SynShapes: A Synthetic Image-Annotation Dataset for Edge Detection

Guillermo A. Castillo[1][a], Xavier Soria[2][b] and Angel D. Sappa[1,3][c]
[1]*ESPOL Polytechnic University, ESPOL, Campus Gustavo Galindo Km. 30.5 Via Perimetral, Guayaquil, Ecuador*
[2]*EspochAI, Polytechnic School of Chimborazo (ESPOCH), Macas, 140101, Ecuador*
[3]*Computer Vision Center, Edificio O Campus UAB Bellaterra, Barcelona, Spain*

Keywords: Edge Detection, Synthetic Data, Domain Adaptation, Annotation.

Abstract: In the edge detection area, generating ground-truth edge annotations is a time-consuming, costly and subjective process, often leading to inconsistent labels and limited scalability. To address this issue, we propose SynShapes, a synthetic dataset composed of automatically generated image–annotation pairs designed to address these limitations. The dataset is created using a Python-based pipeline that produces consistent ground-truth edge maps from randomly generated geometric shapes; and in order to reduce the gap between synthetic and real data, a Fourier Domain Adaptation (FDA) strategy is applied. Then, SynShapes is benchmarked by training the TEED edge detector model on multiple datasets: BIPED, BSDS500, MDBD and SynShapes with FDA; evaluations across the training datasets and the UDED dataset show that SynShapes with FDA achieves performance comparable to, and in some cases surpassing those trained on real-world datasets; all without the cost and bias of manual annotation. The dataset is publicly available on: https://vision-cidis.github.io/SynShapes/.

1 INTRODUCTION

Edge detection is one of the most important low-level tasks in computer vision, and it is often the first step in many applications such as segmentation, recognition or image understanding. Classical algorithms, such as, Canny (Canny, 2009) or Sobel (Marr and Hildreth, 1980), among others, have been widely used for decades to extract edges from intensity gradients. However, these traditional techniques are not robust when facing complex real-world scenes, where lighting changes, textures, or noise affect the results. In recent years, deep learning has brought a significant improvement in this field, providing models that learn directly from data rather than relying on hand-crafted filters. Well-known methods like HED (Xie and Tu, 2015), BDCN (He et al., 2019), DexiNed (Soria et al., 2023b) or LDC (Soria et al., 2022) have shown that convolutional neural networks can produce accurate and thin edge maps, even in challenging conditions.

The success of deep learning models depends strongly on the quality of the training data. The main limitation in edge detection is that preparing a ground truth dataset is a very slow and subjective process. Annotators must manually mark all edges in each image, which is time-consuming and prone to human errors. In many cases, it is difficult to decide which transitions are relevant as real edges and which are just texture or shadow. As a result, the final annotations are not always consistent between images or annotators. In addition, some widely used datasets, such as BSDS (Martin et al., 2001) or NYUD-v2 (Silberman et al., 2012), were originally created for segmentation or boundary detection, not specifically for edges. This makes the training less reliable, and models tend to overfit or fail when evaluated on unseen images.

Several recent works have focused on improving efficiency and reducing the number of parameters in deep models for edge detection. Approaches like LDC or TEED (Soria et al., 2023a) demonstrate that a small architecture can reach good performance when trained on clean and well-defined datasets like BIPED (Soria et al., 2023b). However, even in these cases, the problem of the labeled data remains. Creating large and consistent edge datasets still requires many hours of human work and careful validation, which limits the scalability of current methods.

To overcome this limitation, synthetic data offers a very interesting alternative. Using computer-generated scenes, it is possible to obtain perfect and

[a] https://orcid.org/0009-0002-6105-4590
[b] https://orcid.org/0000-0003-2997-2439
[c] https://orcid.org/0000-0003-2468-0031

Castillo, G. A., Soria, X. and Sappa, A. D.
SynShapes: A Synthetic Image-Annotation Dataset for Edge Detection.
DOI: 10.5220/0014632400004084

In *Proceedings of the 21st International Conference on Computer Vision Theory and Applications (VISAPP 2026) - Volume 2*, pages 794-802
ISBN: 978-989-758-804-4; ISSN: 2184-4321

noise-free edge annotations directly from the rendering process, without any human labeling. Synthetic images can be easily created under different conditions, for example, with variations in illumination, texture or viewpoint, and also for other modalities like thermal or infrared images, where collecting real labeled data is even more difficult. Combining synthetic and real images, or applying domain adaptations, could help to improve the generalization of deep learning edge detectors on new real world environments.

In this paper, we explore the use of synthetic images for training deep edge detection models. We propose a simple strategy to generate synthetic data (image–annotation pairs) and incorporate a domain adaptation technique to reduce the gap between real and synthetic data.

For the evaluation, the Optimal Dataset Scale (ODS), Optimal Image Scale (OIS) and Average Precision (AP) metrics are used in order to provide a standardized edge detection benchmark. Experimental results show that models trained with synthetic data achieve a comparable or in some cases better performance than those trained with hand-labeled data, while avoiding the high cost and human bias associated with manual annotation. Our study suggests that synthetic data represent a promising solution for building more scalable and general edge detection systems.

