DeepDetect: Learning All-in-One Dense Keypoints

Shaharyar Ahmed Khan Tareen
University of Houston
Houston, TX, USA
stareen@cougarnet.uh.edu

Filza Khan Tareen
National University of Sciences and Technology
Islamabad, Pakistan
ftareen.bse2021mcs@student.nust.edu.pk

***Abstract*—Keypoint detection is the foundation of many computer vision tasks, including image registration, structure-from-motion, 3D reconstruction, visual odometry, and SLAM. Traditional detectors (SIFT, SURF, ORB, BRISK, etc.) and learning-based methods (SuperPoint, R2D2, LF-Net, D2-Net, etc.) have shown strong performance yet suffer from key limitations: sensitivity to photometric changes, low keypoint density and repeatability, limited adaptability to challenging scenes, and lack of semantic understanding, often failing to prioritize visually important regions. We present DeepDetect, an intelligent, all-in-one, dense keypoint detector that unifies the strengths of classical detectors using deep learning. Firstly, we create ground-truth masks by fusing outputs of 7 keypoint and 2 edge detectors, extracting diverse visual cues from corners and blobs to prominent edges and textures in the images. Afterwards, a lightweight and efficient model: "ESPNet", is trained using these masks as labels, enabling DeepDetect to focus semantically on images while producing highly dense keypoints, that are adaptable to diverse and visually degraded conditions. Evaluations on the Oxford Affine Covariant Regions dataset demonstrate that DeepDetect surpasses other detectors in keypoint density, repeatability, and the number of correct matches, achieving maximum values of 0.5143 (average keypoint density), 0.9582 (average repeatability), and 59,003 (correct matches).**

***Index Terms*—keypoint detection, image matching, deep learning, neural networks, 3D reconstruction, SLAM.**

I. Introduction

Keypoint detection is the backbone of many computer vision applications including image matching, panorama stitching, object tracking, 3D reconstruction, visual odometry, and visual SLAM [1]. Keypoints (often referred to as features or feature points) are the distinctive points in an image that can also be identified to an extent under various transformations or degradations by the detectors. They are usually corners, blobs, ridges, and textured patterns that stand out from their surroundings [1]. Selection of keypoint detector is a critical decision in computer vision as it determines the stability, repeatability, map density, and overall performance of the downstream application [2]–[6]. A variety of detectors have been proposed over the years, ranging from traditional algorithms (SIFT, SURF, Harris Corner Detector, ORB, BRISK, and AGAST) to deep learning based approaches (SuperPoint, R2D2, LF-Net, and D2-Net). After keypoint detection, corresponding feature descriptors are computed by encoding the local pixel neighborhood around each keypoint into a distinctive vector that enhances stability across image variations [1]. A good descriptor is more distinctive and invariant to diverse image transformations or degradations [7]–[10].

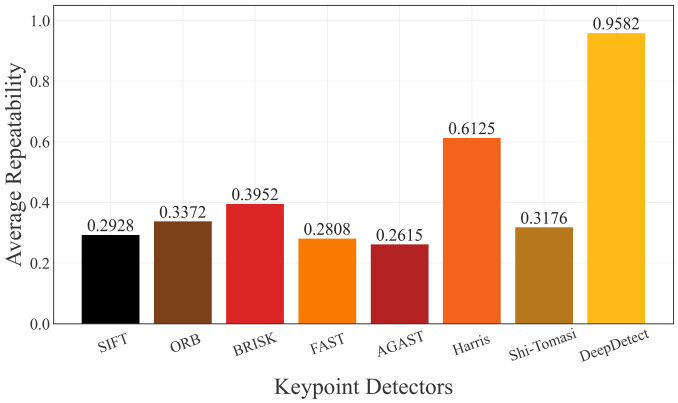


Fig. 1: Average repeatability of keypoint detectors on Oxford Dataset [11].

