Real-Time Topology-Aware Branch Segmentation and Grasp Localization for UAV Interaction

Tong Wang , Zhengran Zhou , Suzuki Satoshi *

Posted Date: 25 May 2026

doi: 10.20944/preprints202605.1572.v1

Keywords: UAV; real-time segmentation; topological perception; branch grasping

 doi:10.20944/preprints202605.1572.v1

Article

Real-Time Topology-Aware Branch Segmentation and Grasp Localization for UAV Interaction

Tong Wang, Zhengran Zhou and Suzuki Satoshi *

Graduate School of Engineering, Chiba University, 1-33 Yayoi-cho, Inage-ku, Chiba 263-8522, Japan
* Correspondence: suzuki-s@chiba-u.jp

Abstract

Reliable localization of interaction-feasible branch regions is a critical prerequisite for autonomous UAV(Unmanned Aerial Vehicle) interaction in natural environments. However, natural tree branches exhibit slender geometries, irregular topologies, frequent occlusions, and unstable bifurcation structures, thus rendering it difficult to extract physically reliable grasp candidates under the limited computational resources of onboard platforms. Hence, we propose a real-time branch-grasp localization framework that integrates semantic perception with topology-aware geometric reasoning. For perception, we introduce a strip-swift pyramid pooling module to enhance the elongated structure representation through progressive pooling and strip-based directional context aggregation. To further improve the deployment efficiency and boundary quality, a reparameterized golden cudgel block and a boundary-optimization module are incorporated into the lightweight segmentation architecture. Based on predicted masks, we develop a topology-guided grasp-localization pipeline. Skeleton-based structural analysis is first performed to remove unstable regions such as branch junctions and overlapping structures. Subsequently, directional kernels are applied to extract geometrically consistent branch segments aligned with feasible interaction orientations. Finally, temporal stabilization and geometric constraints are introduced to suppress localization jitter caused by UAV motion. Experimental results show that the proposed method achieves a mean intersection over union of 89.96% on the Drone-Branch dataset. When deployed on the NVIDIA Jetson Orin Nano with TensorRT acceleration, the system achieves a stable latency of 13.9 ms, thus demonstrating its effectiveness and real-time suitability for onboard UAV branch interaction and grasp-candidate localization.

Keywords: UAV; real-time segmentation; topological perception ; branch grasping

1. Introduction

Autonomous UAV interactions with the natural environment have been widely investigated for applications such as environmental inspection, long-duration monitoring, aerial manipulation, and contact-based environmental interactions [1–3]. Unlike conventional aerial perception tasks, which primarily require scene understanding, physical interaction demands the localization of geometrically feasible regions that can support stable contact between a UAV and the environment.

In these scenarios, tree branches provide naturally distributed support structures in outdoor environments and have recently been investigated as potential interaction targets for UAV perching and grasp-assisted stabilization [4–6]. However, reliable branch interactions remain highly challenging because of the slender geometry, irregular topology, frequent occlusions, and highly anisotropic appearance of natural branches. More importantly, not all visible branch regions are physically suitable for interaction. Unstable structures, such as bifurcations, overlapping branches, and highly curved segments, may result in unreliable grasping or unstable physical contact.

Consequently, the problem extends beyond generic branch detection to the topology-aware localization of interaction-feasible branch regions. Such localization requires fine-grained structural understanding, including local continuity, orientation consistency, and branch-thickness estimation.

 doi:10.20944/preprints202605.1572.v1

Conventional bounding-box-based object detectors are unsuitable for this task because they cannot adequately represent the geometric and topological properties required for physically feasible interaction analysis. By contrast, pixel-level semantic segmentation enables a detailed structural representation of elongated targets and has demonstrated strong potential in thin-structure perception tasks, such as road extraction, vessel segmentation, and branch analysis [7–11]. Therefore, semantic segmentation is a critical prerequisite for extracting reliable grasp candidates from natural branch structures.

Despite recent progress in real-time semantic segmentation, existing lightweight architectures fail to preserve the structural continuity of slender branch regions under embedded deployment constraints. In particular, conventional square receptive fields are not suitable for modeling the directional continuity and anisotropic geometry of branches, thus resulting in fragmented predictions and unstable topological representations. Furthermore, most existing branch-perception methods focus primarily on visual recognition while disregarding the structural constraints required for physically feasible UAV interactions.

Hence, this study proposes a real-time topology-aware branch-grasp localization framework for onboard UAV systems. The main contributions of this study are as follows:

1) **Strip-Swift Pyramid Pooling Module (SSPPM):** We propose an SSPPM for the effective multiscale representation of elongated branch structures. It replaces conventional parallel pyramid designs with a progressive pooling strategy while introducing a dedicated strip pooling mechanism for anisotropic context modeling. This design captures long-range directional continuity and improves computational efficiency by reducing GPU load imbalance, thus yielding consistent accuracy improvements across datasets.
2) **Deployment-Friendly Backbone with Edge Refinement:** We develop a high-efficiency backbone that integrates reparameterized golden cudgel blocks (GCBlocks) with a boundary-optimization module (BOM). The architecture leverages multibranch training for enhanced representation while collapsing into a streamlined single-path structure during inference, thus achieving improved boundary precision for slender targets without additional deployment overhead.
3) **Topology-Guided Grasp Localization Framework:** We propose a topology-guided grasp localization pipeline that bridges semantic perception and interaction-oriented geometric reasoning. By performing topology-aware skeleton pruning to remove structurally unstable regions and applying directional kernel-based extraction to identify geometrically consistent branch segments, the framework generates feasible grasp candidates aligned with stable interaction orientations. This design improves the structural reliability and temporal stability of grasp localization for onboard UAV branch interaction.

The remainder of this paper is organized as follows: Section 2 reviews the existing literature on nonground perching mechanisms for UAVs and discusses contemporary semantic segmentation methods applied to branch recognition. Section 3 describes the proposed grasp execution pipeline, where the workflow from mask generation via the segmentation model to the integration of depth information for physical feasibility filtering s detailed. Section 4 focuses on the enhancements to the real-time semantic segmentation framework. Specifically, the reparameterized GCBlock and BOM for improved perception efficiency are introduced, in addition to the novel SSPPM designed for elongated branch structures. Section 5 provides a detailed explanation of the post-processing steps for branch masks. This includes a strategy for filtering hazardous multibranch regions and bifurcation nodes, as well as a methodology for identifying optimal horizontal branches with suitable inclination angles. Furthermore, it describes a grasp-box locking mechanism designed to mitigate jitter interference caused by UAV flight dynamics. Section 6 presents the experimental results, including comprehensive ablation studies of model improvements and field flight tests that validate the robustness of the grasp-point-selection algorithm. Finally, Section 7 concludes the paper and discusses potential directions for future investigations.

 doi:10.20944/preprints202605.1572.v1

2. Existing Studies

UAV Physical Interaction with Natural Environments

Recent advances in aerial robotics have enabled UAVs to physically interact with their surrounding environment for applications such as inspection, aerial manipulation, infrastructure maintenance, and energy-efficient stabilization [12–14]. Existing studies have demonstrated UAV attachment and perching on various artificial structures, including walls, ceilings, poles, and power lines [15–17]. These studies validate the feasibility of aerial interactions and highlight the importance of reliable target localization prior to physical contact.

Bio-inspired robotics has promoted the development of aerial grasping and perching systems. Avian-inspired grasping mechanisms, passive tendon-driven structures, and adaptive compliant claws have been proposed to improve grasp robustness under uncertain contact conditions [18–20]. Additionally, high-speed grasping strategies for micro-UAVs and specialized perching systems for cables or timber structures have demonstrated the potential for aerial interaction in constrained environments [21,22].

However, most existing systems assume predefined or geometrically simplified targets. Reliable interactions with natural tree branches remain considerably challenging because of their irregular topologies, varying diameters, occlusions, and complex bifurcation structures. These challenges have shifted the focus from generic target detection to interaction-oriented structural perception, which enables the identification of geometrically feasible and structurally stable branch regions.

Interaction-Oriented Branch Perception

Existing studies on tree-branch perception are primarily driven by agricultural and forestry applications, such as pruning, yield estimation, and structural inspection [23,24]. Early approaches primarily relied on handcrafted feature extraction and object-detection techniques, whereas more recent methods have increasingly employed semantic or instance segmentation to obtain fine-grained branch masks [25,26]. For example, RGB-D perception and instance segmentation have been investigated for pruning-point localization and branch-structure analysis in cluttered environments [27].

Unlike generic object recognition, branch understanding is classified under topology-sensitive thin-structure perception problems characterized by elongated geometries, sparse spatial distributions, and strong structural continuity dependence. Similar topology-preserving perception challenges have been investigated in related domains such as thin-structure object detection, thin-obstacle segmentation, and retinal vessel segmentation, where maintaining structural continuity is essential for downstream analysis [28–30].

For UAV branch interactions, the objective extends beyond identifying visible branch regions to localizing interaction-feasible branch segments. Such localization requires fine-grained structural understanding, including local continuity, orientation consistency, branch-thickness estimation, and the avoidance of unstable structures such as bifurcations and overlapping branches. Consequently, conventional object detectors based on bounding-box representations are insufficient for interaction-oriented branch analysis.

