by multiplying A by a random matrix $R \in \mathbb{R}^{D \times k}$, thus reducing the dimensions from D to k. R usually consists of elements from the normal distribution $N(0,1)$. Higher performance can be achieved by replacing the $N(0,1)$ elements with values from $\{-1, 0, 1\}$. To achieve higher performance with minimal loss of information, [22] proposes using the following uniform distribution for entries of matrix R:

$$R_{ij} = \begin{cases} -1 & \text{with probability of } \frac{1}{2\sqrt{D}} \\ 0 & \text{with probability of } 1 - \frac{1}{\sqrt{D}} \\ 1 & \text{with probability of } \frac{1}{2\sqrt{D}} \end{cases} \quad (1)$$

This would result in $\sqrt{D}$-fold increase in speed and very little loss in accuracy. Finally, the resulting descriptors are normalized to facilitate their comparison.

III. Experimental results and settings

A. Dataset

We evaluated our proposed method using a dataset of precisely registered images from the HPatches full image sequence [23]. The dataset comprises 116 sets of images, with each set containing six images. The geometric transformation between the first image and the rest of the images in each set is precisely estimated and provided by the authors. The dataset includes sets with both viewpoint changes and intensity changes, but we limited our evaluations to sets with viewpoint changes because classical methods are not necessarily intensity invariant, and we wanted to compare our method with classical methods.

B. Evaluation Metrics

We use the following metrics for evaluation:

- N_{all}: The number of all initial matches between keypoints across the images.
- $NOCC$: The number of correct correspondences specifies the quantity of correctly matched keypoints.
- $ROCC$: The ratio of correct correspondences measures the effect of outliers among all established correspondences and is defined as follows:

$$ROCC = \frac{NOCC}{N_{all}} \quad (2)$$

- $RMSE$: If two geometric transformations A and B are identical, then $A \times B^{-1}$ is an identity matrix. When the geometric transformation between image pairs is unknown, A and B^{-1} are the forward and backward estimated geometric transformations, respectively. In cases where the geometric transformation between image pairs is already known, B is the known geometric transformation. If f is the transformation function to transform the point P_i using the transformation matrix $A \times B^{-1}$, then the transformation error $RMSE$ can be defined as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} ||P_i - f(P_i)||^2} \quad (3)$$

  A smaller $RMSE$ indicates higher accuracy in terms of the estimation of the geometric transformation between image pairs.
- N_{valid}: If the $RMSE$ error is less than a specified threshold, we can consider the estimated transformation as a valid estimation. N_{valid} defines the number of valid estimations among all testing image pairs.

The evaluations were performed on Kaggle with no acceleration, 4 virtual CPU cores, and 32GB of physical memory.

C. Simulation Results

For evaluation, we utilized VGG16, which was pre-trained on the ImageNet dataset. In order to ensure a fair and meaningful comparison, we limited our evaluation to the viewpoint change portion of the HPatches dataset, which contains 295 pairs of images. For geometric transformation estimation, we employed the USAC algorithm [24] in this study.

Figure 2 displays the results of comparing the matching of SIFT keypoints using various descriptors extracted from various layers of VGG16. Sub-figures a to d depict the matching outcomes using a single layer of VGG16, whereas

sub-figure e shows the matching outcomes using our proposed method. It is evident that the earlier layers of VGG16 do not possess the quality required to establish a large number of correct correspondences among keypoints. As the level of abstraction increases, the number of correctly matched keypoints also increases. Our proposed method combines features from different abstraction levels to generate a rich descriptor capable of establishing more matches among the same keypoints.