The manuscript is organized as follows. Section 2 presents works related to the edge detection problem, including synthetic and real dataset. Section 3 details the proposed approach, including the synthetic image generation, the synthetic-to-real adaptation strategy and the relevant details of the model used for edge detection. Then, Section 4 presents quantitative and qualitative results using the proposed SynShapes dataset. Finally, discussion and conclusions are given in Section 5.

2 RELATED WORK

As mentioned in the previous section, the performance of deep learning–based edge detectors is tightly coupled to the quality of their ground-truth (GT) annotations. By quality, we refer to both the *completeness* of annotated edges and the spatial *precision* of the edges themselves (Cetinkaya et al., 2024).

To address these two key challenges, several datasets rely on annotations provided by multiple labelers (e.g., BSDS (Arbelaez et al., 2011), MDBD (Mély et al., 2016)), with the aim of reaching a consensus on the annotated edges. However, this strategy has been shown to be insufficient. On the one hand, many datasets exhibit a lack of agreement among annotators, which often results in multiple annotations for the same physical edge (see Fig. 1*(top)* and *(middle)* rows). However, despite the involvement of multiple annotators, the images still contain missing edges; this can be observed by comparing the highlighted regions with the corresponding original images, where visually evident boundaries remain unlabeled.

As a consequence, the problem of unreliable or incomplete edge annotations remains open. For this reason, in datasets such as BSDS, state-of-the-art CNN-based methods—including HED (Xie and Tu, 2015) and RCF (Liu et al., 2019)—apply a consensus-thresholding step to the multi-annotator labels prior to training, ensuring that the model learns from more reliable edge maps. Following this strategy, subsequent architectures—including BDCN (He et al., 2019) and transformer-based EDTER (Pu et al., 2022)—also rely on consensus-filtered GTs. Lightweight approaches such as PiDiNet (Xu et al., 2022) and PiDiNext (Li et al., 2025b) adopt the same preprocessing step for BSDS.

More recent works explicitly address annotation ambiguity. Models such as UAED (Zhou et al., 2023) and MUGE (Zhou et al., 2024) incorporate uncertainty modeling to capture the subjectivity inherent in multi-annotator datasets. These methods convert deterministic labels into learnable Gaussian distributions, using the predicted variance as an uncertainty estimate to guide training. This probabilistic framework has achieved unprecedented performance on BSDS dataset. The latest trend explores self-supervised edge GT assistance using large-scale segmentation models such as SAM, as demonstrated in SAUGE (Liufu et al., 2025).

From a different perspective, the issue of multi-annotator inconsistency was addressed through the introduction of the BIPED dataset (Poma et al., 2020), which was later refined in (Soria et al., 2023b). BIPED provides high-quality, multi-level consensus annotations for each image, enabling improved generalization across edge detection models. Methods such as DexiNed (Soria et al., 2023b), DiffusionEdge (Ye et al., 2024), and EDMB (Li et al., 2025a) achieve state-of-the-art performance under diverse architectural paradigms. Likewise, lightweight detectors—including LDC (Soria et al., 2022), TEED (Soria et al., 2023a), and LED-Net (Ji et al., 2025)—benefit substantially from the curated annotations provided by BIPED, delivering strong results despite their compact designs.

Despite these advances and the considerable ef-

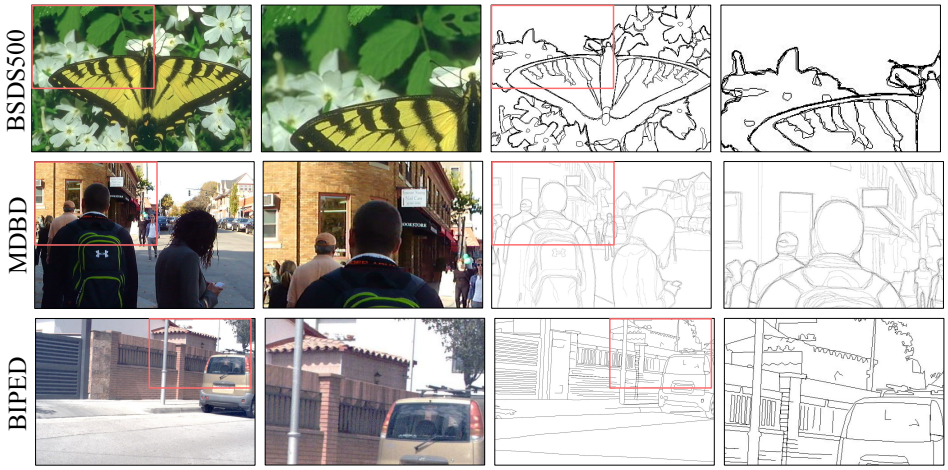

Figure 1: RGB images and their corresponding ground-truth annotations from different datasets. These examples highlight several common issues: *(i)* lack of annotator consensus, illustrated by the differing edge maps in the zoomed regions of BSDS and MDBD; and *(ii)* incomplete edge annotations—BSDS500: edges corresponding to the green leaves at the top are missing; MDBD: horizontal window edges as well as text contours in the panels are not annotated; and BIPED: edges related to brick patterns on walls and fine details on the car, such as the windshield wiper, are absent.

