JURNAL INFOTEL
VOL. 18, NO. 3, AUGUST 2026, PP. 357–370. DOI:10.20895/INFOTEL.V18I3.11937

RESEARCH ARTICLE

Exploring DB-LETNet for Adversarial Patch Detection: A Comparative Study of Lightweight Segmentation Architectures on Face Images

Ahmad Taufiq Nur Rohman[1], Kurniawan Nur Ramadhani[2,*], and Mahmud Dwi Sulistiyo[3]

[1,2,3]School of Computing, Telkom University, Bandung, Indonesia

*Corresponding email: kurniawannr@telkomuniversity.ac.id

Received: August 02, 2026; Revised: August 13, 2026; Accepted: August 31, 2026.

Abstract: Face recognition systems are vulnerable to physical adversarial patches, while many segmentation-based detectors remain computationally heavy. This study investigates which lightweight segmentation design choices are effective for adversarial patch detection on face images and derives DB-LETNet V8 as a dual-branch candidate architecture. Eight LETNet-derived variants are explored by isolating block count, channel width, convolution type, decoder structure, and branch fusion. DB-LETNet V8 is then evaluated in a six-model benchmark against five baseline segmentation architectures under identical training conditions. The benchmark uses digitally composited PGD-generated patches on CelebA-HQ faces at 112×112 resolution. The comparative evaluation is repeated across three random seeds on in-distribution and out-of-distribution test sets. DB-LETNet V8 has 0.385M parameters and achieves 0.891±0.019 IoU on in-distribution tests while being 80 times smaller than U-Net. Under out-of-distribution shift, it reaches 0.664±0.043 IoU. It ranks second after U-Net and obtains the highest mean IoU among lightweight models, although its margins over LETNet Baseline are not significant in paired seed-level tests. Dual-branch semantic-edge fusion gives the largest design gain of +4.1% IoU, whereas ghost convolutions reduce IoU by 24 percentage points. These results indicate that DB-LETNet V8 provides the most favorable mean lightweight accuracy-to-size trade-off in this benchmark, although all models remain sensitive to cross-attack distribution shift.

Keywords: adversarial patch detection, dual-branch segmentation, face recognition security, lightweight architecture, patch localization

© Jurnal Infotel ISSN: 2085-3688; e-ISSN: 2460-0997

1 Introduction

Face recognition has become part of everyday security infrastructure, including mobile authentication, border control, and access management [1], [2]. Surveys of deep face recognition show the scale and diversity of this use [3]. This broad use also exposes a subtle physical vulnerability: an attacker can alter the visual input without accessing the recognition system itself. One practical form of this threat is an adversarial patch. It is a localized visual pattern that is optimized or designed to mislead a recognition model when it appears in the camera input [4]. In physical settings, such a patch can be printed and placed on the face area to induce evasion or impersonation. Sharif et al. first demonstrated adversarial eyeglass frames as a practical physical attack [4]. Their later work formalized a broader optimization framework for adversarial objectives [5]. Komkov and Petiushko extended the attack surface to hat-based patches [6]. Xiao et al. used generative models to improve patch transferability across face recognition systems [7]. Wei et al. studied sticker-based physical attacks [8]. Zolfi et al. examined fabric-mask attacks against face recognition models [9]. Liu et al. further showed that compact-printer black-box attacks can produce effective physical impersonation patches [10]. A recent survey confirms that these attacks vary by carrier medium, generation method, victim model, and capture condition [11]. This diversity makes detection difficult because the visual pattern seen in use may differ from the pattern seen during training.

These physical-attack studies motivate patch localization before recognition. However, they do not resolve which lightweight segmentation architecture is most suitable when the defense is framed as a pixel-level localization module for face images. This paper therefore studies patch detection as binary segmentation rather than as a complete recognition-recovery pipeline. In this component-level setting, the localization model should identify patch pixels accurately while limiting parameter count, MACs, and measured desktop inference latency. Heavy encoder-decoder models such as U-Net [12] provide strong localization but are difficult to justify when model size is constrained. In contrast, ENet and BiSeNet reduce parameter count for efficient segmentation [13], [14]. LETNet also targets lightweight segmentation, but these models were designed for natural scene segmentation or edge detection rather than adversarial textures [15].