Although semantic segmentation provides richer structural representations, most existing methods focus primarily on visual-recognition accuracy while disregarding the geometric and topological constraints required for physically reliable UAV interactions.

Real-Time Thin-Structure Segmentation

Reliable branch interactions require real-time semantic segmentation that can operate on resource-constrained embedded UAV platforms. Existing real-time segmentation architectures generally reflect two major paradigms.

1) **Single-branch Architectures:** Models such as STDC [31] and Fast-SCNN [32] prioritize inference efficiency through lightweight backbones and aggressive downsampling strategies. Although these architectures achieve high frame rates, they typically sacrifice spatial continuity in deeper layers, thus resulting in fragmented predictions for slender branch structures.

 doi:10.20944/preprints202605.1572.v1

2) **Multi-branch Architectures:** Methods such as BiSeNet [33] and DDRNet [34] employ bilateral pathways to simultaneously preserve spatial details and semantic context. While these architectures improve boundary quality, their frequent feature-fusion operations and complex memory-access patterns may introduce additional computational overhead during deployment.

Despite their architectural differences, most existing segmentation frameworks rely predominantly on isotropic square receptive fields (e.g., 3×3 and 7×7 kernels). However, such receptive fields are not suitable for the highly directional and anisotropic geometry of natural branches [35]. Consequently, existing lightweight models fail to preserve the topology continuity and directional consistency of elongated branch structures under real-time deployment constraints.

Furthermore, most current branch-perception pipelines focus primarily on semantic recognition while disregarding the structural stability and temporal consistency required for interaction-oriented grasp localization. In dynamic UAV scenarios, unstable topological representations and fragmented masks can significantly degrade the reliability of grasp-candidate extraction.

Research Gap

In summary, although significant progress has been achieved in UAV interactions, branch perception, and real-time semantic segmentation, several critical limitations remain.

1) Inadequate multiscale representation for anisotropic branch structures – Existing lightweight segmentation models rely on isotropic square receptive fields, thus resulting in fragmented predictions for slender branches.
2) Boundary precision vs. deployment efficiency in dual-path architectures – Current dual-path models improve context but necessitate frequent fusion operations and incur memory overheads, thus rendering them inefficient in terms of boundary refinement for slender targets.
3) Inadequate topology-guided reasoning for grasp localization under dynamic flight – Most pipelines end at semantic recognition, which implies structural stability (e.g., bifurcations, curved segments) and temporal consistency are disregarded, thereby resulting in jittery and infeasible grasp candidates during UAV operations.

These limitations warrant a unified framework that jointly enables anisotropic multiscale perception, efficient boundary-aware segmentation, and topology-constrained grasp localization for onboard UAV branch interactions.

3. System Architecture and Workflow

The proposed system performs real-time branch perception and grasp decision-making for autonomous UAV perching. As illustrated in Figure 1, the framework integrates deep-learning-based perception with topology-aware geometric reasoning to identify safe grasping locations in complex natural environments. The system establishes a closed-loop pipeline from RGB-D sensing to final UAV grasp planning, which comprises four main stages: (1) perception and mask preprocessing, (2) topology-aware structural extraction, (3) physically consistent grasp segment detection, and (4) spatiotemporal decision optimization.

Figure 1 illustrates the workflow of the proposed framework. The perception module first extracts branch regions from RGB-D images and then performs mask refinement. Subsequently, the refined mask is skeletonized for topology analysis to remove structurally unsafe regions such as branch intersections and occlusions. Based on the remaining safe skeletons, candidate grasp segments are detected and verified based on geometric and physical constraints. Finally, a spatiotemporal decision module stabilizes the grasp target across frames to ensure reliable grasp point selection during UAV flights. The detailed design of the segmentation network and the grasp-candidate generation strategy will be introduced in the following sections.

 doi:10.20944/preprints202605.1572.v1

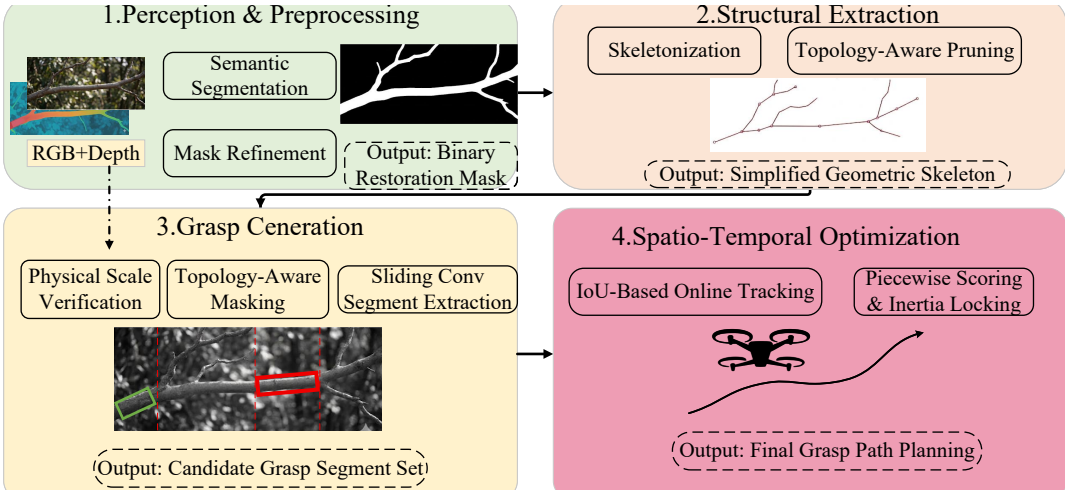

Figure 1. Structure of proposed system.

1) **Perception and Feature Refinement**: To accommodate the slender and multiscale nature of tree branches, the system utilizes a lightweight backbone enhanced by the SSPPM. This stage yields a high-fidelity semantic mask that captures long-range directional context, while a BOM ensures sharp edge definition, thus providing a clean input for subsequent structural analysis.
2) **Topology-Aware Structural Analysis**: Because branch junctions and overlapping segments result in unstable perching, this module evaluates the skeletonized mask to identify potential collision or slip risks. By analyzing the local connectivity and transition density, the system removes complex topological singularities and forms "safety buffers," thereby retaining only skeleton segments that offer a stable contact geometry.
3) **Physical Scale-Based Candidate Extraction**: This stage bridges topological features with executable grasping. A set of directional kernels designed based on the gripper's physical dimensions is applied to the safe skeleton to extract continuous branch segments. By projecting pixel-level candidates into three-dimensional space using depth information, the system filters segments based on real-world metric constraints (e.g., length and width), thus yielding a set of physically feasible grasp boxes.
4) **Spatiotemporal Decision Optimization**: To mitigate perception fluctuations caused by UAV egomotion and environmental noise, we introduce a temporal stabilization mechanism. It employs a SORT-inspired tracking module with a hybrid IoU–Euclidean metric to maintain consistent identity for grasp candidates. A distance-dependent scoring strategy is further implemented to adaptively transition between long-term stability and precise spatial alignment as the UAV approaches the target. Finally, an inertia-based switching constraint is applied to prevent jitter, thus ensuring a smooth and reliable grasp selection during dynamic flight.

4. Improved DDRNet

As illustrated in Figure 2, the proposed model retains the core dual-path skeleton and bilateral fusion mechanism of DDRNet. To satisfy the stringent real-time requirements of drone-borne platforms, we introduce three strategic enhancements:

(1) GCBlock for efficient feature extraction,
(2) SSPPM for balanced multi-scale context aggregation, and
(3) BOM with focal loss to refine the structural contours.

 doi:10.20944/preprints202605.1572.v1

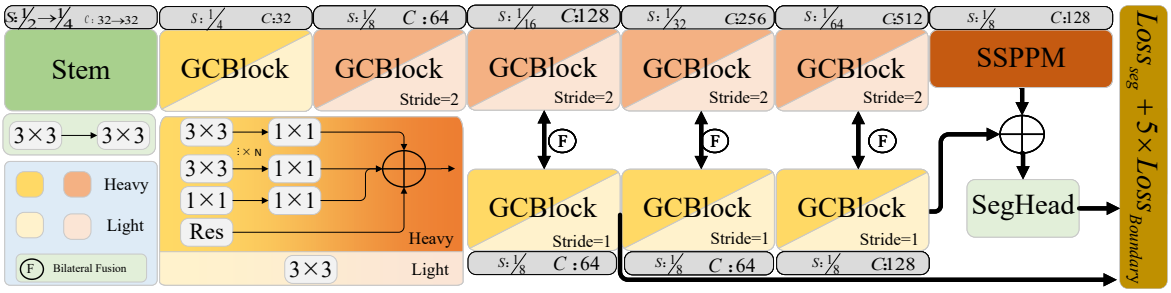

Figure 2. The structure of the improved DDRNetBrnach.

4.1. GCBlock

To enhance feature-extraction capabilities while maintaining real-time inference speed, the GCBlock [36] is integrated into the DDRNetBranch. The core philosophy of GCBlock is structural reparameterization, which is characterized by the principle of "self-enlargement during training and self-contraction during inference."