Our proposed method employs a downscaled version of the input image for descriptor extraction and employs dimensionality reduction to reduce the size of the extracted descriptor. By varying the initial scale for a constant descriptor size of 640 and computing the evaluation metrics for stable results for all different scales, we observed changes in the quantity of correct matches and the quality of geometric transformation estimation. Results are considered stable if the $RMSE$ error rate is less than 5 pixels. Table I presents the mean evaluation results for all stable pairs. Decreasing the scale factor improves the $RMSE$ up to a scale factor of 0.7, and there is also a noticeable improvement in $NOCC$ and $ROCC$. As we are mainly concerned with the quality of the estimated geometric transformation, we have chosen the scale factor of 0.7 as the initial downscaling factor. It's worth mentioning that by downscaling the input image with a scale factor of 0.7, the necessary computations for descriptor extraction decrease by 51%, which is important in terms of computational cost.

The KeyPoint matching process is to find the keypoints with the most similar descriptors. This is typically done through an exhaustive search by comparing every two descriptors. Therefore, the size of descriptors plays a crucial role in the runtime of this process. A smaller descriptor size makes this process faster. To determine the most efficient descriptor size with the least information loss, we conducted a grid search by varying the size of final descriptors at a fixed initial scale of 1, and we observed the evaluation metrics for stable pairs. Stable pairs are defined as pairs of images with an $RMSE$ error rate less than 5 pixels. Table II presents our evaluation results. Based on the $RMSE$ error rate, we have chosen the size of 128 for the final descriptor size.

The main advantage of our proposed method is the use of a network specifically trained for a task other than keypoint description and applying it for keypoint description with no further training or modifications. Keypoints detected by classical methods are competitive to that of detected by deep-learning methods. Therefore we have used classical keypoint detectors in our evaluations. Table III shows the comparison result of descriptors extracted using our proposed method with some of the well-known algorithms. SIFT, BRISK, AKAZE, and ORB are used in this evaluation. For each algorithm, descriptors are extracted using our proposed method and using the original method. The $RMSE$ error threshold used for specifying invalid results is 10 pixels. The evaluated metrics for invalid results tend to be random and add noise to the evaluated results. Therefore those results are not taken into account for $RMSE$, N_{all} , $NOCC$, $ROCC$. The number of valid results is reported under N_{valid} column. The initial scale and the descriptor size are 0.7 and 128 respectively. This table shows that descriptors extracted by the SIFT algorithm perform the best among all evaluated methods and using descriptors extracted from VGG16 does not outperform SIFT descriptors in terms of $RMSE$ error rate. The possible explanation for this is that SIFT takes into account extreme rotations in the image. Using VGG16 descriptors on BRISK and AKAZE keypoints seems to have improved performance in both cases. For ORB the results are slightly better in terms of $RMSE$ error rate. In all cases, except for SIFT, the number of valid and successful estimations of geometric transformation shows improvement using VGG16 descriptors.

TABLE I: Initial scale's variation

	$Scale$	$RMSE$	N_{all}	$NOCC$	$ROCC$
1	0.3	1.45	1744.33	1281.13	0.67
2	0.4	1.26	1937.61	1492.62	0.71
3	0.5	1.15	2008.21	1552.18	0.71
4	0.6	1.17	1969.42	1491.12	0.69
5	**0.7**	**1.05**	**1948.77**	**1444.81**	**0.67**
6	0.8	1.12	1934.26	1402.70	0.66
7	0.9	1.12	1886.66	1332.87	0.64
8	1.0	1.17	1818.83	1256.33	0.62

TABLE II: Descriptor size variation

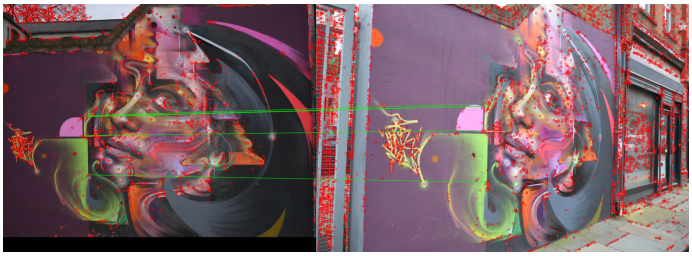
	Descriptor Size	$RMSE$	N_{all}	$NOCC$	$ROCC$
1	64	1.28	1858.38	1308.38	0.62
2	96	1.29	1899.19	1383.33	0.64
3	**128**	**1.15**	**1908.51**	**1413.21**	**0.65**
4	160	1.16	1922.33	1432.45	0.66
5	192	1.22	1931.60	1449.35	0.67
6	224	1.17	1940.50	1460.06	0.67
7	640	1.21	1972.47	1517.90	0.69