The transfer is non-trivial because adversarial patches have optimized high-frequency textures, sharp compositing boundaries, and geometry that changes with the attack and victim model. A design that preserves road boundaries or medical organ contours may not preserve adversarial texture boundaries. No prior work has systematically isolated which lightweight architectural choices work for this specific face-patch segmentation task. This study addresses that gap through three contributions. First, it performs progressive exploration of eight LETNet-derived variants to isolate depth, width, convolution type, decoder structure, and dual-branch fusion. Second, it benchmarks six architectures in total, consisting of DB-LETNet V8 and five baselines, under identical training and evaluation settings across in-distribution and out-of-distribution attacks. Third, it derives empirical guidelines for lightweight architecture selection, emphasizing accuracy-to-size trade-offs rather than hardware-specific deployment claims.

2 Related Work

Existing defenses address related but different requirements. PatchGuard constrains receptive fields and masking operations to provide certified robustness against localized adversarial patches [16]. PatchCleanser extends certified patch defense by using double masking to remove the influence of a single adversarial patch on image classification [17]. These certified approaches provide formal guaranties. However, their assumptions and computational structure are not designed for lightweight pixel-level face segmentation. Liang et al. detect adversarial examples through adaptive noise reduction, which is useful for input-level detection but does not produce a patch mask [18]. Agarwal et al. study image transformation as a defense against adversarial perturbations. Their output remains an altered image rather than an explicit localization map [19]. Local Gradient Smoothing suppresses localized adversarial effects by smoothing gradient patterns, but it also does not identify patch pixels for selective removal [20].

Localization and segmentation defenses are closer to this paper's objective. Segment and Complete detects adversarial patches and completes the corrupted region for object detection [21]. Its target task is object detection rather than compact face-specific segmentation. Jedi localizes adversarial patches through entropy-based analysis and then removes suspicious regions [22]. This differs from a learned lightweight segmentation model. DIFFender uses diffusion-based purification against patch attacks, but diffusion inference is heavier than the segmentation module studied here [23]. PAD uses a patch-agnostic defense strategy that can operate without assuming a specific patch pattern [24]. Its design is not centered on sub-1M segmentation architectures. Segment Anything provides a strong general-purpose segmentation foundation [25]. Its scale places it in a different model-size class from the lightweight models considered in this study. RADAP is the closest face-oriented defense because it studies robust and adaptive protection against diverse adversarial patches on face recognition systems [26]. In contrast, this work isolates the segmentation component. It asks which lightweight architectures most effectively localize patch pixels under controlled training and evaluation conditions.

Lightweight segmentation offers several candidate design patterns but no direct answer for face-patch detection. ENet reduces cost through early downsampling, which may weaken boundary detail [13]. BiSeNet separates spatial and contextual paths for real-time segmentation [14]. BiSeNet V2 improves this idea with guided aggregation between detail and semantic branches [27]. MobileNetV3 provides efficient mobile backbones that can be paired with segmentation decoders [28]. DeepLabV3 uses atrous convolution to expand contextual coverage during semantic segmentation [29]. GhostNet reduces computation by generating cheap feature maps from intrinsic representations [30]. This assumption may not preserve adversarial texture evidence. LETNet is suitable for controlled studies because it uses Dense Attention Blocks and a modular decoder [15]. These components allow targeted substitutions within a sub-1M parameter budget. These designs optimize efficiency in general vision benchmarks, but adversarial patch detection depends on small texture regions and sharp local boundaries.

3 Research Method

3.1 Problem formulation and base architecture

The task is binary segmentation of adversarial patches on face images. Given an input of size $3 \times 112 \times 112$, the network produces a single-channel mask thresholded at 0.5. The resolution matches standard face recognition inputs [1], [31]. The exploration starts from LETNet [15], which uses Dense Attention Blocks (DAB) with dense connections and channel attention across four resolution levels, at 0.763M parameters. LETNet was chosen for its modular design, which enables controlled component substitution within a sub-1M parameter scale.

3.2 Progressive design exploration

Eight architectural variants are organized into four phases, each modifying exactly one design dimension. Table 1 summarizes the variants.

Table 1: Progressive design exploration summary

Variant	Focus	Change	Params
Baseline	Base	LETNet, 128ch, 3 DAB/stage	0.763M
V1	Depth	2 DAB/stage	0.483M
V2	Width	64 channels	0.251M
V3	Conv	GhostConv	0.672M
V4	Conv	Inverted residual	2.083M
V5	Decoder	Bilinear upsample	0.746M
V6	Width	96 channels	0.368M
V7	Structure	Dual branch + gate	0.381M
V8	Structure	V7 + RegionContext	0.385M

The variants test four design questions. V1 and V2 compare depth and width reduction. V3 and V4 test whether mobile-efficient convolution blocks transfer to adversarial texture segmentation. V5 and V6 adjust decoder simplicity and intermediate channel width. V7 introduces the main structural change: a semantic decoder branch for contextual patch cues and an edge branch for artificial boundary cues, combined through gated fusion. V8 adds a RegionContext module with parallel dilated convolutions at rates 1, 2, 4, and 8 to capture patch size variation. Figure 1 illustrates the complete DB-LETNet V8 architecture.