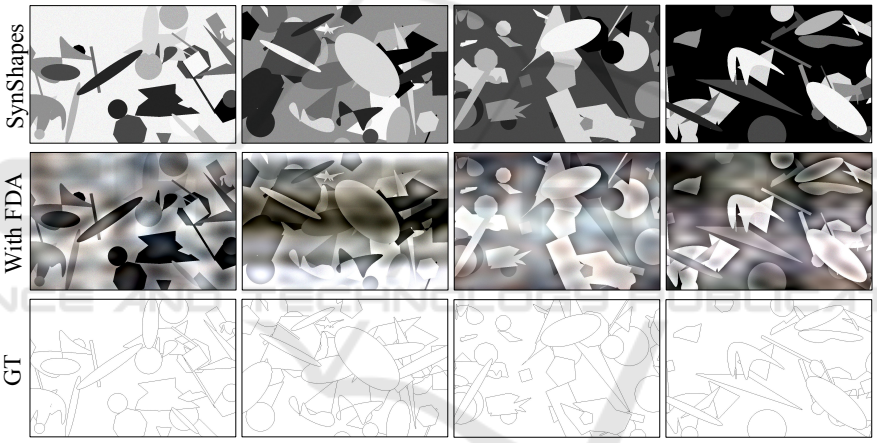

Figure 2: Samples of images from the proposed dataset: (*top*) Synthetic images; (*middle*) Synthetic images with Fourier Domain Adaptation; (*bottom*) Ground Truth edge maps.

fort devoted to densely annotating nearly every edge in BIPED, some images still contain unannotated boundaries, as illustrated in the zoomed region of Fig. 1 *(bottom)*. These missing annotations ultimately have a negative impact on training quality.

An alternative approach to mitigate the aforementioned issues is the use of synthetic imagery, which enables fully reliable edge generation and provides precise supervision for training future edge detection models. Synthetic data has been successfully leveraged in a wide range of computer vision tasks, including: *(i)* camera calibration (Charco et al., 2021), *(ii)* object detection (Tian et al., 2018), *(iii)* image dehazing (Sappa et al., 2022), and *(iv)* depth map estimation (Antunes et al., 2023), among others.

Inspired by these works, we introduce SynShapes (see Fig. 2), a dataset composed of synthetic images paired with automatically and accurately generated edge maps derived directly from the underlying synthetic geometry. These edge maps serve as precise ground-truth annotations, free from the ambiguity and incompleteness inherent to manual labeling. To the best of our knowledge, SynShapes is the first synthetic dataset specifically designed for edge detection that provides exact, pixel-level ground truth.

3 PROPOSED APPROACH

This section describes the proposed pipeline for generating the synthetic image dataset along with its corresponding ground truth edge maps. In addition, we

present the strategy adopted to adapt the synthetic images to the real domain, as well as the edge detection model selected to evaluate the usability of the proposed dataset.

3.1 Synthetic Image Generation

To address the challenge of edge annotation, we propose the SynShapes dataset (Fig. 2), which is generated using a Python-based pipeline with fixed random seeds to ensure full reproducibility. The images are composed of geometric shapes, with randomness applied to selected parameters, including color, quantity, size, and spatial position. Fixed parameters include the image resolution (1280×720), the total number of images (125), the edge thickness (1 pixel), and the noise type (speckle). The set of geometric primitives comprises ellipses, circles, rectangles, triangles, regular polygons, and irregular polygons.

Regarding the number of shapes, a random range of [4–10] instances is sampled for each geometric category; consequently, each image contains between 24 and 60 shapes in total. Each shape is rendered onto a pixel-wise label map and assigned a unique identifier. Ground truth edges are then derived by marking pixels whose labels differ from those of any of their eight-connected neighbors (horizontal, vertical, and diagonal). This procedure naturally accounts for occlusions, as shapes drawn later may partially or fully overlap previously rendered ones; the complete dataset generation process requires less than two minutes.

For the size parameter, distinct ranges are defined for each geometric primitive to reflect their inherent geometric properties, in contrast to the quantity parameter, for which the same range is applied across all shape types. Additionally, certain shapes incorporate extra sources of variability: ellipses and triangles may be randomly rendered in narrower or more elongated forms; rectangles may have either their width or height reduced; and irregular polygons may be generated with either straight or curved boundaries.

During experimentation, selecting the color palette proved challenging due to the combinatorial nature of color choices, where even the addition or removal of a single color could significantly affect performance metrics. To mitigate this variability, grayscale colors (identical values across RGB channels) were adopted, resulting in a final palette of eight intensity levels: (0,0,0), (36,36,36), (73,73,73), (109,109,109), (146,146,146), (182,182,182), (219,219,219), and (255,255,255).

Regarding noise modeling, the skimage library is employed, specifically using the speckle noise model with a randomly sampled variance in the range of [0.02–0.10]. Additional experiments were conducted using Gaussian, Poisson, and salt-and-pepper noise; however, these alternatives yielded inferior performance and were therefore discarded.