As illustrated in Figure 2, the GCBlock adopts a complex multiconvolution and multipath architecture during the training phase. By employing vertical convolution stacking (e.g., 3×3 convolution followed by 1×1 convolution) and horizontal multibranch parallelism, the block significantly strengthens the network's representational ability. This design enables the model to function as an "internal teacher," thereby effectively capturing deep semantic features and precise boundary information without requiring external pretrained models or complex distillation schemes.

Despite its complex training structure, the GCBlock is simplified through mathematically equivalent transformations prior to inference. First, the convolution layers are fused with their subsequent BatchNorm layers. Subsequently, through vertical fusion (combining sequential 3×3 and 1×1 kernels) and horizontal summation (adding weights from all parallel paths), the entire module collapses into a single, standard 3×3 convolutional layer. This process eliminates the memory-access cost typically associated with multipath structures, thereby ensuring superior inference efficiency.

Within the DDRNetBranch architecture, the GCBlock is deployed with different strides to satisfy specific branch requirements: Semantic Branch: The GCBlock operates with a stride of 2. In this configuration, it performs efficient downsampling while expanding the receptive field to capture the global semantic context. Detailed Branch: The GCBlock operates with a stride of 1. This branch maintains high-resolution feature maps, with emphasis on the extraction of fine-grained boundaries and geometric details through the diverse paths of the GCBlock.

4.2. SSPPM

The original deep aggregation pyramid pooling module (DAPPM) captures multiscale contextual information by combining multiple parallel branches with heterogeneous kernel sizes, strides, and upsampling operations. Each branch extracts features at a specific receptive field, and the resulting outputs are upsampled and added element-wise for fusion. Although effective in enriching representation, this fragmented multibranch design presents several disadvantages. Specifically, the heterogeneous computational demand across branches result in GPU load imbalance; some computing units are underutilized, whereas others are overloaded. Moreover, the repeated use of large-kernel convolutions and upsampling introduces redundant operations and a significant memory-bandwidth overhead, which consequently degrades the inference efficiency.

To overcome these limitations and further enhance the capture of long-range dependencies, we propose a SSPPM, as shown in Figure 3. The SSPPM reformulates the DAPPM by replacing the fragmented parallel branches with a streamlined serial pooling–convolution pipeline while simultaneously introducing a dedicated strip-pooling branch to manage complex, anisotropic spatial contexts.

 doi:10.20944/preprints202605.1572.v1

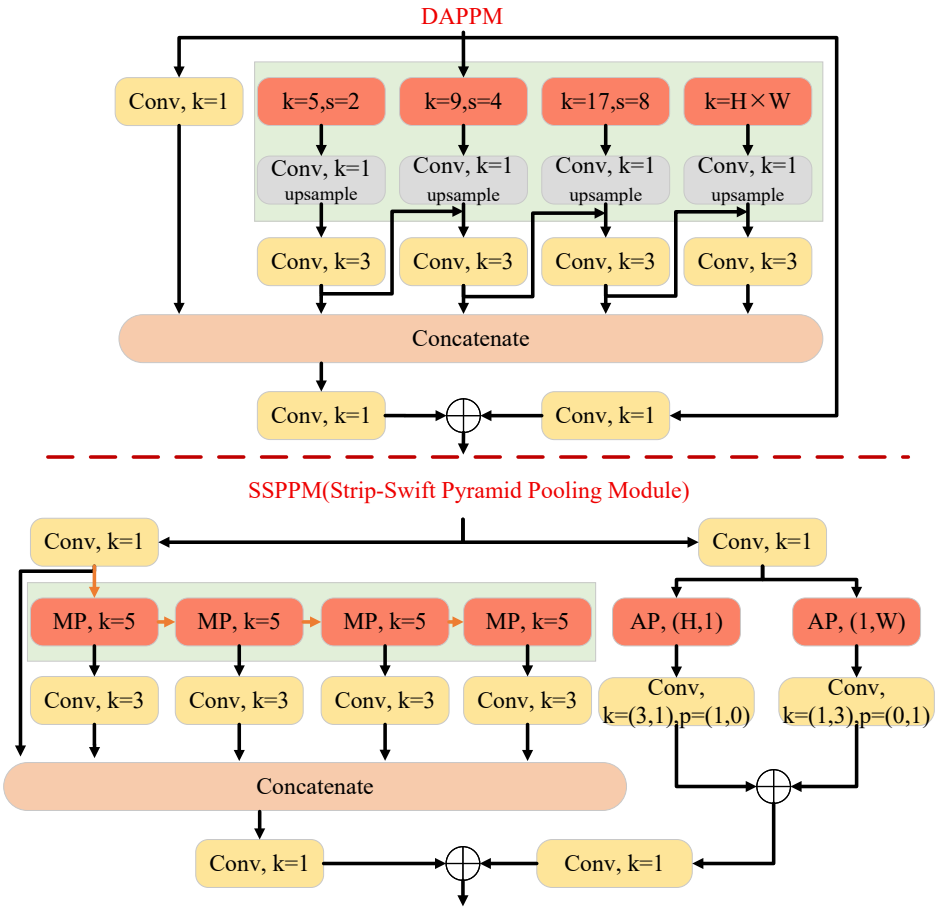

Figure 3. Strip-Swift Pyramid Pooling Module(SSPPM).

To address the GPU load imbalance, the SSPPM first utilizes a "swift" serial pipeline. It employs four successive 5×5 max-pooling layers (stride=1, padding=2) to ensure that spatial resolution is preserved while the receptive fields are gradually expanded. The equivalent receptive field of the k-th layer can be computed using the following recursive formulation:

$$l_k = l_{k-1} + \left((f_k - 1) * \prod_{i=1}^{k-1} s_i \right) \tag{1}$$

where l_k denotes the receptive field at the k-th layer, l_{k-1} the receptive field of the previous layer, f_k the kernel size of the k-th operation, and s_i the stride of layer i. Because pooling was is conducted with a stride $s_i = 1$, the effective receptive field expands linearly. For example, stacking n layers of 5×5 pooling yields 5, 9, 13, and 17 receptive fields for $n = 1, 2, 3, 4$, respectively. Following each pooling operation, a 3×3 convolution is applied to densify feature extraction. This avoids the sparse computation characteristics of large-kernel convolutions and strided operations in the original DAPPM, thus achieving equivalent multiscale coverage at a substantially lower cost.

Although the serial pooling pipeline efficiently captures local and regional contexts, it relies on square windows. Using large, square pooling windows cannot accommodate objects that have long-range banded or discretely distributed structures because they inevitably incorporate contaminated information from irrelevant regions. Hence, a strip pooling branch is integrated into the SSPPM. Instead of a square kernel, strip pooling requires the spatial pooling extent of $(H, 1)$ or $(1, W)$. By averaging all feature values in a row or column, it deploys a long kernel shape along one spatial dimension to capture long-range relations while maintaining a narrow shape along the other to prevent irrelevant regions from interfering. The input undergoes average pooling along both horizontal and

 doi:10.20944/preprints202605.1572.v1

vertical dimensions, and the results are input to one-dimensional convolutional layers with a kernel size of 3 to modulate the current location and its neighboring features.

Prior to processing, a 1×1 convolution reduces the channel dimensionality, thereby alleviating the computational overhead. Finally, the outputs from the initial 1×1 branch, the multiscale serial cascade, and the horizontal/vertical strip pooling layers are concatenated together. The final 1×1 convolution combines these diverse spatial and semantic embeddings into a compact representation.

In summary, the SSPPM maintains its ability to capture multiscale contextual cues while significantly improving efficiency and representational ability. By combining a serial pooling–convolution pipeline with anisotropic strip pooling, it achieves balanced GPU utilization; avoids the noise of large, square pooling windows; and offers faster, more accurate inferences.

4.3. BOM

Accurately defining the boundaries between slender and irregular branches for branch removal or manipulation is a significant challenge in autonomous drone operations. Owing to the complex backgrounds and inherent thinness of branches, standard segmentation losses typically result in blurred contours and thus imprecise grasping box localization. Hence, we propose a BOM. This design is inspired by two key studies: the joint segmentation and boundary-detection framework proposed in [37] and the focal-loss formulation introduced in [38].

The first study [37] presents a dual-branch network with a shared backbone that enables semantic segmentation and boundary detection to interact across multiple scales. This joint learning paradigm allows boundary information to substantially enhance the structural accuracy and contour sharpness of segmentation results, while semantic cues from segmentation simultaneously improve the accuracy of boundary detection. The second study [38] reformulates the standard cross-entropy loss by introducing a modulating factor that dynamically downweights easy-to-classify samples (e.g., abundant background pixels), thus causing the training to prioritize difficult or rare samples. This approach effectively addresses the class-imbalance problem and improves the model's ability to capture rare and informative features.