3.3 Loss formulation

The progressive exploration trains single-branch variants with BACL (Boundary-Aligned Composite Loss), which combines BCE, Dice Loss [32], and Sobel Gradient Alignment Loss to balance per-pixel, region, and boundary supervision. Dual-branch variants use BGCAGLoss, which applies full-mask supervision to the fused output and Branch A, while Branch B is supervised with a boundary-focused target derived from the patch contour. For the comparative evaluation, Focal Loss [33] ($\gamma = 2$, $\alpha = 0.75$) replaces BCE to handle class imbalance, while Dice Loss and InteriorRecallLoss are retained:

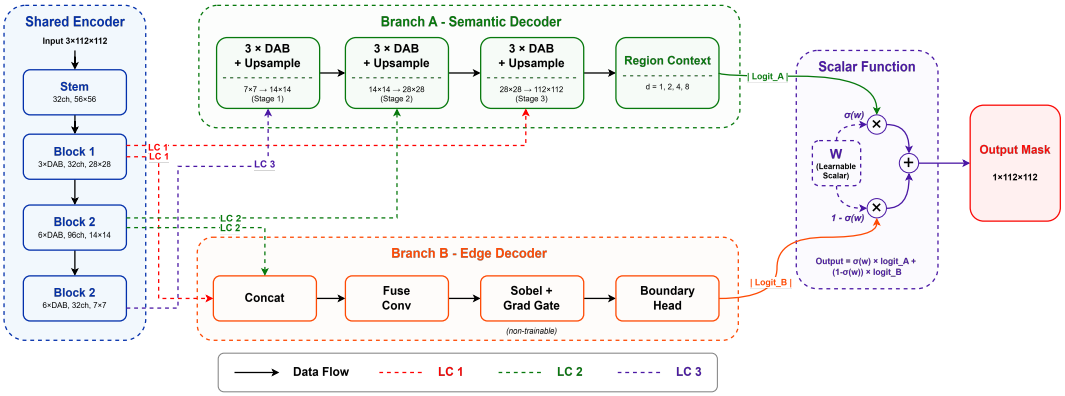

Figure 1: Architecture of DB-LETNet V8 showing the shared encoder, dual-branch decoder with semantic path (Branch A) and edge path (Branch B), gated fusion mechanism, and RegionContext module.

$$L_{Interior} = 1 - \frac{\sum_{i=1}^{N} w_i \cdot y_i \cdot p_i + \epsilon}{\sum_{i=1}^{N} w_i \cdot y_i + \epsilon} \tag{1}$$

where $w_i = 1 + d_i / D_{max}$ weights interior pixels by morphological distance from the nearest boundary, with $D_{max} = 5$. The comparative loss is $L_{comp} = L_{Focal} + L_{Dice} + 0.3 \cdot L_{Interior}$. For dual-branch training, auxiliary supervision prevents branch dormancy:

$$L_{total} = L_{comp}(\hat{y}_{fused}) + 0.4 \cdot L_{branchA}(\hat{y}_A) + 0.3 \cdot L_{branchB}(\hat{y}_B) \tag{2}$$

where $L_{branchA} = L_{Focal} + L_{Dice}$ uses the full mask and $L_{branchB}$ uses focal supervision on a morphologically extracted boundary ring. This separation encourages Branch A to cover patch regions, and Branch B to specialize in edge evidence.

3.4 Dataset and training protocol

Training and validation use 25,000 CelebA-HQ [34] face images at 112×112 with digitally composited PGD-generated [35] patches against MobileFaceNet [31]. This split contains 20,000 training images and 5,000 validation images, with 55% clean and 45% patched samples. The patched samples span four geometries: blob, facemask, glasses, and triangle. They also span three appearances: noise, PGD-optimized texture, and solid color. This combination produces twelve configurations. Figure 2 shows representative samples. Evaluation uses two separate test sets of 7,323 images each. The in-distribution (ID) test uses Auto-PGD [36] against MobileFaceNet. The out-of-distribution (OOD) test uses RSTAM [37] against IResNet50 [38]. In the OOD setting, both the attack algorithm and victim model differ from training.

