electronics

MDPI

Article

A Lightweight Three-Keypoint Recognition Method for Line-Laser-Based Robot Vision-Guided Adhesive Dispensing

Chuanyi Cui [1,2,3,4], Changzhen Dong [5], Chao Chen [6], Wenping Liu [5] and Fuxin Du [5,*]

[1] Technology Innovation Center of Electromagnetic Functional Structure, Jinan 250023, China
[2] Shandong Province Key Laboratory for Electromagnetic Control and Multifunctional Integration Technology of Aerospace Electromagnetic Functional Structure, Jinan 250023, China
[3] The Aeronautical Science Key Lab for High Performance Electromagnetic Windows, Jinan 250023, China
[4] AVIC Research Institute for Special Structures of Aeronautical Composites, Jinan 250023, China
[5] School of Mechanical Engineering, Shandong University, 17923 Jingshi Road, Jinan 250061, China; 202534517@mail.sdu.edu.cn (C.D.)
[6] School of Rail Transportation, Shandong Jiaotong University, Jinan 250357, China
* Correspondence: dufuxin@sdu.edu.cn

Abstract

Line-laser stripes in robot vision-guided adhesive dispensing exhibit narrow local structures, while keypoint localization is highly sensitive to spatial information and online processing imposes stringent requirements on computational efficiency. To address these challenges, this paper proposes a lightweight three-keypoint recognition method for line-laser-based adhesive dispensing. First, a two-stage top-down vision pipeline is adopted: an object detector locates the region of interest (ROI), after which the left, center, and right points are predicted within the local ROI. The keypoint network uses RepViT-M0.6 as a lightweight feature extraction backbone. To compensate for the low spatial resolution of its final-stage features, LiteFlattenSimCCHead is designed to transform deep features into position-sensitive representations better suited to keypoint localization through low-dimensional feature transformation, progressive spatial reconstruction, lightweight local feature fusion, and position-preserving feature flattening. SimCC coordinate classification and DARK distribution-aware decoding are then combined to obtain the coordinates of the three keypoints. Experimental results show that the proposed method achieves an AP of 0.9532, a mean localization error of 3.6385 px, an RMSE of 4.8272 px, and a PCK@5 of 0.7842. Compared with the Lite-HRNet-18 baseline, the mean error and RMSE are reduced by 16.5% and 20.1%, respectively, while AP increases by 3.80 percentage points. For single-ROI inference in the second stage, the final model reaches 116.34 FPS on the GPU and 49.32 FPS on the CPU. These results demonstrate that the proposed method improves localization accuracy while maintaining a lightweight model and high inference efficiency. Therefore, the reported results demonstrate the performance of the vision and keypoint-recognition method, rather than the positioning accuracy of the complete robotic adhesive-dispensing system.

Keywords: robotic adhesive dispensing; line-laser vision; three-keypoint recognition; RepViT; SimCC; lightweight network

Check for updates

Academic Editor: Giuseppe Prencipe

Received: 2 September 2026
Revised: 12 October 2026
Accepted: 14 October 2026
Published: 16 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

As the automotive, aerospace, and large-equipment manufacturing industries move toward flexible, automated, and intelligent production, adhesive bonding and sealing

processes are being widely used for structural joining, corrosion protection, waterproofing, and airtightness assurance. Compared with manual dispensing, industrial robots offer high motion repeatability, stable trajectory execution, and strong continuous-operation capability, and have therefore become important platforms for automated adhesive dispensing and sealing. However, conventional teach-and-repeat robots generally rely on preplanned fixed trajectories and relatively stable workpiece fixturing. When the position, pose, or local geometry of a workpiece changes, trajectories generated offline or taught manually may no longer correspond accurately to the actual dispensing region. Using visual information to determine the position and local geometric characteristics of the target region is therefore essential for improving the environmental adaptability of robotic dispensing systems and enabling online trajectory adjustment.

Machine vision has long been applied to automated adhesive dispensing and sealing. Davidor et al. [1] used structured-light vision to monitor adhesive beads online during deposition, demonstrating the feasibility of structured light for bead-quality perception. Sun et al. [2] further investigated methods for obtaining bead shape and related geometric parameters from visual information and discussed the application of visual inspection to adhesive deposition control. Davies et al. [3] developed a robotic dispensing cell comprising a six-axis robot, an automatic dispensing device, and an online vision inspection system. Bead information obtained by vision was fed back to the control system, enabling online monitoring and control of the dispensing process. These studies show that vision sensing can effectively support geometric-information acquisition and quality control in robotic adhesive dispensing.