TABLE III: Comparison with classical methods

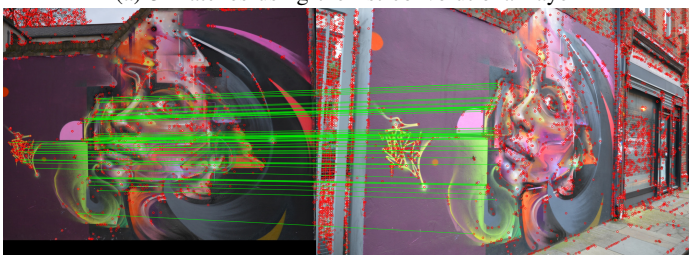
	Method	$RMSE$	N_{all}	$NOCC$	$ROCC$	N_{valid}
1	**SIFT**	**2.01**	**2516.18**	**1094.30**	**0.40**	**268**
2	SIFT + VGG16	2.32	1720.42	779.00	0.38	252
3	BRISK	2.49	3344.30	898.67	0.26	238
4	**BRISK + VGG16**	**2.02**	**2828.03**	**1126.56**	**0.34**	**251**
5	AKAZE	2.66	1612.16	554.98	0.31	236
6	**AKAZE + VGG16**	**2.31**	**1464.05**	**583.44**	**0.32**	**242**
7	**ORB**	**3.36**	**197.01**	**53.11**	**0.25**	**175**
8	ORB + VGG16	3.38	146.45	62.57	0.39	183

IV. Conclusion

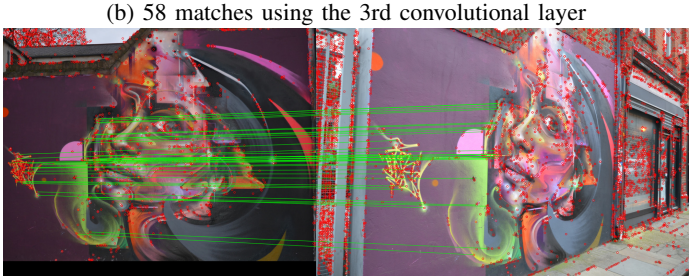
In this study, we proposed a method for extracting distinctive keypoint descriptors from natural images using a pre-trained VGG16 network that was not specifically trained for keypoint description. Our proposed scheme comprises two stages, namely keypoint detection and keypoint description. While deep-learning-based keypoint detection methods are gaining popularity, traditional keypoint detection methods are still competitive. Therefore, we employed a traditional keypoint extraction approach for keypoint detection. For keypoint description, we downscaled the input image and used the first five convolutional layers of VGG16 to extract descriptors. By combining the descriptors from various layers at the keypoint positions and concatenating them, we introduced a rich descriptor. To speed up the keypoint matching procedure,



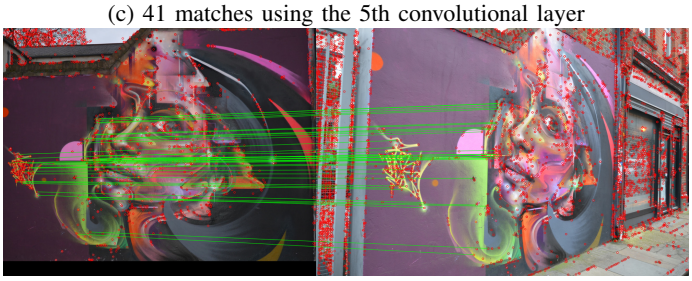
(a) 5 matches using the 1st convolutional layer