The comparative benchmark contains six models in total. These are DB-LETNet V8 and five baselines: U-Net, LETNet Baseline, ENet, BiSeNet V1, and MobileNetV3+DeepLabV3. All models train with Adam optimizer using weight decay 10^{-4}, learning rate 3×10^{-5}, batch size 32, and 30 epochs with cosine annealing to 10^{-7} and gradient clipping at max

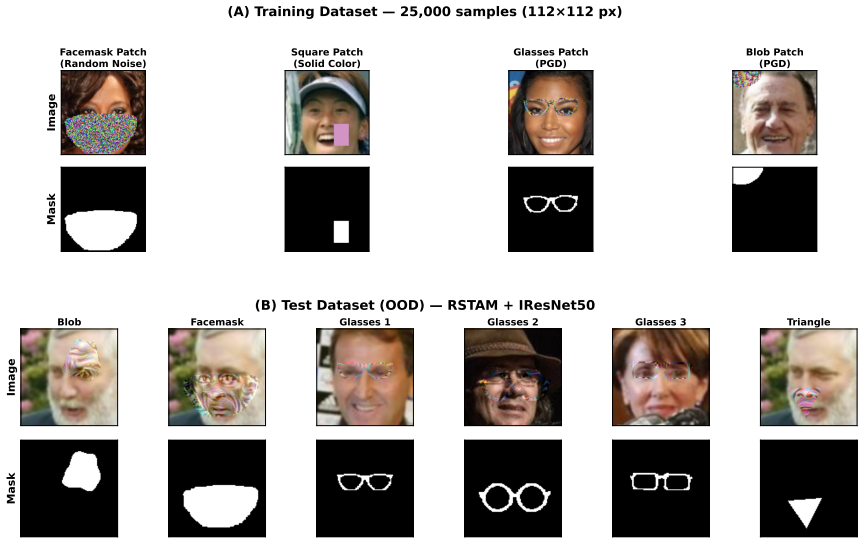

Figure 2: Dataset overview showing clean faces and patched samples across four geometric shapes with three appearance types.

norm 1.0. The progressive exploration uses standard augmentation including random horizontal flip, color jitter, and rotation up to 10 degrees. The comparative evaluation replaces these with TextureRandomPatch augmentation (probability 0.5) combined with random horizontal flip, which randomizes the texture inside patch regions during training to prevent models from memorizing specific attack appearances. No model uses pretrained weights to isolate architectural effects from transfer learning advantages. The progressive design exploration uses seed 42, while the full comparative evaluation is repeated with seeds 42, 123, and 456. All comparative tables report the mean and sample standard deviation across these three runs.

3.5 Evaluation metrics

Performance is measured by IoU, Dice/F1, Precision, Recall, and MAE. IoU serves as the primary metric:

$$\text{IoU} = \frac{|P \cap G|}{|P \cup G|} = \frac{TP}{TP + FP + FN} \tag{3}$$

where P is the predicted mask and G is the ground truth. Dice/F1 complements IoU as an overlap score. Precision and Recall separate false positive masking from missed patch pixels, while MAE measures average per-pixel error before thresholding. For statistical robustness, the comparative evaluation treats the random seed as the paired unit. DB-LETNet V8 is compared with each baseline using the IoU obtained from the same seed on the same fixed test set. The analysis reports the paired mean difference, 95% confidence interval, and two-sided paired t-test p-value. Because only three seeds are available, the statistical test is used as a robustness check rather than the sole basis for model ranking.

4 Results

4.1 Progressive design results

Table 2 presents the progressive exploration results on the in-distribution test set. Single-branch variants (Baseline through V6) use the BACL loss combining BCE, Dice, and SGAL. V7 introduces BGCAGLoss that applies BACL to both the fused output and Branch A while adding BoundaryFocalLoss on Branch B. V8 extends this with an InteriorRecall term.

Table 2: Progressive design exploration results on the in-distribution test set from a single seed (42)

Variant	IoU	F1	Prec.	Rec.	Params	MACs
Baseline	0.8701	0.9282	0.9237	0.9328	0.763M	0.322G
V1	0.8901	0.9397	0.9326	0.9468	0.483M	0.271G
V2	0.8764	0.9314	0.9268	0.9359	0.251M	0.230G
V3	0.6477	0.7822	0.8307	0.7390	0.672M	0.300G
V4	0.8613	0.9238	0.9238	0.9238	2.083M	0.610G
V5	0.8350	0.9059	0.9156	0.8965	0.746M	0.312G
V6	0.8862	0.9376	0.9320	0.9432	0.368M	0.251G
V7	0.9272	0.9622	0.9606	0.9639	0.381M	0.271G
V8	0.9530	0.9759	0.9743	0.9776	0.385M	0.343G

The exploration shows a clear design hierarchy. Dual-branch fusion gives the largest gain, followed by multi-scale context and moderate channel-width tuning. Reducing blocks per stage improves the baseline, but halving channels hurts accuracy until V6 restores capacity at 96 channels. Mobile-optimized convolution does not transfer well: Ghost modules drop IoU to 0.648, and inverted residual blocks remain below the original DAB design.