After detection and description, "feature matching" is performed to match the descriptors and establish geometric relationships between the images [7]. NNDR is an effective matching strategy [12], that combines nearest neighbor search with a distance ratio test to reduce incorrect matches. The resulting matches are filtered by using model fitting algorithms: RANSAC [13], PROSAC [14], and MSAC to reject the outliers. Reliable image matching depends on the effectiveness of four stages: keypoint detection, keypoint description, feature (descriptor) matching, and outlier rejection [7], enabling robots or machines to accurately perceive, localize, and generate dense maps of their surroundings. Therefore, achieving high performance in the primary stage: **keypoint detection**, is crucial, as weaknesses at this stage propagate and lead to degradation in the performance of downstream tasks.

Classical detectors are invariant to different image transformations (scale, rotation, affine, blur) but sensitive to extreme variations (intense darkness, extreme photometric/geometric changes, severe smoke or fog) [8]. With standard settings, they detect sparse keypoints that inadequately cover texture-rich and low-contrast regions [8]. They also lack semantic understanding to prioritize important regions in the images. Learning-based detectors (SuperPoint, R2D2, LF-Net, D2-Net) improve keypoint generalization and stability but their qualitative results show similar shortcomings [15]–[18]. They detect sparse keypoints and lack semantic awareness, potentially missing important regions in cluttered or low-visibility scenes [15]–[18]. These issues highlight the need for an intelligent and dense keypoint detector that is robust to complex image transformations and focuses on important regions.

Preprint Version: Submitted to 22nd IEEE International Conference on Frontiers of Information Technology (FIT 2025)

(a) Original Image (b) SIFT$_{(\text{Normal})}$ (c) SIFT$_{(\text{Low})}$ (d) DeepDetect

Fig. 2: Keypoint detection on two images using SIFT with normal (default) thresholds, SIFT with extremely low thresholds, and DeepDetect. Number of detected keypoints: Top Scene — 279, 23990, 46309 and Bottom Scene — 1009, 75465, 51115, respectively. SIFT with standard thresholds yields limited keypoints, whereas SIFT with extremely low thresholds produces high number of keypoints, detecting noisy features on semantically irrelevant regions. DeepDetect produces highly dense keypoints with substantially lower noise, well concentrated in semantically important regions of the images.

Key Contributions: Our contributions are highlighted below:

- We present **DeepDetect**, an **intelligent**, **adaptable**, **all-in-one**, and **dense** keypoint detector that uses deep learning to learn the strengths of 7 strong keypoint detectors (SIFT, ORB, BRISK, FAST, AGAST, Harris Corner Detector, and Shi-Tomasi Corners) and 2 well-known edge detectors (Canny and Sobel). It provides highly dense and semantically meaningful keypoints, demonstrating robustness to poor visibility and brightness, low contrast, and complex scenes, without requiring manual tuning (as in classical detectors).
- We propose a multi-detector, multi-edge fusion pipeline that generates diverse and rich keypoint supervision masks, which serve as labels for training DeepDetect.
- We use a lightweight and efficient model: "ESPNet", to ensure that DeepDetect has a small memory footprint: **1.82 MB**, making it well-suited for deployment on edge devices such as mobile phones.
- We demonstrate that DeepDetect outperforms traditional keypoint detectors both qualitatively and quantitatively by significant margins. On the Oxford dataset [11], DeepDetect achieves the highest performance, with an average repeatability of **0.9582**, average keypoint density of **0.5143**, and **59,003** correct matches, outperforming the traditional detectors even with their extremely low thresholds.
- The code of DeepDetect is available at the following link: **https://github.com/saktx/DeepDetect**

II. Related Work

Classical feature detectors have laid the foundation of modern computer vision. However, they rely on handcrafted criteria such as gradient magnitude, intensity variation, or corner response thresholds. Although computationally efficient, they often fail with standard thresholds under extreme image transformations, lacking adaptability to changing visibility conditions [8]. Learning based methods aim to learn keypoint detection, description (or sometimes both) but they typically produce sparse keypoints [15]–[18] which are not suitable for computer vision applications that require dense keypoints such as dense 3D reconstruction, AR/VR, and SLAM.