With advances in three-dimensional vision and robotic perception, research has gradually expanded from adhesive-bead quality inspection to workpiece localization, seam recognition, and robot trajectory generation. Maiolino et al. [4] integrated an RGB-D sensor with a robot offline-programming system, dynamically updated the work-environment model using visual information, and generated robot sealant-dispensing programs, thereby improving adaptability to changes in parts and the environment. Hanh et al. [5] used laser triangulation and image processing to obtain complex two- and three-dimensional seam curves and transformed visual coordinates into the robot coordinate system, allowing the robot to perform sealing according to the actual seam position. Pereira et al. [6] investigated a force- and vision-based monitoring method for robotic sealing, using an end-effector force sensor to acquire contact forces during dispensing and machine vision to monitor bead width and texture. Prezas et al. [7] developed a robotic perception system for adhesive application to large components that integrates workpiece recognition, localization, process monitoring, and quality inspection, and used it for closed-loop robot control. Collectively, these studies have advanced robotic adhesive dispensing from fixed-trajectory execution toward vision-based perception and online adjustment.

Line-structured light obtains local geometric information by analyzing changes in the shape of a laser stripe on a workpiece surface. Its non-contact operation, well-defined structure, and convenient local-profile extraction make it valuable for industrial vision measurement. Qiao et al. [8] proposed a high-resolution three-dimensional inspection method based on line-structured light for aerospace sealant quality assessment, enabling automatic measurement of sealant thickness, width, positional accuracy, and the profile of lap regions. Sarti et al. [9] systematically reviewed vision inspection, data analytics, and digital technologies in robotic sealing, showing that research has expanded from conventional two-dimensional vision to deep learning, three-dimensional sensing, and data-driven process perception, covering trajectory planning, process monitoring, and quality inspection. Existing studies have thus focused mainly on complete bead-quality evaluation, three-dimensional profile measurement, and seam-trajectory extraction. By contrast, direct

localization of local geometric keypoints in line-structured-light images captured at a robot end effector, followed by the use of those keypoints to describe the position and orientation of the target dispensing region, remains relatively underexplored.

Online structured-light vision for weld seams is methodologically related to this problem. Although welding and adhesive dispensing are different manufacturing processes, both can use laser-stripe deformation over local geometric structures to obtain seam information. Accordingly, methods for laser-stripe and feature-point extraction provide useful references for line-laser-based visual localization in adhesive dispensing. To recognize weld seams under intense noise from arc light and spatter, Wang et al. [10] combined Laplacian-of-Gaussian filtering, a NURBS-Snake model, and geometric fitting to extract the structured-light centerline and weld features. Chen et al. [11] recognized and localized fillet welds through weld-likelihood calculation, candidate-region screening, and local structural features, avoiding explicit computation of the structured-light centerline and improving adaptability under complex interference. More recently, Chen et al. [12] used a deep neural network to suppress complex background interference and extract laser stripes, and then located weld feature points through contour tracking. These studies indicate that, compared with reconstructing an entire stripe or contour, stable extraction of local feature points with explicit geometric meaning can also provide effective visual information for subsequent robot localization and motion guidance.

The development of deep learning has introduced new approaches to keypoint localization in complex industrial environments. Compared with conventional methods that rely on threshold segmentation, centerline extraction, and handcrafted geometric rules, data-driven models can learn local structures and their spatial relationships directly from images. Yu et al. [13] proposed Lite-HRNet, which preserves high-resolution representations through lightweight designs such as conditional channel weighting and maintains strong keypoint localization performance while reducing computational complexity. Nevertheless, robot online-vision tasks must consider not only localization accuracy but also inference efficiency and practical deployment cost. Wang et al. [14] proposed RepViT, revisiting mobile convolutional networks from the design perspective of lightweight Vision Transformers. RepViT achieves favorable computational efficiency and low latency across multiple vision tasks, providing a new option for lightweight vision models in resource-constrained scenarios.

In addition to the feature extraction network, the representation and decoding of keypoint coordinates affect final localization accuracy. Conventional two-dimensional heatmap methods represent keypoint responses in a feature space of limited resolution, and coordinate discretization may introduce quantization errors. Li et al. [15] proposed SimCC, which reformulates two-dimensional keypoint localization as two one-dimensional coordinate-classification problems along the horizontal and vertical axes and enables finer-grained position prediction through coordinate discretization. Zhang et al. [16] systematically analyzed heatmap-based keypoint-coordinate representation and its encoding-decoding process and proposed Distribution-Aware Coordinate Representation of Keypoints (DARK). DARK reduces localization bias introduced by coordinate quantization and decoding through distribution-aware coordinate decoding and more accurate heatmap encoding. These studies provide an important methodological basis for precise coordinate prediction with lightweight networks.

However, directly applying general-purpose keypoint-recognition models to robot line-laser adhesive-dispensing scenarios has several limitations. First, line-laser stripes are narrow and locally concentrated. Keypoints are typically located in local regions with pronounced contour changes, and repeated downsampling can weaken spatial responses associated with precise position. Second, although increasing feature resolution benefits

localization, complex high-resolution prediction structures increase the computational burden and are therefore unfavorable for online robot vision. Moreover, the multiple keypoints recognized in this study are used to describe local position and orientation relationships in the dispensing region. The model must therefore achieve not only low individual-point localization error but also stable geometric relationships among keypoints. Balancing lightweight feature extraction with precise keypoint localization is consequently the central problem addressed in this work.