Based on these insights, we integrate focal loss into the boundary-detection branch of our framework. In branch-grasping tasks, boundary pixels are much sparser than nonboundary pixels, and conventional cross-entropy loss tends to bias the model toward background regions, thus resulting in blurred or inaccurate contours for slender branches. By contrast, focal loss directly mitigates this imbalance by prioritizing harder, less-represented boundary samples. The original focal loss is adapted to the boundary-detection task, thereby yielding a boundary focal loss L_{boundary} of the form

$$L_{\text{boundary}} = -\frac{1}{N}\sum_{i}\alpha(1-p_i)^{\gamma}\log(p_i) \tag{2}$$

where p_i denotes the predicted probability of pixel i as a boundary, N is the total number of pixels, $\alpha = 0.25$ balances the positive and negative boundary pixels, and $\gamma = 2$ controls the degree of focus on hard boundary samples. The final objective function is expressed as

$$L_{\text{total}} = L_{\text{seg}} + 5 \cdot L_{\text{boundary}} \tag{3}$$

where L_{seg} is the segmentation loss, and the boundary loss is assigned a larger weight to ensure adequate supervision for edge refinement. Boundary annotations are derived from the segmentation masks by applying edge-detection operators.

To illustrate the role of BOM more effectively, Figure 4 presents the intermediate edge-detection results derived from the context branch. This branch leverages deep semantic features to achieve a robust category representation. Subfigure (a) presents the final segmentation output, where the contours of slender branches are expected to be sharp and continuous. Subfigure (b) shows the ground-truth edge map (GT_{edge}) extracted from the labels, which serves as the target for the proposed boundary-aware supervision. The core effectiveness of the module is captured in subfigure (c), which

 doi:10.20944/preprints202605.1572.v1

represents the predicted edge intensity (entropy map). In our framework, focal loss is applied to align (c) with (b). Because the edge pixels in (b) are extremely sparse, focal loss ensures that the model is not biased toward overwhelming the background pixels. As shown in (c), the high-entropy regions are precisely localized along the branch boundaries with minimal noise. This high-fidelity edge prediction effectively "guides" the segmentation process in (a), thus preventing the common issue of blurred or fragmented contours in slender structures and ensuring a robust geometric basis for subsequent grasping tasks.

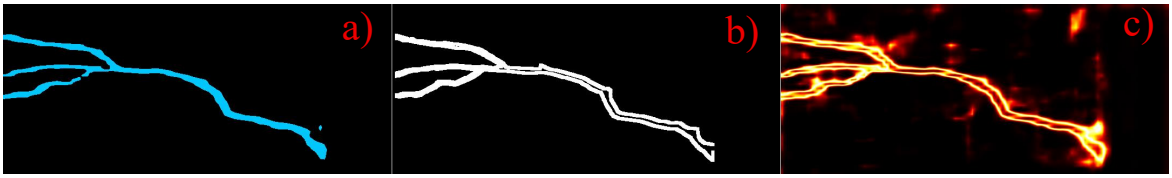

Figure 4. Visualization of boundary optimization module (BOM). (a) Final predicted segmentation result; (b) ground-truth edge map (GT_{edge}) used for supervision; (c) predicted edge confidence map (entropy).

5. Topology-Guided Grasp Detection and Stabilization

5.1. Topology-Aware Structural Analysis

To guarantee safe UAV landing, the system performs a topology analysis on the skeletonized branch mask to detect structurally complex regions.

Let the binary segmentation mask be denoted as $M(x, y) \in \{0, 1\}$, where $M(x, y) = 1$ indicates the branch pixels. A morphological thinning operation is applied to obtain a one-pixel-wide skeleton representation $S = \text{Skeleton}(M)$, which preserves the topological structure of the branch network, as shown in Figure 5.

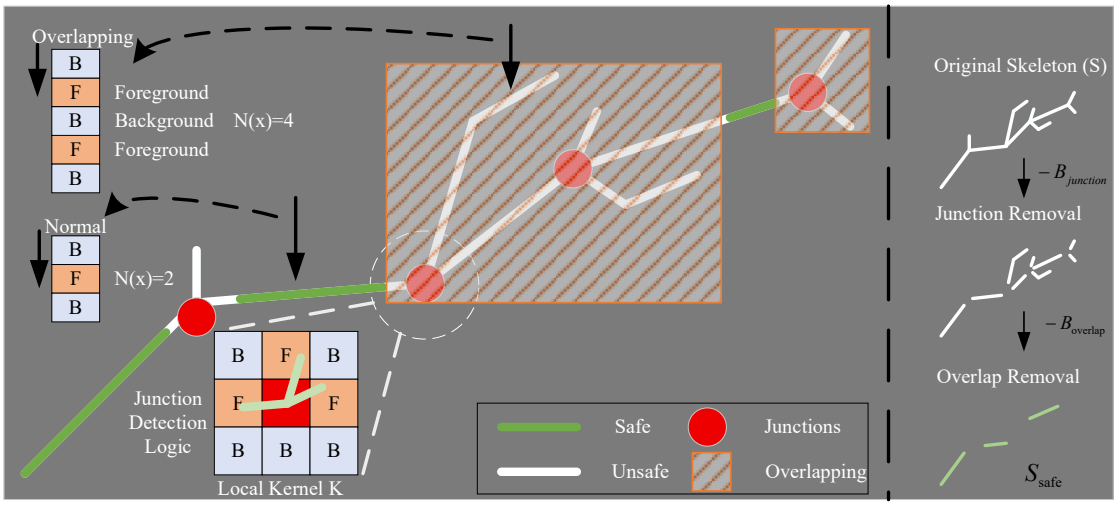

Figure 5. Topology-Aware Structural Analysis.

Branch junctions are detected by applying a local convolution operator to the skeleton map as follows:

$$R(x, y) = (S * K)(x, y) \tag{4}$$

where K is a 3×3 neighborhood kernel that counts the number of connected skeleton pixels. Pixels with high response values $R(x, y)$ correspond to locations where multiple branches intersect, which are considered geometrically unstable for UAV landing.

In addition to local junctions, overlapping branches along the viewing direction are detected through column-wise structural analysis. For each image column x, the number of foreground

 doi:10.20944/preprints202605.1572.v1

transitions is computed as $N(x) = \sum_y |M(x, y+1) - M(x, y)|$. Physically, a high $N(x)$ value indicates the presence of "multilayer" structures or clutter within a single viewing ray, which poses a high risk of entanglement for UAV propellers. Such regions are classified as multibranch clutter and are subsequently pruned. Both the junction areas and multibranch regions are expanded using a dilation operation to form safety buffers B. The skeleton segments located inside these buffers are removed, thus resulting in a simplified safe skeleton representation $S_{\text{safe}} = S \setminus B$.

5.2. Physical Scale-Based Candidate Extraction

After topology-aware pruning, graspable branch segments are extracted from S_{safe}, as illustrated in Figure 6. This module establishes a hierarchical transformation from a discrete topological representation to physically feasible grasp candidates.

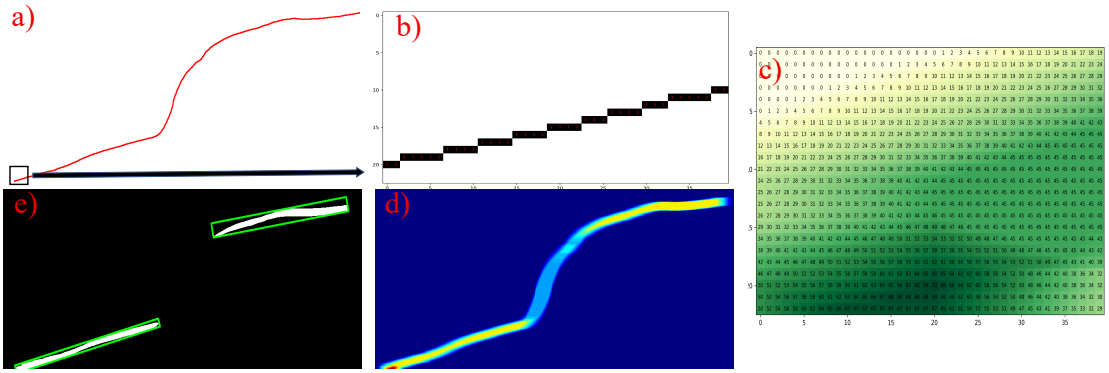

Figure 6. Hierarchical workflow of candidate grasp extraction process: (a) Refined skeleton matrix used as primary input; (b) magnified view of local sparse matrix (0/1 values) before convolution; (c) numerical output matrix showing accumulated scores after adaptive convolution; (d) global energy heatmap (convolutional response field) highlighting candidate zones; (e) final oriented grasp boxes filtered by physical scale and geometric constraints.

To detect straight and continuous branch segments, we employ a set of n elongated convolution kernels $K_i \in \mathbb{R}^{h_i \times w_i}$ that explicitly encode the directional continuity, where $h_i \gg w_i$ (or vice versa) enforced anisotropic sensitivity. These kernels are constructed along multiple orientations θ_i to include potential branch directions. The convolution response is computed as follows:

$$F_i(x, y) = (S_{\text{safe}} * K_i)(x, y), \tag{5}$$

where α_i is the weighting coefficient (in practice, uniform weights are adopted). The resulting map F forms a global energy field in which the high-response regions correspond to continuous, well-aligned branch segments. We extract the candidate regions Ω by applying a threshold $\tau = 35$, which represents the minimum structural support required for a stable grasp based on the gripper's contact pad length.