4.2 Branch ablation

Table 3 isolates the contribution of each branch through controlled ablation.

Table 3: Branch ablation results from a single seed (42), showing individual and combined branch performance

Configuration	IoU	F1	Prec.	Rec.	Params
Branch A only (Semantic + RCB)	0.860	0.921	0.906	0.939	0.377M
Branch B only (Edge + Sobel-gate)	0.857	0.919	0.886	0.962	0.352M
Dual (A + B + Gated Fusion)	0.882	0.934	0.916	0.957	0.385M

The branches reach similar IoU in isolation but fail differently. The semantic branch is more conservative and precise, while the edge branch provides higher recall by responding to sharp patch boundaries. Figure 3 visualizes this specialization using auxiliary responses from the same full V8 checkpoint. Gated fusion benefits from this complementarity by using edge cues for coverage and semantic cues to suppress false positives, yielding higher IoU than either branch alone.

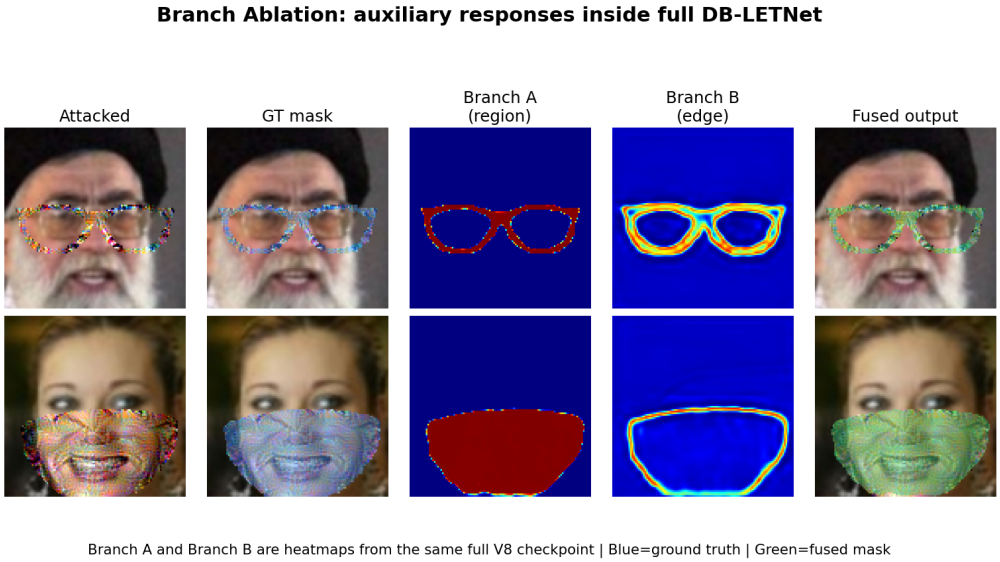

Figure 3: Visual branch specialization in DB-LETNet V8. Branch A responds to patch regions, Branch B emphasizes boundary evidence, and the fused output combines both cues into a final segmentation mask.

4.3 In-distribution comparison

Table 4 presents the ID comparison under the standardized three-seed protocol. DB-LETNet V8 achieves 0.891±0.019 IoU, lower than its single-seed progressive score because the comparative setting uses a different loss and texture-randomizing augmentation for fair cross-architecture evaluation.

Table 4: In-distribution comparison of all evaluated architectures (mean ± SD, seeds 42, 123, and 456)

Model	Params	IoU	F1	Prec.	Rec.	MAE	Inf. (ms)
U-Net	31.044M	0.980±0.002	0.989±0.001	0.991±0.002	0.988±0.000	0.002±0.000	5.34
DB-LETNet V8	0.385M	0.891±0.019	0.940±0.011	0.938±0.003	0.946±0.018	0.011±0.002	44.73
LETNet Baseline	0.763M	0.865±0.017	0.924±0.011	0.891±0.023	0.966±0.007	0.013±0.002	61.57
ENet	0.356M	0.795±0.027	0.876±0.021	0.876±0.004	0.899±0.027	0.023±0.004	14.46
BiSeNet V1	0.217M	0.729±0.020	0.820±0.013	0.824±0.002	0.828±0.023	0.027±0.003	2.77
MobileNetV3+DLV3	11.020M	0.606±0.006	0.721±0.006	0.758±0.005	0.733±0.013	0.044±0.001	7.75

Model capacity alone does not determine performance. MobileNetV3+DLV3 has 11M parameters but ranks last, while DB-LETNet V8 obtains the highest mean IoU among sub-1M models. Its clearest advantage is precision, reaching 0.938±0.003 compared with 0.891±0.023 for LETNet Baseline. This supports interpreting V8 as a lightweight design that suppresses false positives rather than a statistically dominant IoU model over LETNet Baseline. ENet suffers from rough boundaries, and BiSeNet V1 is fastest but least accurate among the lightweight models.