Based on the above analysis, this study develops a lightweight three-keypoint recognition method for line-laser-based robot vision-guided adhesive dispensing. The main contributions of this work are summarized as follows:

(1) A task-oriented three-keypoint representation is established for line-laser-based robotic adhesive dispensing. The center keypoint directly represents the target dispensing position, while the left and right keypoints provide auxiliary geometric constraints for estimating the local orientation and pose-related structure. This representation connects image-based keypoint localization with the geometric requirements of adhesive dispensing.

(2) LiteFlattenSimCCHead is implemented as the core lightweight localization module of this study. It addresses the mismatch between the low-resolution features produced by RepViT-M0.6 and the fine spatial information required by narrow line-laser structures. By reconstructing position-sensitive features before coordinate prediction, the proposed head improves the localization capability of the lightweight keypoint network without replacing the lightweight backbone.

(3) A controlled experimental protocol is established to distinguish the contributions of the prediction head and coordinate decoding. The effect of LiteFlattenSimCCHead is evaluated by replacing Standard SimCCHead under the same RepViT-M0.6 backbone, while the effect of DARK is evaluated using the same Model B checkpoint with only the decoding procedure changed. The final configuration achieves an AP of 0.9532, a mean localization error of 3.6385 px, an RMSE of 4.8272 px, and a PCK@5 of 0.7842.

2. Related Work

Existing studies related to this work mainly involve vision-guided robotic adhesive dispensing, line-laser perception, and deep-learning-based keypoint localization. Vision-guided dispensing methods use visual information to locate target regions, reconstruct local contours, generate dispensing paths, or provide feedback for process control. Line-laser sensors further provide active profile information for identifying local geometric transitions on the workpiece surface. However, these studies generally focus on object-level localization, contour extraction, or complete trajectory generation. The use of a small number of task-oriented keypoints to jointly describe the target dispensing position and its local geometric orientation remains insufficiently investigated.

Deep-learning-based keypoint methods provide a feasible approach for learning the relationship between image appearance and geometric position. Heatmap-based methods represent keypoint locations using two-dimensional spatial response maps, whose computational and memory costs depend on the output resolution. Coordinate-classification methods instead decompose two-dimensional localization into independent classification problems along the horizontal and vertical axes, providing an alternative representation for fine-grained localization. Lightweight backbones can reduce feature-extraction costs, but their repeated downsampling produces low-resolution deep features that may weaken the local positional information required for narrow line-laser structures. Consequently, directly combining a lightweight backbone with a conventional prediction head may not

provide an appropriate balance between computational efficiency and fine-grained localization accuracy.

The present work addresses this gap through a task-oriented three-keypoint representation for line-laser-based robotic adhesive dispensing. The center keypoint represents the target dispensing position, while the left and right keypoints provide auxiliary geometric constraints for describing the local orientation and pose-related structure. Unlike generic human-pose or object-keypoint configurations, the semantic definitions of these keypoints are determined by the geometric requirements of the dispensing task.

The central technical contribution is LiteFlattenSimCCHead, which is implemented to adapt the low-resolution final-stage features of RepViT-M0.6 to the fine localization of narrow line-laser structures. Rather than directly predicting coordinates from the low-resolution backbone output, the head transforms the deep features into a position-sensitive representation before axial coordinate classification. RepViT-M0.6, SimCC, and DARK are adopted as existing techniques; the novelty of this work lies in the task-oriented three-keypoint representation, the LiteFlattenSimCCHead designed for this representation, and their integration within a two-stage line-laser dispensing pipeline. The contribution of the prediction head is evaluated independently under a fixed RepViT-M0.6 backbone, while the contribution of DARK is evaluated using the same trained checkpoint with only the decoding procedure changed.

3. Experimental Platform and Dataset

3.1. Robotic Line-Laser Adhesive-Dispensing Experimental Platform

To acquire line-laser images under realistic dispensing conditions, a robot vision-guided adhesive-dispensing experimental platform was constructed, as shown in Figure 1. The platform mainly comprises an industrial robot, an adhesive-dispensing unit, a line-laser vision sensor, and a host computer. The line-laser vision sensor and the dispensing actuator are mounted on the robot end effector and acquire local-profile images of the target dispensing region while the robot moves.