From the extracted regions Ω, the connected components are identified and converted into oriented grasp boxes $g = (x_c, y_c, l, w, \theta)$, where (x_c, y_c) is the center; l and w are the length and width, respectively; and θ is the principal orientation estimated via PCA. To ensure physical feasibility, each candidate is validated using depth information. For depth Z and focal length f_x, the physical length is calculated as $L_{\text{phys}} = (Z/f_x)L_{\text{pixel}}$. Only candidates satisfying the hardware constraints of the UAV gripper (e.g., the metric length, width, and orientation compatibility) are retained in the final candidate set $G = \{g_1, g_2, \ldots, g_k\}$.

5.3. Spatio-Temporal Decision Optimization

Owing to UAV motion and environmental disturbances, the perceived grasp candidates may fluctuate across frames. To ensure stable decision-making, we propose a temporal optimization framework comprising candidate monitoring and a distance-aware inertial locking mechanism.

 doi:10.20944/preprints202605.1572.v1

Inspired by the SORT framework, this system associates candidate boxes across frames to maintain temporal continuity. For candidate g_i^t in the current frame and g_j^{t-1} in the previous frame, we define a hybrid similarity metric S_{match} as follows:

$$S_{match} = \lambda \cdot IoU(g_i^t, g_j^{t-1}) + (1 - \lambda) \exp(-\|c_i - c_j\|), \tag{6}$$

where c_i and c_j denote the normalized box centers. This joint metric enhances monitoring robustness against rapid UAV maneuvers and partial occlusions.

To manage the transition from a long-range approach to short-range perching, each monitored candidate is evaluated using the following dynamic scoring function:

$$\text{Score } = w_A(d) \cdot A + w_H(d) \cdot H + w_D(d) \cdot D, \tag{7}$$

where A is the box area, H the observation count (monitoring persistence), and D the spatial alignment with the UAV's optical center. The weights are adaptively adjusted based on the UAV-to-branch distance d.

1) **Stable mode (large d):** The system prioritizes w_H and w_A to lock onto large, consistently observed branches.
2) **Centering mode (small d):** The weight w_D is increased to emphasize spatial centering, thus minimizing the landing offset for a precise grasp.

Finally, to prevent "target jitter" from perception noise, an inertia constraint is applied: the system maintains the current target unless a new candidate satisfies $\text{Score}_{\text{new}} > \gamma \cdot \text{Score}_{\text{current}}$. This hysteresis-like behavior ensures that the UAV remains locked onto a stable branch segment during the critical perching phase.

6. Experiment

This section describes the comprehensive experimental framework designed to validate the proposed methodology. It encompasses dataset preparation, evaluation protocols, and quantitative analysis of the results. The experimental campaign is structured into two distinct phases: (1) offline evaluation of segmentation performance and (2) edge deployment validation with flight-based visual verification. The overarching objective is to substantiate the segmentation accuracy and cross-dataset generalization capability of the proposed DDRNet-Branch architecture, specifically in the context of UAV-based perception for autonomous branch grasping.

6.1. Experimental Setup and Evaluation Metrics

6.1.1. Dataset Description

The experimental evaluation in this study uses two distinct data sources to ensure a comprehensive assessment of the proposed method for autonomous branch grasping. First, a proprietary, task-specific dataset is curated to reflect the operational conditions of a UAV performing close-range branch grasping. This private collection comprises 757 images sourced from both direct UAV-mounted camera captures and supplementary online images. The data are predominantly characterized by local branch structures situated within 0 to 2 m from the sensor, which aligns with the intended grasping range of the aerial platform.

Second, to benchmark the performance against established standards and enhance the generalizability of the findings, a public dataset was employed. Specifically, the branch component of the Tree dataset of Urban Street was utilized. This publicly available resource contains 1,485 high-resolution urban scene images meticulously annotated for instance-level segmentation. As detailed in the original dataset released by Zhejiang Agriculture and Forestry University (2022), the images span 13 distinct tree species—including branches of albizia julibrissin, flowering cherry, and ginkgo biloba—captured across diverse seasonal and climatic conditions in Chinese urban environments. The dataset provides

 doi:10.20944/preprints202605.1572.v1

1,193 training, 149 validation, and 143 testing images, thus offering a rigorous benchmark for the pixel-level recognition of arboreal structures amidst complex street-level backgrounds.

6.1.2. Experimental Setup and Implementation Details

The proposed framework was developed and evaluated on two distinct computing platforms to simulate the transition from offline training to real-time aerial deployment, as summarized in Table 1. Model training was conducted on a high-performance workstation equipped with an Intel Core i9-14900K CPU and an NVIDIA GeForce RTX 4080 GPU, thus providing the necessary throughput for large-scale architectural optimization. For onboard validation, the model was deployed on an NVIDIA Jetson Orin Nano Edge module. This platform, which features 1,024 CUDA cores and 8 GB of memory, represents a typical resource-constrained environment for UAVs, where inference is executed using FP16 precision to maximize power efficiency.

To ensure an equitable comparison, all models were trained from scratch without pretrained weights. The network parameters were updated using the SGD optimizer over 800 epochs with a batch size of 4. The software environment utilized PyTorch and CUDA versions customized to the respective hardware constraints, i.e., CUDA 12.1 for the training workstation and CUDA 11.4 for the edge platform. Details regarding the specific input resolutions and data-augmentation strategies are provided in the subsequent dataset and ablation-study sections.

Table 1. Summary of development and deployment environments.

Model Training Workstation			
CPU	Intel Core i9-14900K	Optimizer	SGD
GPU	NVIDIA RTX 4080	Batch Size	4
CUDA	12.1	PyTorch	2.5.1
Python	3.10	Epochs	800
Edge Inference Platform (UAV Onboard)			
Device	Jetson Orin Nano	CUDA	11.4
JetPack	5.1.3	PyTorch	2.1
Python	3.8	Precision	FP16

6.1.3. Evaluation Metrics

A robust set of quantitative criteria is indispensable for validating the segmentation efficacy and deployment viability. This study employed four principal metrics: pixel accuracy (Acc), intersection over union (IoU), parameter count (Params), and frames per second (FPS). The fundamental elements for computing these metrics are defined as follows:

True positive (TP): Pixels correctly identified as belonging to the target class.

True negative (TN): Pixels correctly classified as background pixels.

False positive (FP): Background pixels erroneously labeled as the target class.

False negative (FN): Target pixels incorrectly classified as the background.

Pixel accuracy (Acc) quantifies the global ratio of correctly classified pixels relative to the total pixel population. It is mathematically expressed as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

The derived forms of this metric include the overall accuracy (aAcc) for the entire dataset and the mean accuracy (mAcc), which calculate the average class-specific accuracies to mitigate class-imbalance bias.

Intersection over union (IoU) provides a more stringent measure of spatial overlap between the predicted segmentation mask and the annotated ground truth. For a specific class, it is expressed as

$$IoU = \frac{TP}{TP + FP + FN} \tag{9}$$

 doi:10.20944/preprints202605.1572.v1

The aggregate metric utilized for model comparison was the mean intersection over union (mIoU), which was obtained by averaging IoU scores across all semantic categories. A higher mIoU value correlates strongly with superior delineation of branch boundaries and morphology.

Parameters (Params) denotes the sum of all trainable weights and biases within the model architecture. This metric serves as a proxy for the memory footprint and computational storage requirements. For UAV-based grasping applications, minimizing the parameter count is critical for enabling efficient inference within the stringent power and memory budgets of embedded flight controllers.

Frames per second (FPS) is the empirical measure of inference throughput and indicates the number of image frames processable by the model per second on the target Jetson AGX Xavier hardware. A high FPS is essential for dynamic visual servoing and real-time grasp adjustments during aerial maneuvers.

6.2. Ablation and Comparative Experiments

6.2.1. Ablation Experiment

Ablation experiments were conducted on both the public UrbanStreet Branch dataset [39] and our self-constructed dataset to evaluate the contribution of each proposed component comprehensively. As described in Section 6.1.1, the original UrbanStreet dataset contains multiple tree categories, which we unified into a single-branch class to align with our task setting. This ensures consistency between the public benchmark and our custom dataset, both of which focus on slender-branch segmentation in complex outdoor environments.

Beginning from the DDRNet baseline, we progressively incorporated the proposed modules, including the GCBlock (replacing RB/RBB), SSPPM (replacing DAPPM), and BOM. The quantitative results are summarized in Table 2, which presents the individual and combined contributions of each component. All the models were evaluated with an input resolution of 1024 × 1024. The FPS was measured on a desktop GPU using PyTorch inference, while latency was reported on an NVIDIA Jetson Orin Nano with TensorRT acceleration (FP16, batch size = 1).

Table 2. Ablation-study results on Drone-Branch and UrbanStreet datasets.