A. Classical Detectors and Descriptors

SIFT [12], a powerful scale and rotation invariant algorithm, uses Difference-of-Gaussians for keypoint detection and describes them with gradient-based histograms. SURF [19] is a faster alternative to SIFT that uses integral images and box filters for detection. It also provides feature descriptors but they are less distinctive than SIFT. KAZE [20] detects nonlinear scale-space keypoints directly in the image space yielding highly distinct features but at a higher computational cost whereas AKAZE is accelerated version of KAZE that uses binary descriptors (M-LDB) for faster computation while retaining much of the distinctiveness of KAZE. ORB [21] combines FAST detector with a rotated version of the BRIEF descriptor for speed and rotation invariance.

BRISK [22] uses a circular sampling pattern for detection and generates binary descriptors, offering scale and rotation invariance with high speed and moderate distinctiveness. FAST [23] is an efficient corner detector that examines a circle of 16 pixels around a candidate keypoint. AGAST [24] is an improvement over FAST, designed to be faster and more robust. Harris Corner Detector (HCD) detects corners by analyzing local intensity changes and Shi-Tomasi Corners focus on stronger minimum eigenvalues, leading to more stable corner detection. Canny [25] is a multi-stage detector that uses gradient computation, non-maximum suppression, and hysteresis thresholding to produce thin and fine edges. Sobel [26] uses convolution with gradient kernels to find edges in the x and y directions, providing thicker edges.

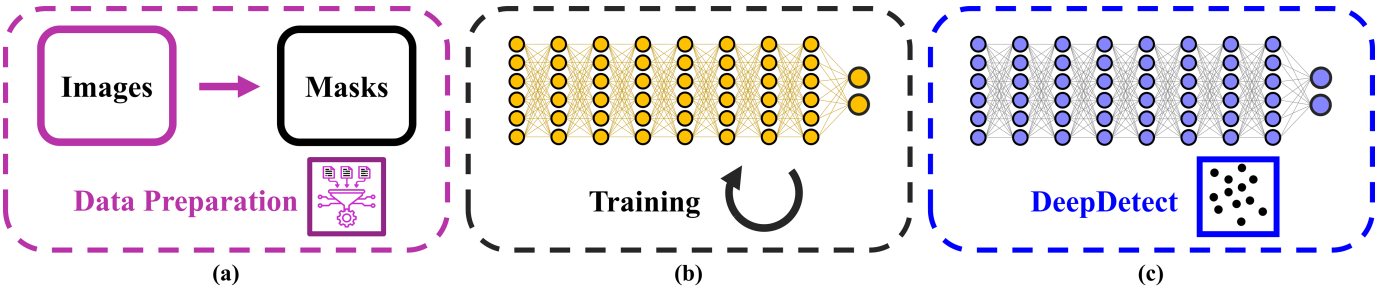


Fig. 3: Main stages in the development of DeepDetect: (a) Fusion masks are created by using 7 keypoint detectors (SIFT, ORB, BRISK, FAST, AGAST, Harris, Shi-Tomasi) and 2 edge detectors (Canny, Sobel) in the images, combining their strengths to capture rich and diverse visual representations. (b) The fused binary masks are then used as labels for the corresponding images to supervise and train ESPNet (a lightweight, efficient model). (c) DeepDetect is ready to detect all-in-one, dense keypoints with semantic awareness and adaptability to complex image degradations, without requiring manual tuning.

B. Learning based Detectors and Descriptors

SuperPoint [15] is a well-known learning based detector and descriptor trained via synthetic homographies, offering high repeatability and descriptor quality across varying conditions. R2D2 [16] learns both repeatability and descriptor reliability via deep networks, filtering out unstable keypoints for robust matching. LF-Net [17] jointly learns detection and description in a differentiable pipeline using image pairs and geometric constraints. D2-Net [18] uses a CNN to jointly detect and describe dense keypoints in a single forward pass, particularly effective for challenging photometric and geometric changes. Although having some strengths, these algorithms lack semantic awareness about the scenes and do not provide dense keypoints that are especially required in mapping applications such as dense 3D reconstruction, AR/VR, and visual SLAM.