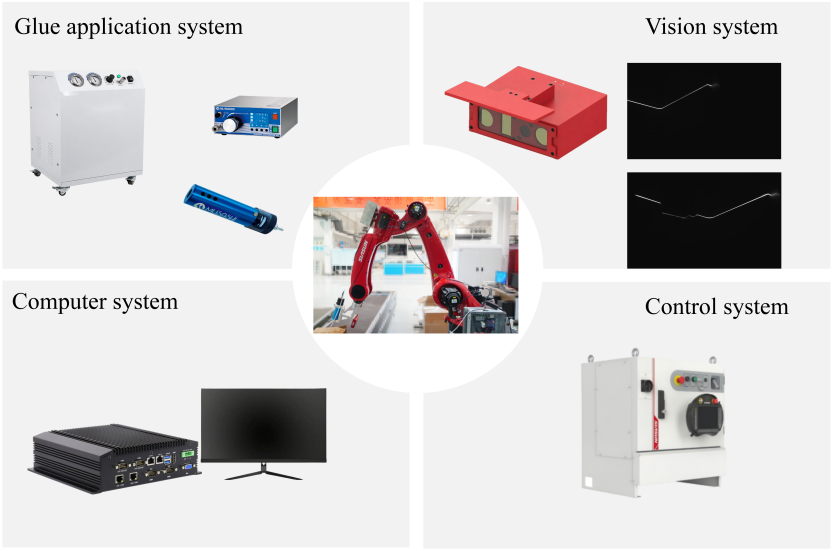


Figure 1. Overall experimental platform comprising the robotic arm, line-laser sensor, and adhesive-dispensing gun.

A Siasun SR25A-20/1.80 six-axis industrial robot (SIASUN Robot & Automation Co., Ltd., Shenyang, China) was used. It has a maximum payload of 20 kg, a maximum reach of 1803 mm, and a repeatability of ±0.05 mm, satisfying the requirements for mounting

the vision sensor and dispensing end effector and for executing continuous trajectories. The dispensing unit consists of a dispensing controller and an end-effector material-delivery component. The ML-6000X-CTR-N-GP dispensing controller (MUSASHI ENGINEERING, INC., Tokyo, Japan) regulates adhesive output through pneumatic pulses and supports RS-232C communication and external I/O control. During experiments, the controller regulates adhesive output according to the specified pressure and coordinates with robot motion to complete the dispensing operation.

The vision unit uses a PSN0270 line-laser sensor (Wuxi V-Sensor Technology Co., Ltd., Wuxi, China) with a central laser wavelength of 660 nm, a Z-direction accuracy of 0.03–0.06 mm, and a maximum acquisition rate of 60 fps. During an experiment, the line laser is projected onto the workpiece surface. Changes in the local geometric profile deform the laser stripe, and the sensor captures the corresponding grayscale image and transmits it to the host computer for subsequent target-region extraction and keypoint recognition.

The system's visual-processing pipeline comprises target-region localization, three-keypoint recognition, and geometric-information calculation. This study focuses on accurate localization of keypoints within the target region to provide visual information for subsequent determination of the dispensing position, estimation of local orientation, and calculation of robot motion parameters.

3.2. Data Acquisition and Definition of the Three Keypoints

Data were collected using a line-laser camera mounted on the robotic arm. The experiments were conducted on a single workpiece, and the robotic arm moved along a predefined scanning trajectory while the line-laser sensor observed the workpiece from different viewpoints and trajectory positions. During acquisition, the laser line was projected onto the workpiece surface, and the camera synchronously captured image sequences containing local profile information. Representative keyframes were then extracted from the acquired sequences to construct the dataset. All images had an original resolution of 1920×1200 pixels.

For each target dispensing instance, three keypoints were annotated according to the local geometric structure shown in Figure 2. The center keypoint P_C denotes the target adhesive dispensing point and provides the primary positional reference for the robotic tool. The left and right keypoints, denoted by P_L and P_R, are auxiliary pose-reference points located on the two sides of P_C. Their relative positions with respect to P_C describe the local lateral arrangement and directional geometry required for estimating the orientation of the dispensing tool. Therefore, the triplet (P_L, P_C, P_R) represents the target dispensing position together with the local pose-reference geometry, rather than three independent adhesive points.

The annotation was based on the geometric structure associated with the dispensing target rather than on an arbitrary visible stripe boundary. When the laser stripe was broadened, the keypoint was placed at the geometric center of the corresponding target structure. For rounded transitions, the keypoint was assigned to the intended geometric transition location instead of the beginning or end of the blurred transition region. When reflections or isolated secondary stripes were present, the annotation followed the continuous stripe structure connected to the target dispensing geometry, whereas isolated reflection artifacts were not treated as valid keypoints.

The final dataset contains 4460 images and 5656 annotated target instances. The selected keyframes were randomly partitioned at the image level using the dataset-splitting procedure implemented in the code. The training, validation, and test subsets contain 3568, 445, and 447 images, respectively, with 4525, 564, and 567 annotated target instances. Because all images were acquired from the same workpiece and neighboring viewpoints

along the scanning trajectory may produce similar laser-stripe patterns, the samples may still exhibit visual and geometric correlation. The current annotation files do not preserve explicit sequence or acquisition-session identifiers. Therefore, the present results should be interpreted as an image-level random-split evaluation on one workpiece rather than as a strictly independent-sequence or workpiece-independent generalization evaluation.