Model			Efficiency			Drone-Branch			UrbanStreet		
a	**b**	**c**	**Params(T/D)/M**	**FPS_{PC}**	**$latency_{Jet}$/ms**	**mIoU**	**mAcc**	**aAcc**	**mIoU**	**mAcc**	**aAcc**
			5.73 / 5.73	127.9	14.135	87.64	94.57	99.00	85.00	91.70	97.12
✓			20.69 / 9.21	149.6	13.734	87.58	94.74	98.97	84.91	91.61	96.83
	✓		5.61 / 5.61	186.9	13.877	89.11	95.53	99.13	85.83	92.38	97.29
		✓	5.73 / 5.73	126.4	14.15	87.91	95.44	98.99	89.20	95.74	97.95
✓	✓		20.57 / 9.09	214.4	13.752	88.43	95.49	99.05	84.95	92.07	96.94
	✓	✓	5.61 / 5.61	188.3	13.953	89.96	95.73	99.12	89.89	95.96	98.11
✓	✓	✓	20.57 / 9.09	221.4	13.769	89.94	95.62	99.05	89.81	95.91	98.09

a: GCBlock, b: SSPPM, c:BOM, Params(T/D): Training and Deployment parameters (M).

Based on Table 2, several clear trends can be observed by comparing representative configurations across the two datasets and hardware platforms. Beginning from the DDRNet baseline, introducing the SSPPM (b) yielded the most significant and consistent improvement in accuracy. On the Drone-Branch dataset, it increased the mIoU from 87.64% to 89.11% (+1.47%), whereas on UrbanStreet, it indicated an improvement by +0.83%. This suggests that the strip-and-serial pooling structure of the SSPPM effectively captures both the close-range, fragmented branch features (0–2 m) in the private dataset and the complex arboreal structures in urban environments. By contrast, the individual contributions of the GCBlock (a) and BOM (c) were relatively limited when used in isolation, thus indicating that multiscale context modeling is the primary factor for improving segmentation e in these challenging scenarios.

 doi:10.20944/preprints202605.1572.v1

When the modules were combined, the interaction between the components revealed the distinct characteristics of the two datasets. For the UrbanStreet dataset, adding the BOM (c) to the baseline or other variants resulted in a significant increase in accuracy (e.g., variant c achieved an mIoU of 89.20%, which represents a +4.2% increase). This is because the UrbanStreet dataset contains high-resolution images with diverse tree species and complex backgrounds, where the boundary-aware supervision of the BOM effectively resolves the semantic ambiguity between branches and urban clutter. On the Drone-Branch dataset, although the single-module gain of the BOM is smaller because of the extremely thin and irregular nature of close-range branches, the combination (bc) achieved the best overall performance (89.96% mIoU, an increase by +2.32% over the baseline). This indicates that the high-level semantics from the SSPPM and the fine-grained edge refinement from the BOM are complementary, thus ensuring both regional consistency and contour sharpness.

In terms of computational efficiency, a significant performance divergence was observed between the PC framework and the embedded Jetson platform. On the desktop GPU using PyTorch, the GCBlock (a) and its combinations (ab, abc) substantially improved the throughput. For instance, the (abc) configuration recorded 221.4 FPS, which is 73.1% higher than that of the baseline. This improvement suggests that the reparameterized design of the GCBlock and the streamlined serial pipeline of the SSPPM effectively simplify the computational graph during PyTorch inference, thus improving memory-access efficiency and reducing kernel overhead. However, on the Jetson Orin Nano, the latency remained relatively stable across all variants (13.7–14.1 ms). This stability is primarily attributed to the deployment-level optimizations of TensorRT. Because TensorRT performs operator fusion and kernel auto-tuning, the structural differences between the training-time multibranch blocks collapse mathematically. Consequently, the final inference speed on the embedded platform becomes more dependent on the hardware peak throughput than on the intermediate graph complexity of the framework.

More importantly, despite these architectural modifications, none of the proposed enhancements introduced additional inference overhead on the embedded platform. All configurations (bc and abc) maintained or even slightly reduced the latency compared with the baseline while achieving higher accuracy. In the most optimized configuration (abc), the model achieved near-peak accuracy on both datasets while remaining within the real-time constraints required for drone-borne operations. These results confirm that the proposed design successfully balances high-precision segmentation for close-range grasping with the stringent efficiency requirements of edge-computing devices.

6.2.2. Model Comparison Experiment

To further evaluate the practical advantages of the proposed abc model, we conducted a comparative study against several representative baselines, including heavyweight models DeepLabV3 [40] and DeepLabV3+ [41] (both with a ResNet-18 backbone), the mobile-oriented MobileNetV3 [42] with an L-RASPP head, and the real-time segmentation model BiSeNetV2 [43]. The trade-offs among segmentation accuracy, inference speed, and model complexity are illustrated in Figure 7.

All the models were evaluated under identical experimental settings on the UrbanStreet dataset with a unified input resolution of 1024×1024. Both the mIoU and aAcc have been reported to provide more comprehensive assessments of segmentation performance.

The results revealed clear differences among the compared methods. DeepLabV3 and DeepLabV3+ achieved strong segmentation performance (mIoU above 91% and aAcc of approximately 98.6%), although their inference speeds remained limited to approximately 40–50 FPS on a desktop GPU. This is primarily because of the computational overhead introduced by atrous spatial pyramid pooling and dense feature processing at high resolutions. Similarly, MobileNetV3 achieved competitive accuracy but exhibited comparable latency constraints. Such performance is generally insufficient for high-speed UAV scenarios, where low latency is critical for safe navigation and interaction tasks.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 25 May 2026 doi:10.20944/preprints202605.1572.v1

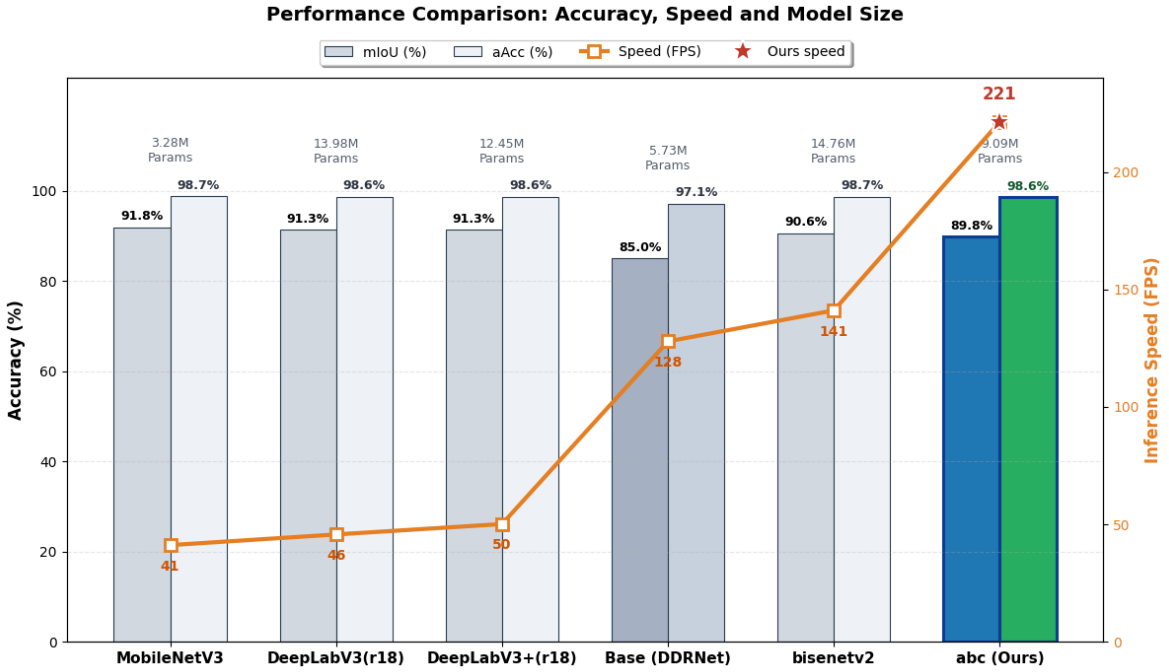

Figure 7. Efficiency comparison of different models.

BiSeNetV2, which is designed for real-time applications, significantly improved the inference speed (over 140 FPS) while maintaining a relatively high accuracy (90.6% mIoU and 98.7% aAcc). However, a clear tradeoff remained between speed and fine-grained segmentation quality, particularly in challenging scenarios involving thin structures.

By contrast, the proposed abc model achieved an inference speed of 221.41 FPS, thereby outperforming all baselines substantially. This represents a speedup exceeding 4.4× over DeepLabV3+ and a 1.5× improvement over BiSeNetV2 while maintaining competitive accuracy (89.81% mIoU and 98.64% aAcc). Despite exhibiting a slightly lower mIoU (approximately 1–2% compared with the strongest baselines), the model preserved a high overall classification accuracy and significantly enhanced the temporal resolution.

As illustrated in Figure 7, the proposed method achieved a favorable balance between efficiency and accuracy, thus positioning itself on the optimal tradeoff frontier for UAV-based applications. By replacing the standard DAPPM with the serial-structured SSPPM and introducing reparameterized GCBlocks, the model improved the feature representation for slender structures while maintaining an extremely high computational efficiency. This design not only supports accurate branch perception but also offers sufficient computational resources for other critical onboard tasks, such as flight control and real-time decision-making on edge devices.