III. DeepDetect – Methodology

A. Data Preparation

Data preparation is the first stage towards the development of DeepDetect as shown in Figure 3. We selected 41000 images from the MS-COCO [27] and NewTsukuba [28] datasets, applying extreme reductions in brightness and contrast to 25% of them. Afterwards, they were split into training and validation sets of 33000 and 8000 images, respectively. The strengths of the 7 keypoint detectors and 2 edge detectors are then unified by creating a combined binary mask of their outputs for each image. SIFT detects blob-like structures, while the other 6 keypoint detectors identify corners or ridges. Canny highlights fine details, whereas Sobel captures thicker, high-intensity edges in the images. Binary masks are generated with pixels belonging to the detected keypoints and edges marked as 1 and all other pixels as 0. Extremely low detection thresholds are used for the low visibility scenes, whereas for normal scenes moderate thresholds are used. This mitigates irrelevant keypoints from the normal scenes while maximizes their detection in the challenging scenes. For each image, its keypoint and edge masks are merged to create a rich, multi-cue supervision mask that serves as its label during training. Given an image $I \in \mathbb{R}^{H\times W\times 3}$, set of keypoint detectors D = {SIFT, ORB, BRISK, FAST, AGAST, Harris, Shi-Tomasi} and set of edge detectors E = {Canny, Sobel}, the final combined mask M(I), shown in Figure 4(d), is obtained as:

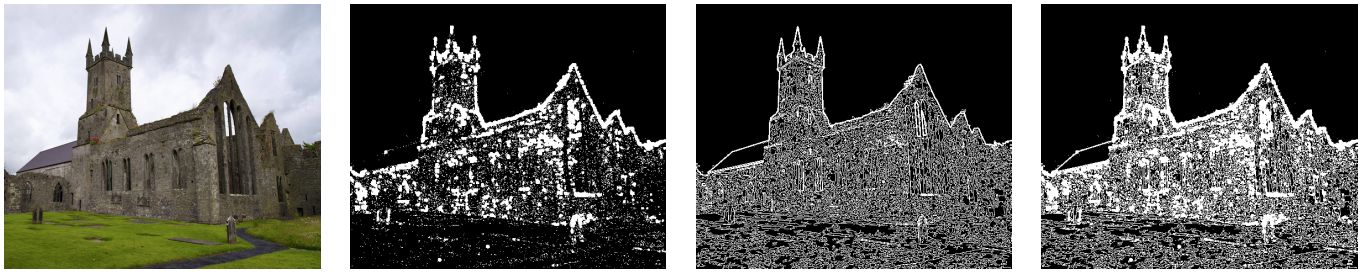
$$M(I) = \bigvee_{d\in D} M_D(I) \vee \bigvee_{e\in E} M_E(I) \tag{1}$$

where $\vee$ denotes the pixel wise logical OR operation, $M_D(I)$ shows the fusion mask obtained from the keypoint detectors as in Figure 4(b) and $M_E(I)$ shows the fusion mask obtained from the edge detectors as in Figure 4(c). The construction of the final mask can also be expressed as:

$$M(I) = \begin{cases} 1, & \text{if } \exists\, p \in D \cup E \text{ such that } M_p(I) = 1 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

B. ESPNet based Training

Efficient Spatial Pyramid Network (ESPNet) [29] is a light weight model designed for highly efficient pixel-wise prediction tasks such as "semantic segmentation" on resource-constrained devices. It has encoder-decoder architecture: the



(a) Original Image (b) Keypoints Mask (c) Edges Mask (d) Combined Mask

Fig. 4: Binary masks obtained from 7 keypoint and 2 edge detectors, along with their combined version (which provides richer representations for training).

encoder captures contextual information through successive convolution and down-sampling layers while the decoder progressively up-samples the feature maps to restore the spatial resolution. ESPNet is based on ESP modules which factorize the standard convolutions into point-wise (1×1) projections followed by spatial pyramids of dilated convolutions [29]. These modules also have residual connections that assist in smooth feature propagation through the model. ESPNet works on a lower dimensional space due to the point-wise projections, reducing its memory footprint and computational cost. Compared to heavy image segmentation models like U-Net [30] and PSPNet [31], it is upto $180\times$ smaller, $22\times$ faster, and an excellent choice for low complexity tasks [29].