4.4 Out-of-distribution comparison

Table 5 evaluates distribution shift where both the attack algorithm and victim model differ from training. Table 6 breaks down OOD IoU by patch geometry.

Table 5: Out-of-distribution comparison of all evaluated architectures (mean ± SD, seeds 42, 123, and 456)

Model	Params	IoU	F1	Prec.	Rec.	MAE	Inf. (ms)
U-Net	31.044M	0.782±0.045	0.827±0.035	0.903±0.001	0.795±0.045	0.039±0.010	5.22
DB-LETNet V8	0.385M	0.664±0.043	0.745±0.033	0.857±0.001	0.712±0.045	0.048±0.006	57.96
LETNet Baseline	0.763M	0.629±0.054	0.715±0.050	0.813±0.018	0.723±0.060	0.058±0.015	58.55
ENet	0.356M	0.554±0.049	0.643±0.047	0.819±0.002	0.635±0.049	0.064±0.006	15.01
MobileNetV3+DLV3	11.020M	0.392±0.036	0.481±0.040	0.763±0.012	0.451±0.042	0.070±0.005	8.67
BiSeNet V1	0.217M	0.304±0.046	0.415±0.050	0.787±0.009	0.342±0.058	0.103±0.005	4.99

Table 6: Per-attack OOD IoU results across patch geometries, averaged over three random seeds

Model	Blob	Mask	Gls1	Gls2	Gls3	Tri.	Mean
U-Net	0.713	0.790	0.945	0.943	0.937	0.687	0.782
DB-LETNet V8	0.456	0.826	0.768	0.768	0.734	0.619	0.664
LETNet Baseline	0.545	0.676	0.674	0.681	0.634	0.634	0.629
ENet	0.338	0.712	0.649	0.653	0.589	0.540	0.554
MobileNetV3+DLV3	0.163	0.803	0.263	0.189	0.289	0.352	0.392
BiSeNet V1	0.271	0.264	0.259	0.297	0.238	0.417	0.304

All models degrade under distribution shift. U-Net remains best overall, while DB-LETNet V8 keeps the highest mean IoU among lightweight models at 0.664±0.043. The per-attack breakdown shows geometry-specific failures: DB-LETNet V8 performs best among lightweight models on facemask and glasses patches but loses accuracy on less regular blob patterns. Its OOD precision remains high at 0.857±0.001, while recall drops to 0.712±0.045. LETNet Baseline has slightly higher recall but lower precision, indicating a precision-recall trade-off rather than an absolute winner under shift.

4.5 Statistical robustness check

Table 7 reports paired seed-level IoU comparisons, where Δ represents DB-LETNet V8 minus the comparator for each of the three seed pairs (42, 123, and 456). DB-LETNet V8 improves over LETNet Baseline by 0.026 IoU in ID and 0.035 IoU in OOD, but both confidence intervals cross zero. Its gains over ENet, BiSeNet V1, and MobileNetV3+DLV3 are larger in ID, while only BiSeNet V1 and MobileNetV3+DLV3 remain significant in OOD. U-Net is significantly higher in both scenarios. The test therefore supports the intended claim of a favorable lightweight trade-off, not best overall accuracy.

4.6 Efficiency analysis

Figure 4 maps the accuracy-efficiency landscape using parameter count and mean IoU. The plotted values represent the mean IoU over three random seeds (42, 123, and 456), with error bars indicating the corresponding standard deviation. Separate Pareto frontiers

Table 7: Paired seed-level robustness analysis for DB-LETNet V8 against lightweight baselines using IoU