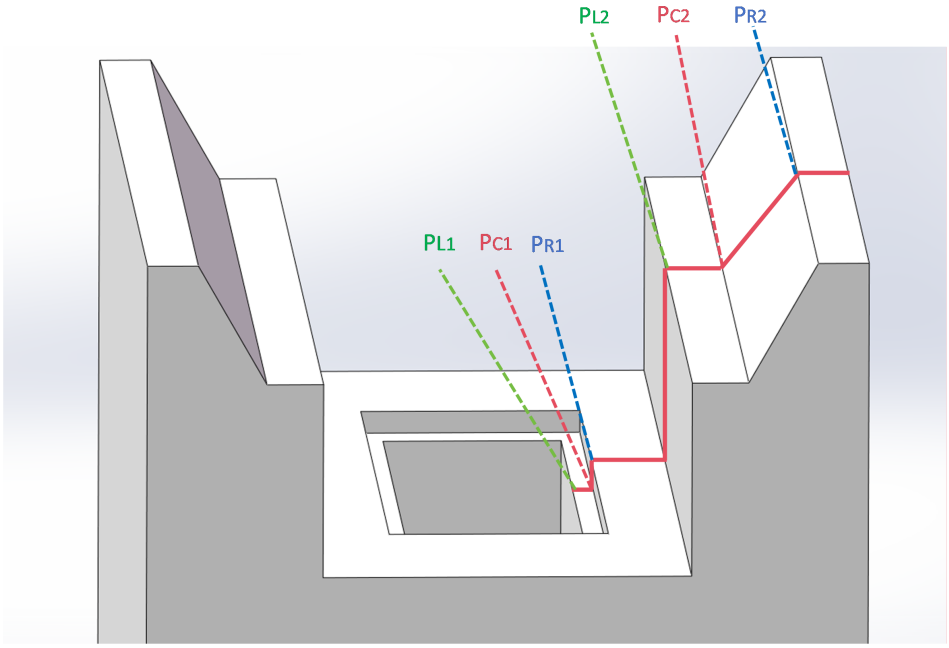


Figure 2. Definition of the three keypoints.

3.3. Two-Stage Top-Down Keypoint Recognition and Data Preprocessing

A two-stage top-down visual-processing pipeline is used to localize the three keypoints in the line-laser adhesive-dispensing region. In the first stage, a YOLO26n one-stage object detector is applied to each original line-laser image to localize the target dispensing regions and generate candidate bounding-box ROIs. The detector was initialized from the pretrained `yolo26n.pt` weights and trained using the annotated target-region bounding boxes. The bounding boxes were defined as the envelope of the three annotated keypoints expanded by 64 pixels on each side in the original image, thereby preserving the local structural context required by the second-stage keypoint model. The detector was trained with an input resolution of 640 × 640, a batch size of 16, and a maximum of 400 epochs. The training configuration used automatic optimizer selection, an initial learning rate of 0.01, cosine learning-rate decay, a momentum of 0.937, a weight decay of 0.0005, and a three-epoch warm-up period. Early stopping was configured with a patience of 50 epochs, and the random seed was set to 42. The saved training record ended at epoch 106. The best detector checkpoint was used to generate the validation and test ROIs.

The first-stage detector was evaluated separately on the test subset. Its standard object-detection performance and the ROI coverage used by the second-stage keypoint model are summarized in Table 1.

Table 1. Test-set detection performance and ROI-matching statistics of the first-stage YOLO26n detector.

Detector	Precision	Recall	mAP@0.5	mAP@ 0.5:0.95	GT	Filtered	Matched	Missed GT	Unmatched	Evaluated	Mean IoU
YOLO26n	0.9940	1.0000	0.9950	0.8919	567	574	567	0	7	567	0.9316

The test subset contains 447 images and 567 ground-truth target instances. At the detector-evaluation confidence threshold of 0.001, the first-stage YOLO26n detector achieved a precision of 0.9940, a recall of 1.0000, an mAP@0.5 of 0.9950, and an mAP@0.5:0.95 of 0.8919. Before constructing the second-stage ROIs, detections with confidence scores below 0.25 were removed, leaving 574 candidate boxes.

Using greedy one-to-one matching within each image at IoU $\geq$ 0.5, 567 detections were matched to all 567 ground-truth instances, with a mean matched IoU of 0.9316. Therefore, no ground-truth instance was missed or excluded from the matched-ROI keypoint evaluation. Seven additional detections remained unmatched and were treated as first-stage false-positive detections; they were not included in the keypoint localization averages. Consequently, Mean, RMSE, Center, Slope, Arm, and PCK@5 were calculated from 567 matched prediction–ground-truth pairs.

In the second stage, each retained ROI is standardized and passed to the keypoint-recognition network, which predicts the left point P_L, center point P_C, and right point P_R. By separating full-image target search from local keypoint localization, the second-stage network can focus on learning the laser stripe and local structural information within its neighborhood. The overall process can be expressed as follows:

$$I \xrightarrow{D} B^{\mathrm{det}} \xrightarrow{T} I^{\mathrm{ROI}} \xrightarrow{K} (P_{\mathrm{L}}, P_{\mathrm{C}}, P_{\mathrm{R}}) \quad (1)$$

where I denotes the original line-laser image, D(.) is the first-stage object detector, B_det is the candidate region, T(.) denotes ROI cropping and scale transformation, I_ROI is the standardized local image, and K(.) is the second-stage keypoint-recognition network.