Finally, it is worth noting that the selected parameter configuration results from more than 100 experimental trials exploring different combinations of shape types, sizes, quantities, noise models, and color palettes. The configuration described above consistently produced the best performance during both training and evaluation.

3.2 Synthetic-to-Real Adaptation

Once the dataset is generated, the Fourier Domain Adaptation (FDA) method (Yang and Soatto, 2020) is applied to reduce the synthetic-to-real domain gap. Following the guidelines in (Yang and Soatto, 2020), a β value of 0.01 was adopted, using the BIPED dataset (Soria et al., 2023b) as the target domain. Fig. 2 *(middle row)* illustrates representative samples of the images obtained after applying the FDA method. Our implementation also incorporates portions of the public code released by the authors.

3.3 Training Strategy

In order to evaluate the usefulness of the proposed dataset, the TEED edge detector (Soria et al., 2023a) was selected and trained on: Barcelona Images for Perceptual Edge Detection, **BIPED** (Soria et al., 2023b); Berkeley Segmentation Dataset, **BSDS500** (Arbelaez et al., 2011); Multicue Dataset for Boundary Detection, **MDBD** (Mély et al., 2016); and **SynShapes-with FDA-datasets** (see Fig. 1 and Fig. 2), enabling a performance comparison among them. In this case, modifications were made to the code released by the authors of TEED (Soria et al., 2023a), with the main modification being the addition of a reproducibility seed.

TEED has been chosen due to its strong balance between efficiency and robustness in RGB edge detection tasks. The model can be trained on the complete BIPED dataset in a very short time while still delivering results that match or surpass current state-of-the-art methods in both quantitative metrics and visual quality. A particularly noteworthy aspect is its ability to generalize: despite having a lightweight architecture, TEED reliably captures fine-grained and semantically relevant edges that are frequently missed by substantially larger models with far higher parameter counts.

TEED is composed of three modules: a main fea-

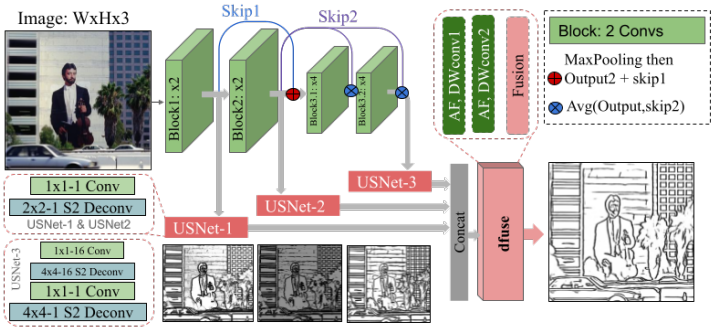

Figure 3: TEED architecture from (Soria et al., 2023a).

ture extractor (encoder), USNet, and the *dfuse* module (see Fig. 3). The main feature extractor, also referred to as the encoding network, extracts multi-scale feature maps from the input image, which are subsequently fed into USNet—the decoding module where edge supervision is applied at each encoding stage. Each USNet produces a preliminary edge-map corresponding to a specific feature scale. These intermediate edge maps are then aggregated and adaptively modulated by the final edge prediction module, denoted as *dfuse*. Overall, this lightweight architecture generates thin, sharp, and noise-free edge-maps, largely, due to the *dfuse* module, which effectively refines the preliminary edge responses provided by USNet in the edge space. Notably, these edge representations are transformed from the original input image features extracted by the encoding network. The training of the entire architecture is regularized using two loss functions, applied at different levels of edge-map refinement—USNet and dfuse.

4 EXPERIMENTAL RESULTS AND COMPARISONS

To ensure a more reliable and reproducible analysis of each dataset's behavior, three independent training runs were performed per dataset using different random seeds and the reported results correspond to their average. This protocol was adopted after observing non-negligible variability in the evaluation metrics when TEED was exposed to different permutations of the training images. In particular, this variability persists despite the use of predefined and standardized training-testing splits provided by the selected datasets, highlighting the sensitivity of the training process to initialization and data ordering. Additionally, no data augmentation techniques were applied to any of the datasets, and all reported results were obtained using the raw unaltered datasets as originally provided. Furthermore, to guarantee fair and comparable experimental conditions across all datasets, a single, fixed set of hyperparameters was employed for every training session. The complete configuration is reported in Table 1.

4.1 Evaluation Metrics

Three evaluation metrics commonly adopted in the edge detection community are employed: fixed contour threshold (ODS), per-image best threshold (OIS), and average precision (AP) (Poma et al., 2020); where only the F-score (F) of ODS and OIS is considered:

$$F = \frac{2PR}{P+R}, \tag{1}$$

where P denotes Precision and R denotes Recall, defined as:

$$P = \frac{TP}{TP+FP}, \tag{2}$$

$$R = \frac{TP}{TP+FN}, \tag{3}$$

Table 1: Optimized hyperparameters used on the TEED model.