In the training stage (see Figure 3), ESPNet is trained on the prepared dataset for 100 epochs with a learning rate of 0.001 and batch size of 64, using cosine annealing scheduler for smoother convergence. We used Python, OpenCV, and PyTorch [32] for coding. The input image size is kept consistent at $3 \times 480 \times 480$ for the model and its predicted output mask is re-sized to match the spatial dimensions of the original image. Given a target mask $M(I) = y \in \{0,1\}^{H \times W}$ and model's raw output $z \in \mathbb{R}^{H \times W}$, the BCE loss $L(y, z)$ is described as:

$$L = -\frac{1}{N} \sum_{i=1}^{N} \Big[y_i \log(\sigma(z_i)) + (1 - y_i) \log(1 - \sigma(z_i)) \Big] \quad (3)$$

where σ is the sigmoid function and $N = H \times W$ represents the total number of pixels in the mask. The minimization of loss in Equation 3 encourages the model to produce probabilities closer to 1 for the keypoint pixels and closer to 0 for others. Figure 5 shows the training and validation loss curves of DeepDetect. We picked the model that provided best validation loss during training. During inference, the output z of the model is passed through the sigmoid function to generate the probability $P_p = \sigma(z)$ for pixel p. The predicted output of the model is then thresholded at $\tau = 0.5$ to obtain the final binary mask. The threshold can be adjusted using Equation 4 as per the required keypoint density in the images.

$$M(I) = \begin{cases} 1, & \text{if } P_p \geq \tau \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

C. DeepDetect — Dense Keypoint Detection

DeepDetect is not only learned to reproduce the geometric diversity of the classical keypoint and edge detectors but also able to capture semantically important regions, leading to dense and foreground focused keypoints. To perform image matching, keypoints are first detected in the image pair using the trained ESPNet. Afterwards, we use SIFT descriptor to compute the corresponding descriptors, because of its highly distinctive description nature. The computed descriptor sets for the two images are then matched using NNDR based matching. RANSAC can be applied to reject the outliers and filter the correct matches. Qualitative results of DeepDetect are shown in Figure 6. For the comparative analysis presented in Table II

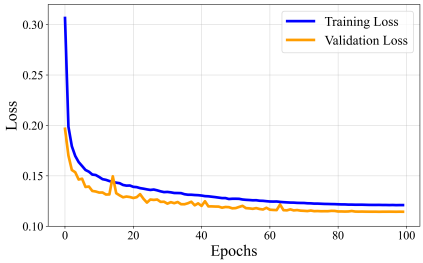


Fig. 5: Training and validation loss curves of DeepDetect.

and Figure 6, we used the ground-truth Homography matrices [11] (not RANSAC) to obtain the number of correct matches.

IV. Experiments & Results

We compare DeepDetect with other keypoint detectors on the Oxford Affine Covariant Regions Dataset [11] using the following evaluation metrics. In all experiments, we used the SIFT descriptor with all the detectors except ORB and BRISK, for which we used their own descriptors.

Keypoint Density: Shows how many keypoints are detected per unit image area. It helps assess whether a detector produces sparse or dense keypoints. Given N detected keypoints in an image with area $H \times W$, the keypoint density is calculated as:

$$\text{Keypoint Density} = \frac{N}{H \times W} \quad (5)$$

Repeatability: Measures the consistency of a detector in finding the same keypoints in images under a given transformation. If N_A and N_B are the number of keypoints detected in two overlapping images A and B, and $N_{A \cap B}$ is the number of keypoints within the common region $A \cap B$, then:

$$\text{Repeatability} = \frac{N_{A \cap B}}{\min(N_A, N_B)} \quad (6)$$

Foreground-Keypoint (F-KP) Ratio: Evaluates the ability of a detector to detect keypoints within the foreground regions in images. It shows how well a detector focuses on foreground regions rather than the background or irrelevant areas with

TABLE I: Comparison of Average Keypoint Density and Foreground-to-Keypoint (F-KP) Ratio of the Detectors.