During training, ROIs are generated from manually annotated bounding boxes to prevent first-stage detection errors from introducing additional interference into keypoint-network learning. Moderate position shifts, scale changes, and rotation augmentations are applied to the annotated boxes, and affine transformation maps each ROI to a 256 $\times$ 256 network input. This processing preserves the principal local structure of the line-laser stripe while enabling the model to accommodate moderate changes in ROI position and scale. The input image is then normalized, and supervision labels for the three keypoints are generated according to the SimCC coordinate representation.

During validation and testing, predicted boxes output by the first-stage YOLO26 detector are used as the ROI source to better approximate the complete vision pipeline during online robot operation. During detection-result preparation, the raw detector confidence threshold is set to 0.001 and the non-maximum suppression IoU threshold is set to 0.7. The resulting boxes are then filtered using the MMPose bounding-box score threshold of 0.25. The retained candidate boxes are converted to fixed-size inputs using the same ROI scaling and affine mapping as in training, but without random geometric augmentation. Thus, test results reflect errors from both first-stage region localization and second-stage keypoint localization and can be used to evaluate the practical recognition performance of the complete two-stage system. The test set contains 567 annotated target instances. According to the saved detector-preparation summary, YOLO26 generated 1246 raw detections on the test set. After confidence filtering, 574 predicted boxes remained. Using an IoU threshold of 0.5 for one-to-one matching, 567 predicted boxes were matched to the 567 ground-truth instances, resulting in a matched-instance recall of 1.0 and a mean matched IoU of 0.931551. Seven filtered predicted boxes remained unmatched, whereas no ground-truth instance was unmatched.

For images containing multiple target instances, one-to-one correspondences are established between predicted and ground-truth boxes based on their intersection over union:

$$\mathrm{IoU}(B_j^{\mathrm{det}}, B_i^{\mathrm{gt}}) = \frac{B_j^{\mathrm{det}} \cap B_i^{\mathrm{gt}}}{B_j^{\mathrm{det}} \cup B_i^{\mathrm{gt}}} \tag{2}$$

Only candidate matches with IoU ≥ 0.5 are retained. Candidate pairs are sorted in descending order of IoU, and greedy one-to-one correspondences are established within each image. Once a predicted ROI or a ground-truth instance has been matched, it cannot be reused. Localization metrics, including Mean, RMSE, Center, Slope, Arm, and PCK@5, are calculated only from successfully matched prediction–ground-truth instance pairs.

An unmatched predicted ROI is treated as a false-positive detection and is excluded from the matched-instance localization averages. An unmatched ground-truth instance is treated as a missed detection and is also excluded from the matched-instance pixel-error averages. These detection events are not converted into artificial pixel localization errors and are reported separately through the matching statistics. In the present test set, all 567 ground-truth instances were matched, so the reported localization metrics were calculated from 567 matched instances, while the seven unmatched predicted boxes were not included in these averages.

All compared second-stage keypoint-recognition models use the same YOLO26-generated validation and test ROI files, namely `yolo26_val_bbox.json` and `yolo26_test_bbox.json`. The detector checkpoint, detector-generated boxes, MMPose bounding-box score threshold, ROI transformation, and input size are kept unchanged across the compared methods. Therefore, the comparison is performed under a common first-stage ROI source and focuses on the second-stage keypoint-recognition models.

In summary, a two-stage recognition pipeline is formed from target-region detection and ROI standardization to final three-keypoint localization. Annotated ROIs are used during training to ensure stable learning by the keypoint network, whereas actual detected ROIs are used for validation and testing. This design combines stable model training with realistic evaluation under practical deployment conditions.

4. Lightweight Three-Keypoint Recognition Method for Line-Laser Images

4.1. Overall Framework

Compared with general keypoint-detection tasks, keypoints in robotic adhesive dispensing have explicit local geometric meanings. The target dispensing region is represented by the center point P_C, left point P_L, and right point P_R. Coordinate errors at any individual keypoint not only affect its own localization but also propagate to calculations of local orientation and structural geometry. The task therefore requires high coordinate accuracy. Meanwhile, the stripe formed by the line laser on a workpiece surface is generally narrow, with useful geometric information concentrated in a local region, making spatial details susceptible to loss during repeated network downsampling. Because the vision system is ultimately intended for online robotic adhesive dispensing, the network must balance localization accuracy and computational efficiency.

To address both the narrow local structure of the line-laser stripe and the requirements for computational efficiency and localization accuracy in online robot vision, a lightweight three-keypoint recognition network is constructed, as shown in Figure 3. The network mainly comprises a RepViT-M0.6 feature extraction backbone, LiteFlattenSimCCHead, SimCC coordinate classification, and DARK refined decoding. RepViT-M0.6 efficiently extracts local structural features from the line-laser image. LiteFlattenSimCCHead transforms low-resolution deep features into position-sensitive representations more suitable for keypoint localization. SimCC and DARK then predict the coordinates of the three keypoints.