Hyperparameter	Value
Epochs	100
Batch Size	4
Learning Rate (LR)	1×10^{-3}
LR Adjustment	$[6 \times 10^{-4}, 1 \times 10^{-4}]$
LR Adjustment Epochs	$[30, 70]$
Weight Decay	8×10^{-3}
Num. Workers	8
Input Size	300×300
Optimizer	AdamW

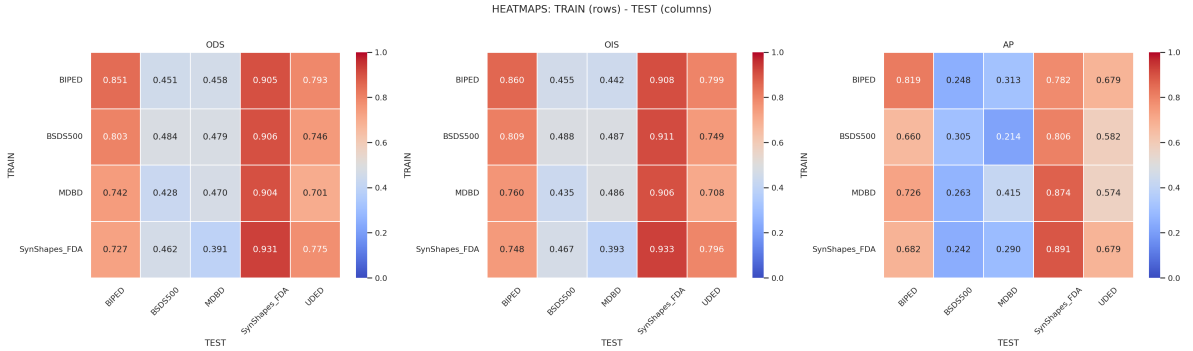

Figure 4: Average metrics results (ODS, OIS and AP) obtained for each trained dataset.

here, TP, FP and FN represent: True Positives, False Positives, and False Negatives, respectively.

For the evaluation and computation of the ODS, OIS, and AP metrics, a modified version of the evaluator implemented in (Li et al., 2025b) was used, which is itself a Python adaptation of the original evaluator used in (Dollár and Zitnick, 2015). The new modification consists of accelerating the evaluation process without significantly altering the resulting metric values.

4.2 Quantitative Results

As mentioned above, multiple training runs with different random seeds are conducted for each dataset, and the resulting metrics are averaged. The TEED model trained on a given dataset is then evaluated on the remaining datasets to assess both the usefulness of that dataset and the model's generalization capability. The performance is summarized using heatmap visualizations, which enable a clearer and more intuitive comparison of metric behavior across all combinations of training and test datasets (see Fig. 4). In these heatmaps, deeper red tones indicate higher metric values, whereas deeper blue tones indicate lower performance. In addition to evaluations on the training datasets themselves, we include UDED, a dataset specifically designed to assess generalization in edge detection (Soria et al., 2023a).

From the heatmap analysis (Fig. 4), BSDS500 and MDBD emerge as particularly challenging evaluation benchmarks. This difficulty is likely attributable to the presence of multiple annotator-derived edge maps within the same image, which leads the evaluator to penalize discrepancies between predicted edges and the aggregated ground truth.

Overall, the model trained on BIPED achieves the strongest performance. It attains the highest metric values when evaluated on both BIPED itself and UDED, and it also exhibits robust generalization when tested on BSDS500, MDBD, and SynShapes with FDA, relative to models trained on the other datasets.

Focusing on the model trained using our synthetic dataset, SynShapes, its performance is competitive with that of models trained on real-world data. Notably, when evaluated on BSDS500, it achieves higher ODS and OIS scores than models trained on MDBD and BIPED. Moreover, on UDED, it outperforms models trained on MDBD and BSDS500, and attains results that are nearly on par with those of the model trained on BIPED.

4.3 Qualitative Results

Finally, Figs. 5–9 illustrate representative edge predictions produced by TEED when trained on the different datasets. These qualitative results show that SynShapes enables strong generalization and effective edge detection on real-world images, as internal details of trees and buildings can be observed in most cases; details that do not appear or do not appear completely in other trained models. As a consequence, it is important to note that, when TEED is trained on SynShapes, the evaluation metrics may penalize some predictions due to missing edges in the ground-truth annotations of real datasets. This contrasts with the results shown in Fig. 9, where all datasets achieve high metric values. Although the edge detection task in this synthetic scenario is less complex than in real-world scenes, none of the evaluated models are penalized by incomplete annotations, since the ground truth is exhaustive by construction.