Detector	Average Keypoint Density	F-KP Ratio
SIFT	0.0090	**0.7606**
SIFT (Low)	0.1993	**0.3559**
ORB	0.0009	**0.8508**
ORB (Low)	0.1256	**0.6797**
BRISK	0.0127	**0.7909**
BRISK (Low)	0.0800	**0.5716**
FAST	0.0270	**0.7442**
FAST (Low)	0.3664	**0.5387**
AGAST	0.0280	**0.7363**
AGAST (Low)	0.3550	**0.5669**
Harris	0.0246	**0.8262**
Harris (Low)	0.2988	**0.7389**
Shi-Tomasi	0.0052	**0.8026**
Shi-Tomasi (Low)	0.0326	**0.6180**
DeepDetect ($\tau = 0.9$)	**0.4170**	**0.7114**
DeepDetect ($\tau = 0.8$)	**0.4493**	**0.7113**
DeepDetect ($\tau = 0.7$)	**0.4733**	**0.7109**
DeepDetect ($\tau = 0.5$)	**0.5143**	**0.7093**

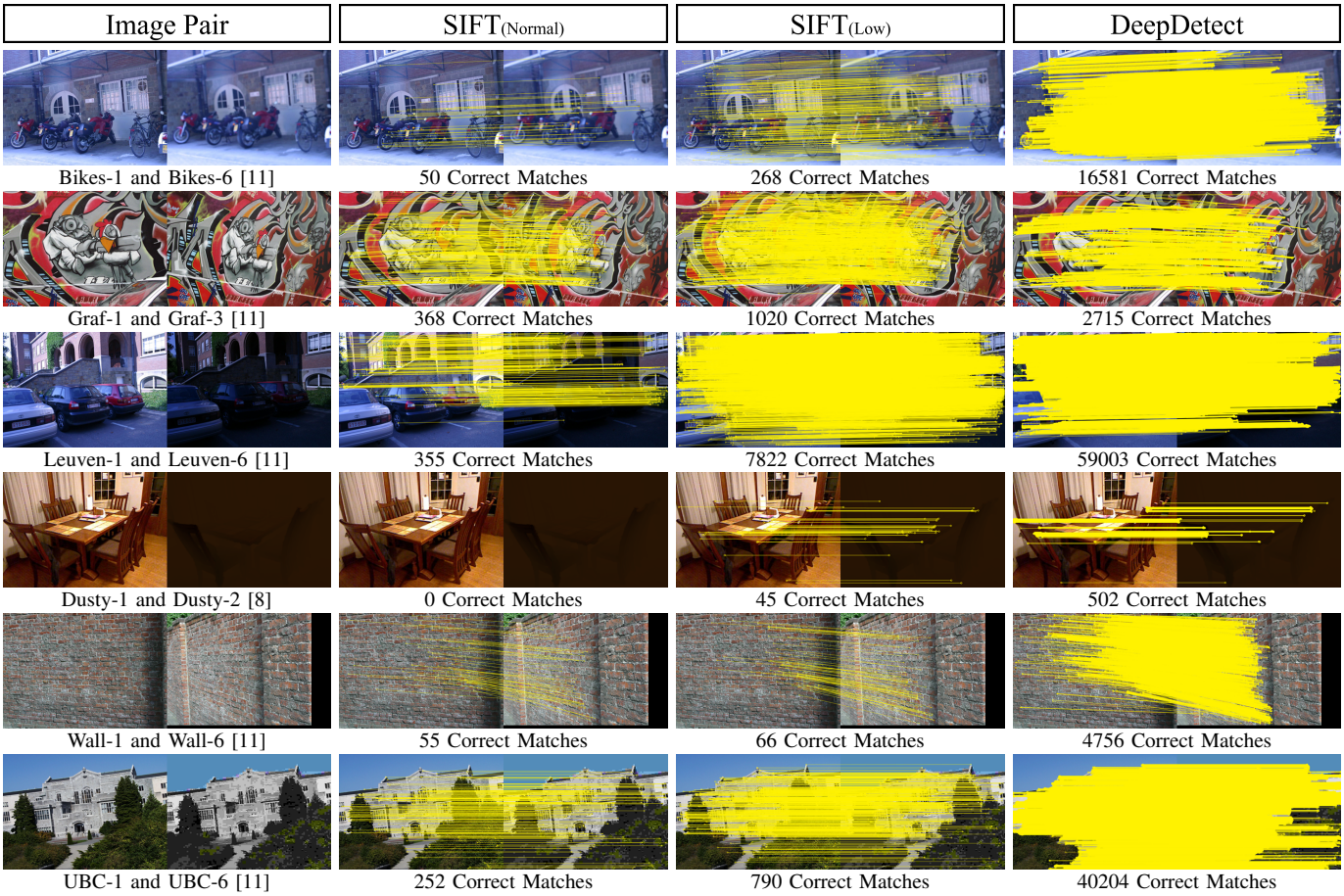


Fig. 6: Qualitative comparison of DeepDetect with SIFT under default and extremely low [8] thresholds. The yellow lines show correct correspondences between the images. DeepDetect provides substantially higher correct matches, without requiring manual tuning (additional results provided in the appendix).

TABLE II: Comparison of Correct Matches Produced by Keypoint Detectors for Selected Image Pairs from Oxford Dataset [11].

Image Pair	SIFT (Low)	ORB (Low)	BRISK (Low)	FAST (Low)	AGAST (Low)	Harris (Low)	Shi-Tomasi (Low)	DeepDetect
Graf (1–6)	20	5	6	13	5	54	6	**102**
Boat (1–2)	6517	3749	3836	1703	16238	33385	2164	**47447**
Wall (1–6)	66	13	14	2156	2357	4270	175	**4756**
Bikes (1–6)	268	577	358	**18112**	15458	4652	449	16581
Leuven (1–6)	7822	3582	3163	55427	53972	24296	4159	**59003**
Trees (1–6)	118	234	181	12078	**12185**	3834	178	9278
UBC (1–6)	790	977	464	13024	11327	28655	257	**40204**

low or no texture. Let N_F be the number of keypoints in the foreground region and N_T the total detected keypoints, then:

$$\text{Foreground-to-Keypoint Ratio} = \frac{N_F}{N_T} \tag{7}$$

Correct Matches: Higher number of correct matches indicates better performance. If N_A and N_B are the number of keypoints detected in two overlapping images A and B, then N_C is the number of correct matches between the two sets of keypoints, such that the keypoints of image A are projected onto image B using the ground-truth Homography, and the distance of each keypoint in A to its corresponding keypoint in B is less than a threshold (we used threshold of 1.0 pixel).