Scenario	Comparator	Δ IoU	95% CI	p
ID	LETNet Baseline	0.026	[-0.031, 0.083]	0.185
ID	ENet	0.097	[0.070, 0.123]	0.004
ID	BiSeNet V1	0.162	[0.159, 0.165]	< 0.001
ID	MobileNetV3+DLV3	0.285	[0.241, 0.329]	0.001
OOD	LETNet Baseline	0.035	[-0.142, 0.211]	0.488
OOD	ENet	0.109	[-0.061, 0.279]	0.110
OOD	BiSeNet V1	0.359	[0.326, 0.392]	< 0.001
OOD	MobileNetV3+DLV3	0.272	[0.192, 0.351]	0.005

are shown for ID and OOD performance. The inference time reported in Table 4 and Table 5 was measured on a desktop RTX 3060 GPU. It should therefore be interpreted as a controlled comparison within this experimental setup, not as a measurement of mobile, SoC, or microcontroller performance. In ID evaluation, DB-LETNet V8 occupies the leading lightweight position, reaching over 90% of U-Net accuracy at roughly 1% of its storage cost. In OOD evaluation, the frontier is less separated, but DB-LETNet V8 remains the highest-IoU lightweight point on average. The smaller OOD margin over LETNet Baseline indicates that specialization improves accuracy but does not remove distribution sensitivity.

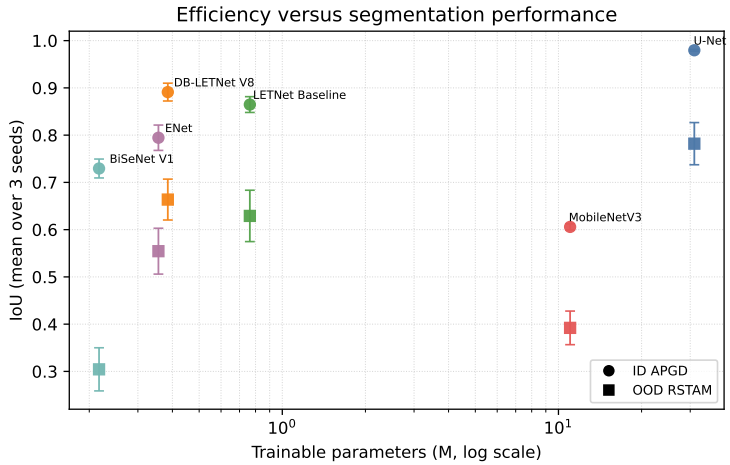

Figure 4: IoU–parameter trade-off across evaluated architectures, showing ID and OOD Pareto frontiers.

5 Discussion

Structural innovation outweighs parameter-level tuning. Moving from the tuned single-branch V6 to the dual-branch V7 gives the largest architectural gain, and branch ablation

confirms that gated fusion outperforms either branch alone. This suggests that separating semantic cues from boundary-sensitive cues [14] transfers well to adversarial patch segmentation and aligns with multi-path segmentation principles. In contrast, efficient blocks designed for classification do not necessarily transfer. MobileNetV3+DLV3 performs poorly despite 11M parameters, and Ghost modules reduce IoU sharply, likely because adversarial patch segmentation depends on localized texture and boundary evidence rather than redundant channel features [30], [39].

The OOD results reveal a specialization-generalization trade-off. DB-LETNet V8 keeps the best mean lightweight IoU and lower MAE, while LETNet Baseline preserves slightly higher recall. The blob case shows this limitation most clearly: DB-LETNet V8 drops to 0.456 IoU, below LETNet Baseline, suggesting that irregular blob contours weaken the boundary-branch advantage that is more effective on facemask and glasses patterns. This means DB-LETNet V8 is preferable when false positive masking must be limited, but simpler models may remain useful when recall is prioritized under uncertain attack geometry. U-Net remains the strongest option when compute is unconstrained. Four limitations constrain generalization. The dataset uses synthetic digital composites at 112×112 resolution. Training covers one PGD-based attack family against MobileFaceNet. Physical-world degradation is not evaluated. Inference time is measured on a desktop RTX 3060 GPU rather than actual edge hardware.

6 Conclusion

This study compared lightweight segmentation architectures for adversarial patch detection on face images through progressive exploration and controlled benchmarking. The results indicate that structural topology matters more than simple parameter reduction. Dual-branch semantic-edge processing with gated fusion provides the most important design gain. Mobile-efficient convolution blocks designed for classification do not transfer reliably to adversarial texture segmentation.

DB-LETNet V8 offers the most favorable mean accuracy-to-size trade-off among the evaluated lightweight models, but it should not be interpreted as the highest-accuracy detector because U-Net remains more robust when compute is unconstrained. The OOD results show that specialization improves lightweight performance but does not fully solve cross-attack generalization. Future work should cover more diverse attacks, printed patches, higher-resolution inputs, and real edge hardware.

Conflict of Interest

The authors declare no conflict of interest related to this work.

References

[1] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4690–4699, June 2019.

[2] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5265–5274, June 2018.

[3] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, vol. 429, pp. 215–244, 2021.

[4] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 1528–1540, 2016.

[5] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "A general framework for adversarial examples with objectives," *ACM Transactions on Privacy and Security*, vol. 22, no. 3, pp. 1–30, 2019.