5 CONCLUSIONS

In this work, we introduce SynShapes, which, to the best of our knowledge, is the first dataset composed of synthetic images paired with exact edge map annotations. SynShapes eliminates the need for time-

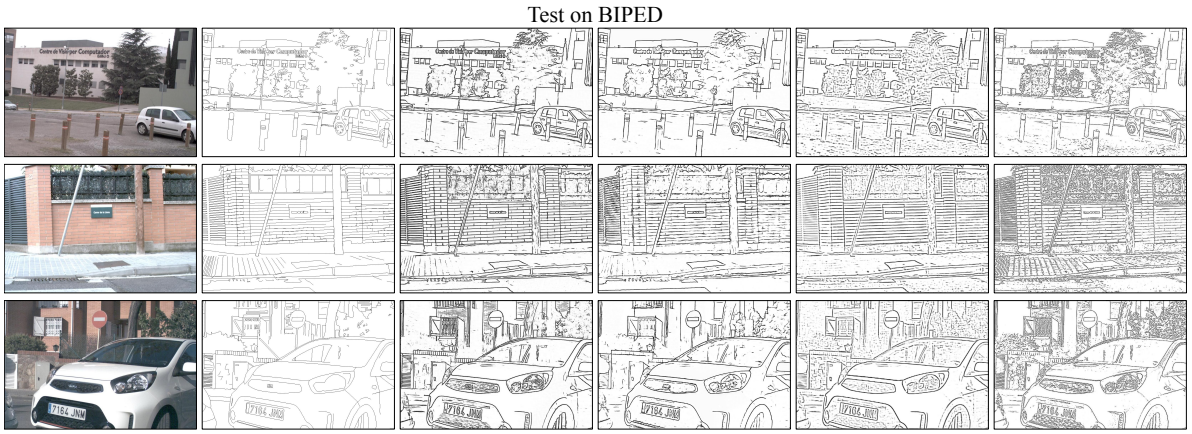

Figure 5: Samples predictions of TEED trained on different datasets and tested on BIPED.

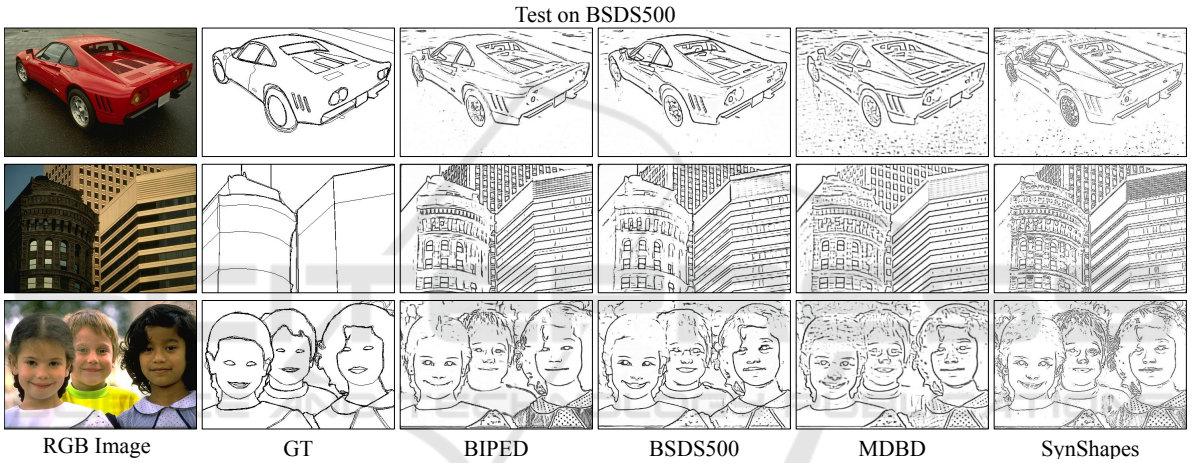

Figure 6: Samples predictions of TEED trained on different datasets and tested on BSDS500.

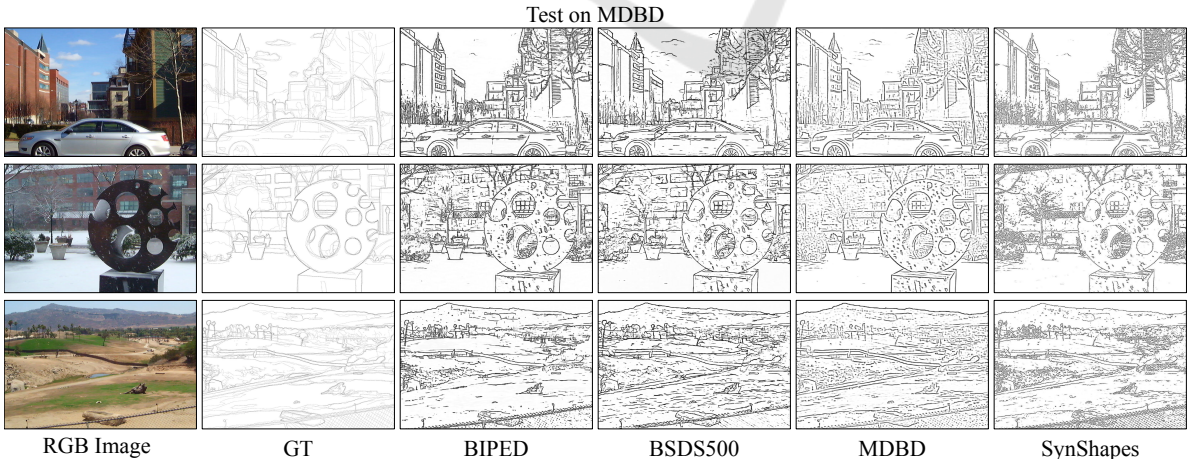

Figure 7: Samples predictions of TEED trained on different datasets and tested on MDBD.