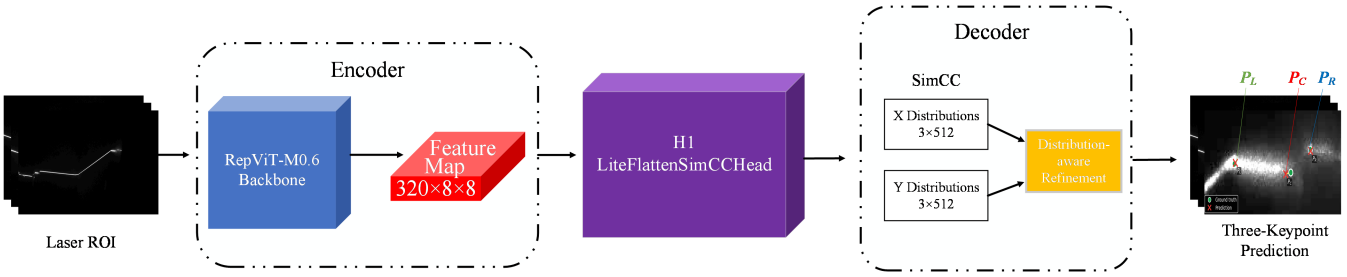


Figure 3. Overall architecture of the lightweight three-keypoint recognition network for line-laser images.

As described above, the line-laser ROI obtained in the target-detection stage is first resized to 256 × 256 pixels and then input to the RepViT-M0.6 backbone for feature extraction. Deep features are obtained after multistage downsampling. RepViT-M0.6 has low computational complexity, but the small final-stage feature map does not directly preserve local positional variations in a narrow laser stripe. LiteFlattenSimCCHead is therefore placed after the backbone. Through channel compression, progressive recovery of spatial resolution, and lightweight local feature fusion, the head transforms low-resolution deep features into keypoint features with richer spatial information and then flattens them into one-dimensional feature representations for coordinate prediction.

During coordinate prediction, SimCC models the X and Y coordinates of each of the three keypoints as classification problems, yielding the corresponding one-dimensional coordinate probability distributions. Compared with directly regressing keypoint coordinates from a low-resolution two-dimensional feature map, this approach decomposes spatial localization into two independent one-dimensional coordinate-classification problems. DARK is subsequently introduced to refine the coordinate-distribution peaks without changing the front-end feature extraction results, yielding the final predicted coordinates of the center point P_C, left point P_L, and right point P_R.

4.2. RepViT-M0.6 Lightweight Backbone

A robotic vision-guided adhesive-dispensing system must continuously process line-laser images while the robotic arm is moving. The keypoint-recognition model must therefore provide both high localization accuracy and fast inference. To support subsequent online visual processing, RepViT-M0.6 is introduced as a lightweight feature extraction backbone, as shown in Figure 4, to reduce network complexity and improve feature extraction efficiency.

RepViT-M0.6 extracts hierarchical features in stages. For a preprocessed line-laser ROI, Patch Embedding first performs initial feature mapping and reduces the feature size. Four subsequent stages progressively decrease the spatial resolution while increasing the number of channels. This hierarchy controls the size of intermediate feature maps while gradually enlarging the receptive field. For the line-laser images considered here, shallow features mainly retain stripe position and local contour changes. As network depth increases, the features gradually integrate structural information over a wider range, providing a feature basis for jointly recognizing the center, left, and right keypoints.

The lightweight design of RepViT mainly arises from its low-cost structures for spatial feature mixing and channel feature transformation. A standard RepViT Block uses RepVGGDW to extract local spatial information and a Channel Mixer comprising 1 × 1 convolutions to exchange information across channels. Between stages, a Downsampling RepViT Block reduces feature resolution and expands the channels. Compared with network structures that continuously maintain high-resolution features, this approach reduces computation during deep feature extraction.