[6] S. Komkov and A. Petiushko, "Advhat: Real-world adversarial attack on arcface face id system," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pp. 819–826, 2021.

[7] Z. Xiao, X. Gao, C. Fu, Y. Dong, W. Gao, X. Zhang, J. Zhu, and J. Zhou, "Improving transferability of adversarial patches on face recognition with generative models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11845–11854, 2021.

[8] X. Wei, Y. Guo, and J. Yu, "Adversarial sticker: A stealthy attack method in the physical world," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 2711–2725, 2023.

[9] A. Zolfi, S. Avidan, Y. Elovici, and A. Shabtai, "Adversarial mask: Real-world universal adversarial attack on face recognition models," in *Machine Learning and Knowledge Discovery in Databases*, pp. 304–320, 2023.

[10] X. Liu, F. Shen, J. Zhao, and C. Nie, "Eap: An effective black-box impersonation adversarial patch attack method on face recognition in the physical world," *Neurocomputing*, vol. 580, p. 127517, 2024.

[11] M. Wang, J. Zhou, T. Li, G. Meng, and K. Chen, "A survey on physical adversarial attacks against face recognition systems," *Neurocomputing*, vol. 669, p. 132485, 2026.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, 2015.

[13] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," 2016.

[14] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 334–349, 2018.

[15] G. Xu, J. Li, G. Gao, H. Lu, J. Yang, and D. Yue, "Lightweight real-time semantic segmentation network with efficient transformer and cnn," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 12, pp. 15897–15906, 2023.

[16] C. Xiang, A. N. Bhagoji, V. Sehwag, and P. Mittal, "Patchguard: A provably robust defense against adversarial patches via small receptive fields and masking," in *Proceedings of the USENIX Security Symposium*, 2021.

[17] C. Xiang, S. Mahloujifar, and P. Mittal, "Patchcleanser: Certifiably robust defense against adversarial patches for any image classifier," in *Proceedings of the USENIX Security Symposium*, 2022.

[18] B. Liang, H. Li, M. Su, X. Li, W. Shi, and X. Wang, "Detecting adversarial image examples in deep neural networks with adaptive noise reduction," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 1, pp. 72–85, 2021.

[19] A. Agarwal, R. Singh, M. Vatsa, and N. Ratha, "Image transformation-based defense against adversarial perturbation on deep learning models," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2106–2121, 2021.

[20] M. Naseer, S. Khan, and F. Porikli, "Local gradients smoothing: Defense against localized adversarial attacks," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1300–1307, 2019.

[21] J. Liu, A. Levine, C.-Y. Lau, R. Chellappa, and S. Feizi, "Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14953–14962, 2022.

[22] B. Tarchoun, A. B. Khalifa, M. A. Mahjoub, N. Abu-Ghazaleh, and I. Alouani, "Jedi: Entropy-based localization and removal of adversarial patches," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4087–4095, 2023.

[23] C. Kang, Y. Dong, Z. Wang, S. Ruan, Y. Chen, H. Su, and X. Wei, "Diffender: Diffusion-based adversarial defense against patch attacks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 130–147, 2024.

[24] L. Jing, R. Wang, W. Ren, X. Dong, and C. Zou, "Pad: Patch-agnostic defense against adversarial patch attacks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24472–24481, 2024.

[25] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3992–4003, 2023.

[26] X. Liu, F. Shen, J. Zhao, and C. Nie, "Radap: A robust and adaptive defense against diverse adversarial patches on face recognition," *Pattern Recognition*, vol. 157, p. 110915, 2025.

[27] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation," *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3051–3068, 2021.

[28] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314–1324, 2019.

[29] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 801–818, 2018.

[30] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghostnet: More features from cheap operations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1580–1589, 2020.

[31] S. Chen, Y. Liu, X. Gao, and Z. Han, "Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices," in *Chinese Conference on Biometric Recognition (CCBR)*, pp. 428–438, 2018.

[32] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proceedings of the International Conference on 3D Vision (3DV)*, pp. 565–571, 2016.

[33] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017.

[34] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," in *International Conference on Learning Representations (ICLR)*, 2018.

[35] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations (ICLR)*, 2018.

[36] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 2206–2216, 2020.

[37] X. Liu, F. Shen, J. Zhao, and C. Nie, "Rstam: An effective black-box impersonation attack on face recognition using a mobile and compact printer," 2022.

[38] I. C. Duta, L. Liu, F. Zhu, and L. Shao, "Improved residual networks for image and video recognition," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pp. 9415–9422, 2021.

[39] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pp. 3–11, 2018.