consuming manual labeling while guaranteeing precise and unambiguous ground truth, free from annotator disagreement and missing-edge artifacts–two factors that critically affect the quality of edge detector training. To further bridge the gap between synthetic and real data, we incorporate a fast and ef-

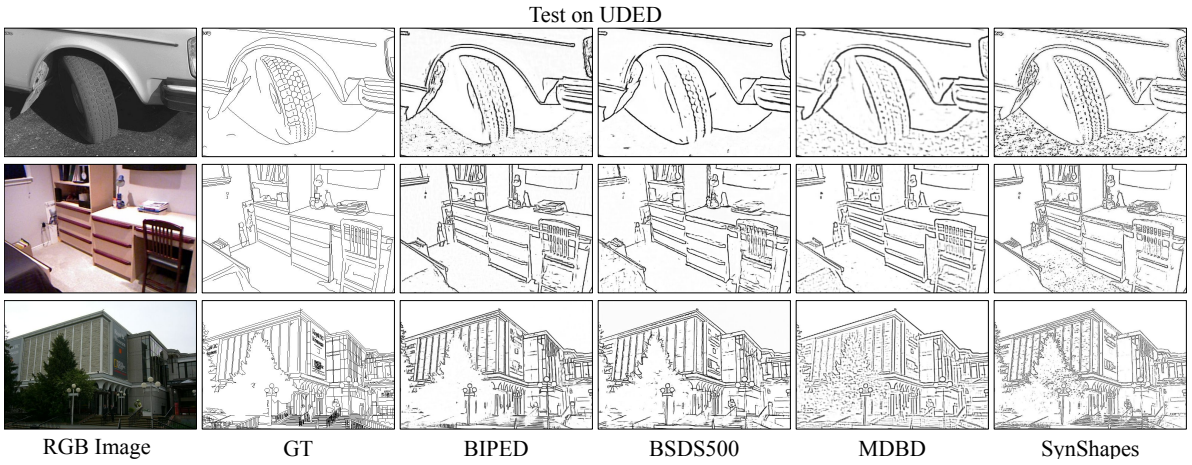

Figure 8: Samples predictions of TEED trained on different datasets and tested on UDED.

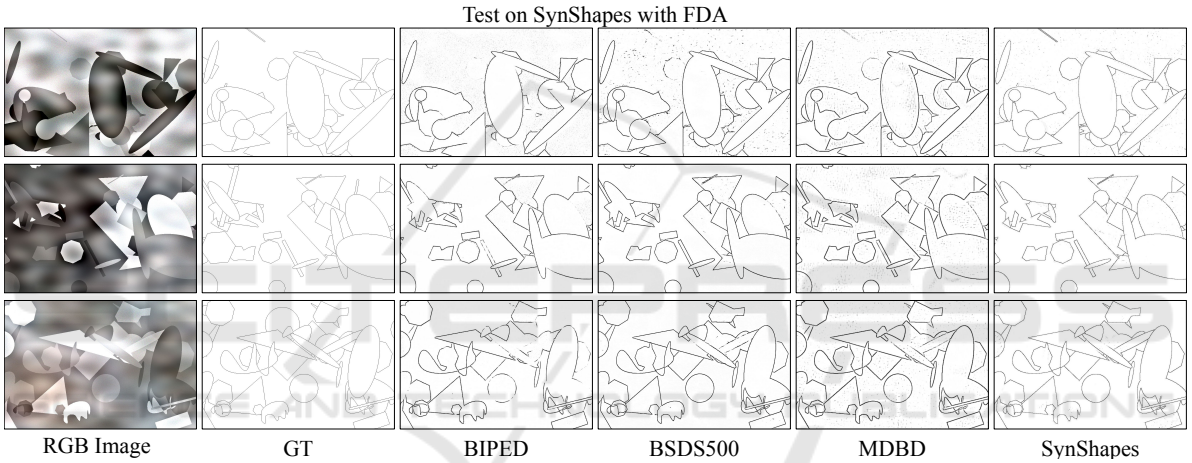

Figure 9: Samples predictions of TEED trained on different datasets and tested on SynShapes.

fective domain adaptation strategy based on Fourier Domain Adaptation. As a result, models trained using SynShapes achieve performance comparable to that obtained with real-world datasets, demonstrating that competitive edge detection on natural images can be learned from accurately annotated synthetic image–edge pairs.

As future work, we plan to evaluate fine-tuning strategies, such as training for a large number of epochs on the synthetic dataset followed by continued training on real-world datasets. Additionally, exploring alternative parameter configurations for synthetic dataset generation to further improve performance, as well as evaluating other domain adaptation techniques proposed in the literature, constitutes a promising direction for future research.