Results show that DeepDetect achieves state-of-the-art performance. On the Oxford dataset [11], it gives an **average repeatability score** of **0.9582**, surpassing other keypoint detectors by a significant margin. The results for **average keypoint density** are presented in Table I, where DeepDetect achieves the highest average density of **0.5143**. Table I also reports average foreground-to-keypoint ratio on 18 custom-selected images with ground-truth binary masks. DeepDetect achieves good ratios, indicating that its keypoints are not only dense but also well-concentrated in semantically important regions in the images. With default thresholds, the classical detectors yield higher F-KP ratios but significantly lower keypoint densities. Conversely, reducing thresholds to increase density decreases the F-KP ratio, highlighting the presence of noisy keypoints. For instance, FAST and AGAST with low thresholds, achieve high densities with low F-KP ratios, showing their inability to concentrate on salient regions. DeepDetect produces considerably higher densities while maintaining strong foreground focus. Visual comparisons in Figure 2 clearly show that Deep-

Detect focuses on semantically meaningful regions and provides very high keypoint density. Table II and Figure 6 show that DeepDetect provides substantially higher correct matches and consistently outperforms classical detectors. The strength of DeepDetect lies in its ability to combine the geometric robustness and diversity of keypoint and edge detectors, and adaptability to extreme image variations gained through the use of supervision masks generated with different thresholds for normal and challenging scenes. DeepDetect demonstrates stronger generalization across degraded visual conditions due to its intelligent nature.