6.3. Edge-Side Visual Validation

To validate the effectiveness of the proposed framework, we present the qualitative results under diverse structural conditions, observation scales, and environmental settings, as shown in Figure 8.

1) **Topology-aware pruning** As shown in Figure 8(a)–(b), multibranch structures introduce numerous junctions and overlapping regions that are unsuitable for UAV grasping. The proposed columnwise topology analysis successfully identified and removed these regions, as highlighted by the shaded red areas. Notably, this behavior remained consistent across different observation scales, thus indicating that the method captures the intrinsic structural properties instead of relying on pixel-level heuristics.

 doi:10.20944/preprints202605.1572.v1

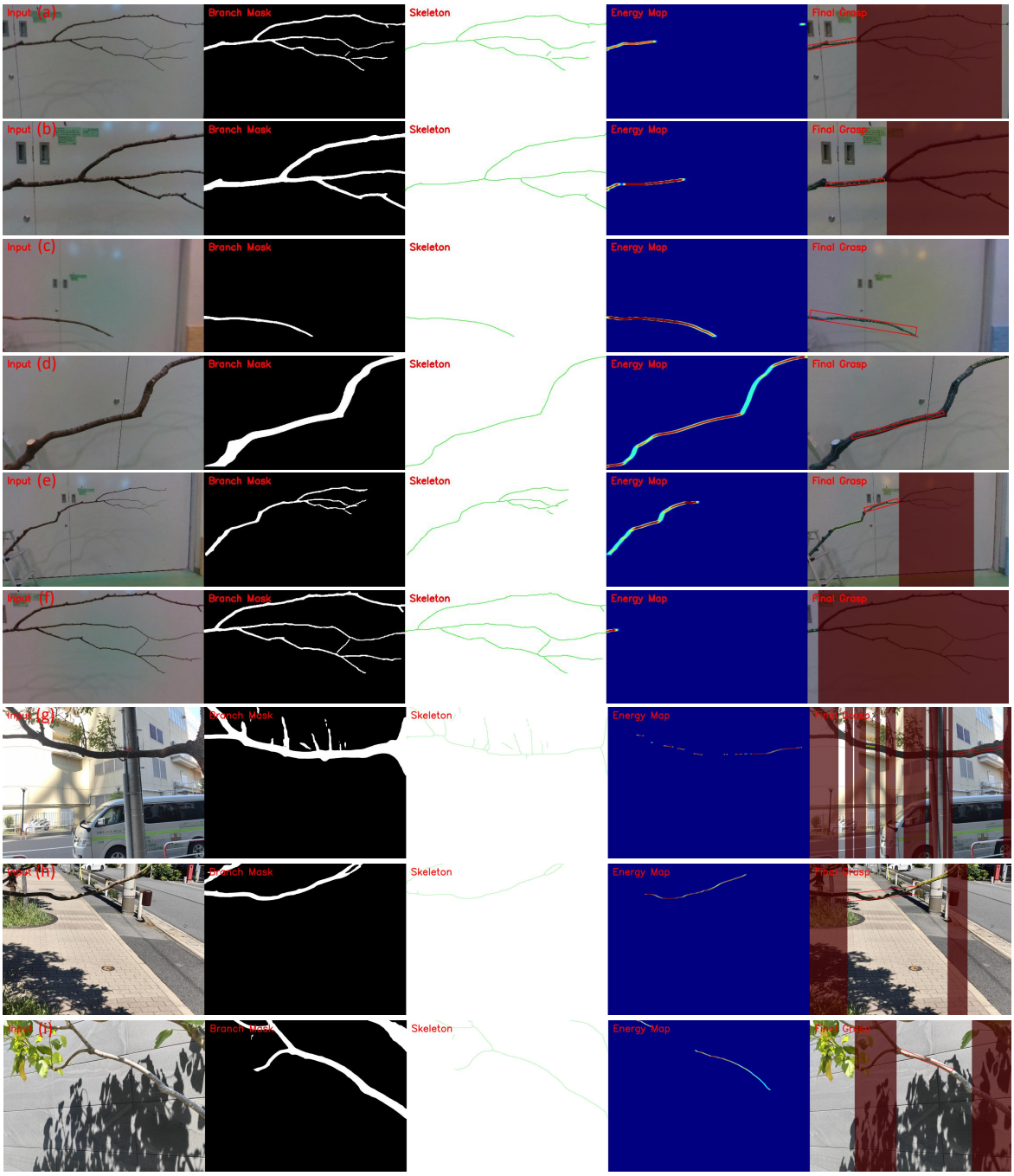

Figure 8. Qualitative results of proposed topology-guided grasp detection framework under different structural conditions. (a)–(b) Multibranch structure under global and closer views, thus demonstrating scale consistency of topology-aware pruning; (c) long straight branch with clear and stable grasp prediction; (d) curved branch where locally consistent segments are successfully detected; (e) full-scene observation with multiple grasp candidates, where the system selects a primary target (red) and secondary candidate (yellow). (f) Short branch segment that does not satisfy the minimum grasp-length constraint, thus resulting in no valid detection; (g)–(i) real-world outdoor scenes featuring natural tree structures, complex lighting, and background foliage, thus demonstrating framework's generalization capabilities beyond controlled indoor environments. The shaded red regions indicate areas removed by the proposed column-wise topology filtering, which correspond to junctions or overlapping structures. The red bounding boxes denote the primary grasp target, whereas the yellow boxes represent secondary candidates.

2) **Structural consistency detection** As shown in Figure 8(c)–(d), the proposed directional convolution effectively detects continuous branch segments under both straight and curved configurations. For straight branches, strong responses are concentrated along the principal direction, whereas for

 doi:10.20944/preprints202605.1572.v1

curved branches, locally consistent segments are preserved. This demonstrates robustness against geometric variations.

3) **Multiple candidate generation and selection** In complex scenes (Figure 8(e)), multiple feasible grasp regions may exist simultaneously. The proposed method generates and ranks multiple candidates based on geometric and physical criteria. The primary grasp target is shown in red, and the secondary candidates are shown in yellow, thus providing additional robustness for dynamic UAV operation.

4) **Physical feasibility constraints** As shown in Figure 8(f), no grasp is generated when the detected branch segment is extremely short, although it is clearly visible. This is due to the minimum-length constraint derived from the UAV gripper requirements. Unlike conventional failures caused by scale or distance, this result reflects an explicit physical constraint instead of a perceptual limitation, thus ensuring that all predicted grasps are executable.

5) **Robustness in real-world outdoor scenarios** To verify the practical applicability of the framework, we extended our evaluation to natural outdoor environments, as shown in Figure 8(g)–(i). Compared with the controlled indoor settings shown in (a)–(f), these real-world scenes present significant perceptual challenges, including cluttered foliage, variable lighting, and complex background interference. Despite these unconstrained conditions, our method successfully isolated the target branches and generated reliable grasp predictions, thus demonstrating the strong generalizability of the proposed visual pipeline for practical UAV operations.

7. Conclusion and Future Endeavors

This paper presented a topology-guided visual post-processing framework for robust grasp-candidate detection in autonomous UAV perching. By combining morphological skeletonization, local-junction suppression, and orientation-aware kernel response mapping, the system effectively translated binary segmentation masks into geometrically safe and physically consistent grasp proposals. Onboard flight experiments in natural forest environments demonstrated that the proposed method reliably filtered structurally hazardous regions, such as branch forks and multi-layer occlusions, while maintaining stable target locking under dynamic UAV motion. The lightweight pipeline operated in real-time on the embedded hardware, thereby establishing a dependable perception-to-decision interface for autonomous perching maneuvers.

Based on the current framework, several directions merit further investigation to enhance the robustness in increasingly complex natural environments.

First, integrating explicit depth gating will strengthen the ability of the system to distinguish between the foreground and reachable branches from visually similar background structures. By confining the topological analysis to a depth interval aligned with the UAV's operational range, the target-selection stability can be improved in densely layered canopies.

Second, the hierarchical navigation from a global tree structure to a specific branch warrants attention. Incorporating a coarse-to-fine attention mechanism would allow the UAV to first localize a suitable tree crown and subsequently refine its focus to an individual graspable branch, thus reducing the search space and improving decision efficiency.

Third, perceptual disturbances caused by partial foliage occlusion should be investigated. Future studies may attempt to identify temporal aggregation strategies and provide depth-discontinuity reasoning to mitigate the effects of transient occlusions and resolve ambiguous cases, such as axially aligned branches that momentarily merge with background masks in a two-dimensional image plane.

Finally, the integration of this perception pipeline with a closed-loop visual servoing controller and its validation on a fully actuated platform with a physical gripper are essential in realizing fully autonomous aerial perching.