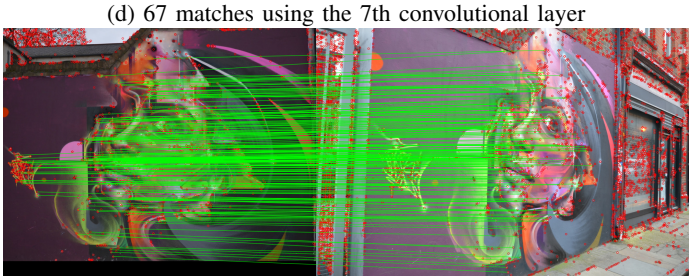
(b) 58 matches using the 3rd convolutional layer



(c) 41 matches using the 5th convolutional layer



(d) 67 matches using the 7th convolutional layer



(e) 240 matches using our proposed method

Fig. 2: The same SIFT keypoints are matched using different descriptors. The nearest neighbor is used to establish the relevance. The proposed method can establish the highest number of correct matches among descriptors extracted from different layers separately.

we reduced the descriptor size and employed a sparse random projection technique. Our evaluation results suggest that the proposed approach can extract competitive descriptors from images compared to well-constructed hand-crafted descriptors. Although we did not use a trained network for extracting unique keypoint descriptions, our proposed method could introduce descriptors that outperformed BRISK and AKAZE in terms of geometric transformation estimation. Moreover, this strategy can be extended to obtain domain-specific image descriptors by using networks trained on images from various domains.

REFERENCES

[1] C. Harris and M. Stephens, "A combined corner and edge detector, in alvey," *Vision Conference*, vol. 15, p. 50, 1988. [Online]. Available: http://dx.doi.org/10.5244/C.2.23.

[2] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," vol. 3951 LNCS, 2006.

[3] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," vol. 22, 2004.

[4] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," vol. 6314 LNCS, 2010.

[5] D. G. Low, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 2004.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, 2008.

[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," 2011.

[8] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," 2011.

[9] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "Kaze features," vol. 7577 LNCS, 2012.

[10] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," 2013.

[11] Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit, "Tilde: A temporally invariant learned detector," vol. 07-12-June-2015, 2015.

[12] D. Detone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," vol. 2018-June, 2018.

[13] N. Savinov, A. Seki, L. Ladický, T. Sattler, and M. Pollefeys, "Quad-networks: Unsupervised learning to rank for interest point detection," vol. 2017-January, 2017.

[14] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable cnn for joint description and detection of local features," vol. 2019-June, 2019.

[15] J. Revaud, P. Weinzaepfel, C. de Souza, and M. Humenberger, "R2d2: Repeatable and reliable detector and descriptor," vol. 32, 2019.

[16] X. Zhao, X. Wu, J. Miao, W. Chen, P. C. Chen, and Z. Li, "Alike: Accurate and lightweight keypoint detection and descriptor extraction," *IEEE Transactions on Multimedia*, 2022.

[17] D. Bojanic, K. Bartol, T. Pribanic, T. Petkovic, Y. D. Donoso, and J. S. Mas, "On the comparison of classic and deep keypoint detector and descriptor methods," vol. 2019-September, 2019.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv:1409.1556,.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database." Ieee, p. 248–255.

[20] Z. Liu, J. Wu, L. Fu, Y. Majeed, Y. Feng, R. Li, and Y. Cui, "Improved kiwifruit detection using pre-trained vgg16 with rgb and nir information fusion," *IEEE Access*, vol. 8, 2020.

[21] A. A. Pravitasari, N. Iriawan, M. Almuhayar, T. Azmi, Irhamah, K. Fithriasari, S. W. Purnami, and W. Ferriastuti, "Unet-vgg16 with transfer learning for mri-based brain tumor segmentation," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, 2020.

[22] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," vol. 2006, 2006.

[23] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors," vol. 2017-January, 2017.

[24] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J. M. Frahm, "Usac: A universal framework for random sample consensus," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 2022–2038, 2013.