CONCLUSION

We introduce **DeepDetect**, an intelligent, all-in-one, dense keypoint detector that unifies the robustness and distinctiveness of 7 keypoint detectors and 2 edge detectors using deep learning. By leveraging multi-detector and multi-edge fusion, we constructed diverse supervision masks that enable DeepDetect to learn rich, dense, and semantically focused keypoints in the images. Extensive experiments on Oxford Affine Covariant Regions demonstrate that DeepDetect consistently outperforms other detectors in repeatability, keypoint density, and correct matches while maintaining robustness under extreme image variations. Its ability to produce highly dense, repeatable, and semantically meaningful keypoints without manual tuning, make it a well suited detector for dense 3D reconstruction, visual SLAM, AR/VR, object tracking, and autonomous navigation applications in adverse environments.

REFERENCES

[1] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[2] S. Cygert and A. Czyżewski, "Toward robust pedestrian detection with data augmentation," *IEEE Access*, vol. 8, pp. 136 674–136 683, 2020.

[3] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International journal of computer vision*, vol. 74, no. 1, pp. 59–73, 2007.

[4] R. Garcia and N. Gracias, "Detection of interest points in turbid underwater images," in *OCEANS 2011 IEEE-Spain*. IEEE, 2011, pp. 1–9.

[5] L. Li, J. Lian, L. Guo, and R. Wang, "Visual odometry for planetary exploration rovers in sandy terrains," *International Journal of Advanced Robotic Systems*, vol. 10, no. 5, p. 234, 2013.

[6] K. Ebadi, L. Bernreiter, H. Biggie, G. Catt, Y. Chang, A. Chatterjee, C. E. Denniston, S.-P. Deschênes, K. Harlow, S. Khattak *et al.*, "Present and future of slam in extreme underground environments," *arXiv preprint arXiv:2208.01787*, 2022.

[7] S. A. K. Tareen and Z. Saleem, "A comparative analysis of sift, surf, kaze, akaze, orb, and brisk," in *2018 International conference on computing, mathematics and engineering technologies (iCoMET)*. IEEE, 2018, pp. 1–10.

[8] S. A. K. Tareen and R. H. Raza, "Potential of sift, surf, kaze, akaze, orb, brisk, agast, and 7 more algorithms for matching extremely variant image pairs," in *2023 4th International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. IEEE, 2023, pp. 1–6.

[9] L. Juan and O. Gwun, "A comparison of sift, pca-sift and surf," *International Journal of Image Processing (IJIP)*, vol. 3, no. 4, pp. 143–152, 2009.

[10] E. Karami, S. Prasad, and M. Shehata, "Image matching using sift, surf, brief and orb: performance comparison for distorted images," *arXiv preprint arXiv:1710.02726*, 2017.

[11] V. G. Group, "Affine covariant regions datasets," http://www.robots.ox.ac.uk/~vgg/data, 2004, accessed: Aug. 14, 2025.

[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[13] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[14] O. Chum and J. Matas, "Matching with prosac-progressive sample consensus," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 220–226.

[15] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[16] J. Revaud, C. De Souza, M. Humenberger, and P. Weinzaepfel, "R2d2: Reliable and repeatable detector and descriptor," *Advances in neural information processing systems*, vol. 32, 2019.

[17] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "Lf-net: Learning local features from images," *Advances in neural information processing systems*, vol. 31, 2018.

[18] 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," in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2019, pp. 8092–8101.

[19] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[20] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "Kaze features," in *European conference on computer vision*. Springer, 2012, pp. 214–227.

[21] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[22] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2548–2555.

[23] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*. Springer, 2006, pp. 430–443.

[24] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *European conference on Computer vision*. Springer, 2010, pp. 183–196.

[25] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 2009.

[26] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the sobel operator," *IEEE Journal of solid-state circuits*, vol. 23, no. 2, pp. 358–367, 1988.

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[28] M. Peris, S. Martull, A. Maki, Y. Ohkawa, and K. Fukui, "Towards a simulation driven stereo vision system," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, 2012, pp. 1038–1042.

[29] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proceedings of the european conference on computer vision (ECCV)*, 2018, pp. 552–568.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[31] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[32] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.