ACKNOWLEDGEMENTS

This work was supported in part by the Air Force Office of Scientific Research Under Award FA9550-24-1-0206; in part by the ESPOL project “Advancing Camouflaged Object Detection with a cost-effective Cross-Spectral vision system (ACODCS)” (CIDIS-003-2024); in part by Grant PID2024-162815NB-I00 funded by MICIU/AEI/ 10.13039/501100011033 and by ERDF/EU, and Grant PID2021-128945NB-I00 funded by MICIU/AEI/ 10.13039/501100011033 and by ERDF/EU. The authors acknowledge the support of the Generalitat de Catalunya CERCA Program to CVC’s general activities, and the Departament de Recerca i Universitats from Generalitat de Catalunya to the SGR Research Group 2021 MACO (reference 2021 SGR 01499).

REFERENCES

Antunes, M., Bergasa, L. M., Montiel-Marín, S., Sánchez-García, F., Pardo-Decimavilla, P., and Revenga, P. (2023). Exploring domain adaptation with depth-based 3d object detection in carla simulator. In *Iberian Robotics conference*, pages 105–117. Springer.

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916.

Canny, J. (2009). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698.

Cetinkaya, B., Kalkan, S., and Akbas, E. (2024). Ranked: Addressing imbalance and uncertainty in edge detection using ranking-based losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3239–3249.

Charco, J. L., Sappa, A. D., Vintimilla, B. X., and Velesaca, H. O. (2021). Camera pose estimation in multi-view environments: From virtual scenarios to the real world. *Image and Vision Computing*, 110:104182.

Dollár, P. and Zitnick, C. L. (2015). Fast edge detection using structured forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1558–1570.

He, J., Zhang, S., Yang, M., Shan, Y., and Huang, T. (2019). Bi-directional cascade network for perceptual edge detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3828–3837.

Ji, S., Yuan, X., Bao, J., and Liu, T. (2025). Led-net: A lightweight edge detection network. *Pattern Recognition Letters*, 187:56–62.

Li, Y., Poma, X. S., Bai, Y., Xiao, Q., Yang, C., Li, G., and Li, Z. (2025a). Edmb: Edge detector with mamba. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7682–7691. IEEE.

Li, Y., Poma, X. S., Chu, T., Xi, Y., Li, G., Yang, C., Xiao, Q., Bai, Y., and Li, Z. (2025b). Pidinext: Lightweight parallel pixel difference networks for edge detection. *Multimedia Tools and Applications*, 84(26):30757–30780.

Liu, Y., Cheng, M.-M., Hu, X., Bian, J.-W., Zhang, L., Bai, X., and Tang, J. (2019). Richer convolutional features for edge detection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 41(08):1939–1946.

Liufu, X., Tan, C., Lin, X., Qi, Y., Li, J., and Hu, J.-F. (2025). Sauge: Taming sam for uncertainty-aligned multi-granularity edge detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 5766–5774.

Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167):187–217.

Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423.

Mély, D. A., Kim, J., McGill, M., Guo, Y., and Serre, T. (2016). A systematic comparison between visual cues for boundary detection. *Vision research*, 120.

Poma, X. S., Riba, E., and Sappa, A. (2020). Dense extreme inception network: Towards a robust cnn model for edge detection. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1923–1932.

Pu, M., Huang, Y., Liu, Y., Guan, Q., and Ling, H. (2022). Edter: Edge detection with transformer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1402–1412.

Sappa, A. D., Suárez, P. L., Velesaca, H. O., and Carpio, D. (2022). Domain adaptation in image dehazing: Exploring the usage of images from virtual scenarios. In *Int. Conf. on Computer Graphics, Visualization, Computer Vision and Image Processing*, pages 85–92.

Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. (2012). Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer.

Soria, X., Li, Y., Rouhani, M., and Sappa, A. D. (2023a). Tiny and efficient model for the edge detection generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1364–1373.

Soria, X., Pomboza-Junez, G., and Sappa, A. D. (2022). Ldc: Lightweight dense cnn for edge detection. *IEEE Access*, 10:68281–68290.

Soria, X., Sappa, A., Humanante, P., and Akbarinia, A. (2023b). Dense extreme inception network for edge detection. *Pattern Recognition*, 139:109461.

Tian, Y., Li, X., Wang, K., and Wang, F.-Y. (2018). Training and testing object detectors with virtual images. *IEEE/CAA Journal of Automatica Sinica*, 5(2):539–546.

Xie, S. and Tu, Z. (2015). Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Cision*, pages 1395–1403.

Xu, J., Xiong, Z., and Bhattacharyya, S. P. (2022). Pidnet: A real-time semantic segmentation network inspired from pid controller. *arXiv preprint arXiv:2206.02066*.

Yang, Y. and Soatto, S. (2020). Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095.

Ye, Y., Xu, K., Huang, Y., Yi, R., and Cai, Z. (2024). Diffusionedge: Diffusion probabilistic model for crisp edge detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 6675–6683.

Zhou, C., Huang, Y., Pu, M., Guan, Q., Deng, R., and Ling, H. (2024). Muge: Multiple granularity edge detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 25952–25962.

Zhou, C., Huang, Y., Pu, M., Guan, Q., Huang, L., and Ling, H. (2023). The treasure beneath multiple annotations: An uncertainty-aware edge detector. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 15507–15517.