Author Contributions: Conceptualization, T.W. and suzuki.S.; methodology, T.W.; software, T.W.; validation, T.W. and ZR.Z.; formal analysis, ZR.Z.; investigation, T.W.; resources, ZR.Z.; writing—original draft preparation, T.W.; writing—review and editing, suzuki.S. All authors have read and agreed to the published version of the manuscript.

 doi:10.20944/preprints202605.1572.v1

Funding: This paper is based on results obtained from a project, JPNP22002, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wu, D.; Yuan, X.; Guan, L. UAV intelligent forest inspection system based on computer vision. In *Proceedings of the 2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA)*, Beijing, China, 29–31 January **2023**; pp. 1150–1154.
2. Li, Q.; Fu, Y.; Qu, S. Research On Forest Resource Supervision Technology Based on Digital Twin UAV. In *Proceedings of the 2024 IEEE 7th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chongqing, China, 8–10 March **2024**; pp. 1327–1330.
3. Bi, Z.; Chi, J.; Zhang, W.J.; et al. A Proposal to Decouple Aerial Manipulation by Multi-Functional End-Effector. In *Proceedings of the 2025 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Location, Date **2025**; pp. 1751–1756.
4. Nekoo, S.R.; Sanchez-Laulhe, E.; Durán, R.G.; et al. Increasing repeatability of the perching on branch for flapping-wing flying robot. In *Proceedings of the 2024 International Conference on Unmanned Aircraft Systems (ICUAS)*, Chania, Greece, 4–7 June **2024**; pp. 618–623.
5. Hu, J.; Chen, P.; Xie, F.; et al. Design and experiment of a sloth-inspired UAV perching climbing grasping mechanism. In *Proceedings of the 2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Jinghong, China, 5–9 December **2022**; pp. 1283–1288.
6. von Frankenberg, F.; Nokleby, S. Detection of long narrow landing features for autonomous UAV perching. In *Proceedings of the 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver, BC, Canada, 4–7 November **2020**; pp. 0565–0570.
7. Liu, Y.; Shen, J.; Zhai, C.; et al. A retinal vessel segmentation network with dual-stage network and vessel pixel emendation. *IEEE Trans. Instrum. Meas.* **2024**, *74*, 1–17.
8. Fernandes, M.; et al. Grapevine Winter Pruning Automation: On Potential Pruning Points Detection through 2D Plant Modeling using Grapevine Segmentation. In *Proceedings of the 2021 IEEE 11th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, Jiaxing, China, 29 July–2 August **2021**; pp. 13–18.
9. Tong, S.; Zhang, J.; Li, W.; et al. An image-based system for locating pruning points in apple trees using instance segmentation and RGB-D images. *Biosyst. Eng.* **2023**, *236*, 277–286.
10. Li, W.; Zhang, J.; Li, J.; et al. Unpaved road segmentation of UAV imagery via a global vision transformer with dilated cross window self-attention for dynamic map. *Vis. Comput.* **2025**, *41*, 1273–1291.
11. Sun, L.; Yang, Y.; Yang, Z.; et al. DUCTNet: An effective road crack segmentation method in UAV remote sensing images under complex scenes. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 12682–12695.
12. Cao, H.; Shen, J.; Zhang, Y.; et al. Proximal cooperative aerial manipulation with vertically stacked drones. *Nature* **2025**, *646*, 576–583.
13. Kaya, Y.F.; Orr, L.; Kocer, B.B.; et al. Aerial additive manufacturing: Toward on-site building construction with aerial robots. *Sci. Robot.* **2025**, *10*, eado6251.
14. Muthusamy, P.K.; Mohiuddin, M.B.; Peringal, A.; et al. Aerial manipulation of long objects using adaptive neuro-fuzzy controller under battery variability. *Sci. Rep.* **2025**, *15*, 10941.
15. Kim, D.; Chang, D.E. An Onboard Integrated Perception and Control Framework for Autonomous Quadrotor UAV Perching on Markerless Hurdles. *Drones* **2026**, *10*, 270.
16. Yin, X.; Wen, S.; Xie, J.; et al. Helical morphology-inspired bistable gripper for UAV upward perching and grasping in field environment. *Bioinspir. Biomim.* **2026**, *21*, 016015.
17. Hamelin, P.; Dandurand, P.; Parkison, S.A.; et al. Shared Autonomy for Safe and Efficient Drone Landing and Takeoff on Power Lines. *IEEE Trans. Field Robot.* **2026**, *in press*.
18. Chen, C.; Yang, M.; Pu, H. Bionic bird claws enable UAV perching and landing. In *Proceedings of the Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, **2025**; Volume 3032, p. 012045.

 doi:10.20944/preprints202605.1572.v1

19. Li, H.; Zhao, Z.; Wu, Z.; et al. Tendon-driven Grasper Design for Aerial Robot Perching on Tree Branches. In *Proceedings of the 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Munich, Germany, 18–24 October **2025**; pp. 21182–21188.
20. Kariyawasam, S.A.; Saikot, M.H.; Cheng, B.; et al. A Hybrid Perching Mechanism for Aerial Robots. *IEEE Robot. Autom. Lett.* **2026**, *in press*.
21. D'Antonio, D.S.; Wu, T.; Bhattacharya, S.; et al. From Hitch to Lift: Autonomous Cable Interlacing by Multi-UAV Teams for Aerial Grasping and Transportation. *IEEE Trans. Robot.* **2026**, *in press*.
22. Yadav, R.D.; Jones, B.; Gupta, S.; et al. An integrated approach to aerial grasping: Combining a bistable gripper with adaptive control. *IEEE/ASME Trans. Mechatron.* **2025**, *in press*.
23. Albaroudi, M.; Alahmad, R.; Alraie, H.; et al. Estimation of Branch Geometry and Hierarchy in Orchard Trees for Robotic Pruning. *J. Robot. Mechatron.* **2026**, *38*, 495–512.
24. Sun, H.; Wu, G.; Xu, H.; et al. Real-time detection and characterization of trunks and upright branches of pear trees for automatic dormant pruning. *Precis. Agric.* **2026**, *27*, 23.
25. Kefalas, A.; Kalampokas, T.; Vrochidou, E.; et al. A vision-based pruning algorithm for cherry tree structure elements segmentation and exact pruning points determination. *Comput. Electron. Agric.* **2025**, *237*, 110735.
26. Dukić, J.; Pejić, P.; Vidović, I.; et al. Towards Robotic Pruning: Automated Annotation and Prediction of Branches for Pruning on Trees Reconstructed Using RGB-D Images. *Sensors* **2025**, *25*, 5648.
27. Tong, S.; Wang, J.; Zhang, J.; et al. An apple tree pruning robot system based on branch segmentation and decision-making control. *Artif. Intell. Agric.* **2026**, *in press*.
28. Li, Y.; Han, J.; Li, H.; et al. Branch-YOLO: An efficient object detector for thin structure objects like pantograph. *Digit. Signal Process.* **2025**, *162*, 105121.
29. Fathi, N. EDFNet: Early Fusion of Edge and Depth for Thin-Obstacle Segmentation in UAV Navigation. *arXiv* **2026**, arXiv:2604.09694.
30. Bilal, H.; Bendechache, M.; Direkoglu, C. Optimized KiU-Net: A Convolutional Autoencoder for Retinal Vessel Segmentation in Medical Images. *IEEE Access* **2025**, *14*, 2784–2799.
31. Fan, M.; Lai, S.; Huang, J.; et al. Rethinking bisenet for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 20–25 June **2021**; pp. 9716–9725.
32. Poudel, R.P.K.; Liwicki, S.; Cipolla, R. Fast-scnn: Fast semantic segmentation network. *arXiv* **2019**, arXiv:1902.04502.
33. Yu, C.; Wang, J.; Peng, C.; et al. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, 8–14 September **2018**; pp. 325–341.
34. Pan, H.; Hong, Y.; Sun, W.; et al. Deep dual-resolution networks for real-time and accurate semantic segmentation of traffic scenes. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 3448–3460.
35. Hou, Q.; Zhang, L.; Cheng, M.M.; et al. Strip pooling: Rethinking spatial pooling for scene parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 13–19 June **2020**; pp. 4003–4012.
36. Yang, G.; Wang, Y.; Shi, D.; et al. Golden cudgel network for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Location, Date **2025**; pp. 25367–25376.
37. Zhen, M.; Wang, J.; Zhou, L.; et al. Joint semantic segmentation and boundary detection using iterative pyramid contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 13–19 June **2020**; pp. 13666–13675.
38. Lin, T.Y.; Goyal, P.; Girshick, R.; et al. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 22–29 October **2017**; pp. 2980–2988.
39. Yang, T.; Zhou, S.; Huang, Z.; Xu, A.; Ye, J.; Yin, J. Tree Dataset of Urban Street: Branch. Available online: https://ytt917251944.github.io/dataset_jekyll/ (accessed on 12 May 2026).
40. Chen, L.C.; Papandreou, G.; Schroff, F.; et al. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
41. Chen, L.C.; Zhu, Y.; Papandreou, G.; et al. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, 8–14 September **2018**; pp. 801–818.

 doi:10.20944/preprints202605.1572.v1

42. Howard, A.; Sandler, M.; Chu, G.; et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea, 27 October–2 November **2019**; pp. 1314–1324.
43. Yu, C.; Gao, C.; Wang, J.; et al. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *Int. J. Comput. Vis.* **2021**, *129*, 3051–3068.