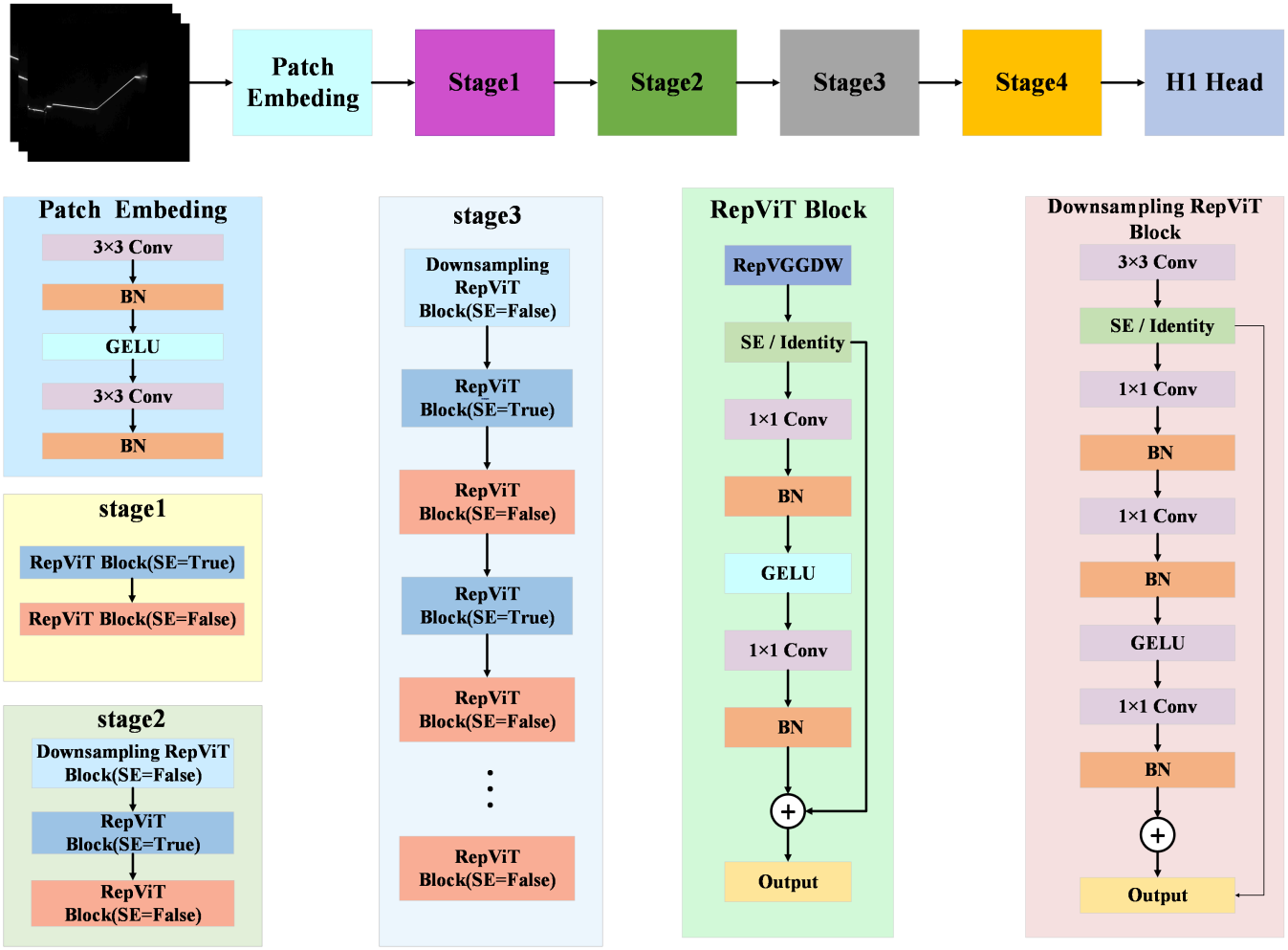


Figure 4. RepViT-M0.6 lightweight backbone and feature sizes at four stages.

In addition, RepVGGDW employs structural reparameterization. Multiple branches are used for feature learning during training, whereas the corresponding branches can be equivalently fused into a single convolution during inference, simplifying the computational path during deployment. The equivalent convolution kernel is expressed as

$$K_{\text{rep}} = K_{3\times3} + \text{Pad}(K_{1\times1}) + K_{\text{id}} \tag{3}$$

This separation between training and inference structures preserves the feature-representation capability of the training architecture while reducing operator complexity during actual inference, which is consistent with the real-time computational requirements of robotic vision-guided adhesive dispensing.

After feature extraction by the RepViT-M0.6 backbone, a deep feature map of $320 \times 8 \times 8$ is obtained. Although the lightweight backbone reduces the computational cost of feature extraction, repeated downsampling substantially lowers the spatial resolution of the final-stage features. For localization of the three keypoints on a line-laser stripe, such low spatial resolution is unfavorable for recovering subtle positional differences around the keypoints. LiteFlattenSimCCHead is therefore designed after RepViT-M0.6 to perform lightweight spatial recovery and coordinate representation of the deep features, mitigating the adverse effect of low-resolution final-stage features on precise keypoint localization.

4.3. LiteFlattenSimCCHead

RepViT-M0.6 obtains deep features with strong structural representation through progressive downsampling while effectively reducing computational cost. However, in the robot line-laser keypoint-localization task considered here, the low spatial resolution of the backbone's final-stage features is mismatched with the position representation required for precise keypoint localization. Line-laser stripes are typically narrow. Determination of the left point P_L, center point P_C, and right point P_R depends not only on the overall contour shape but also on stripe bends, local orientation changes, and relative spatial relationships within the keypoint neighborhoods. Consequently, coordinate prediction based only on low-resolution deep features cannot fully exploit local positional responses.

To address this issue, LiteFlattenSimCCHead is designed after RepViT-M0.6. Rather than reconstructing high-resolution image features, its purpose is to transform the deep structural information output by the backbone into a position-sensitive representation suitable for coordinate localization at low computational cost. The overall design reduces feature-processing cost while preserving complete spatial responses for coordinate representation. Its architecture is shown in Figure 5.

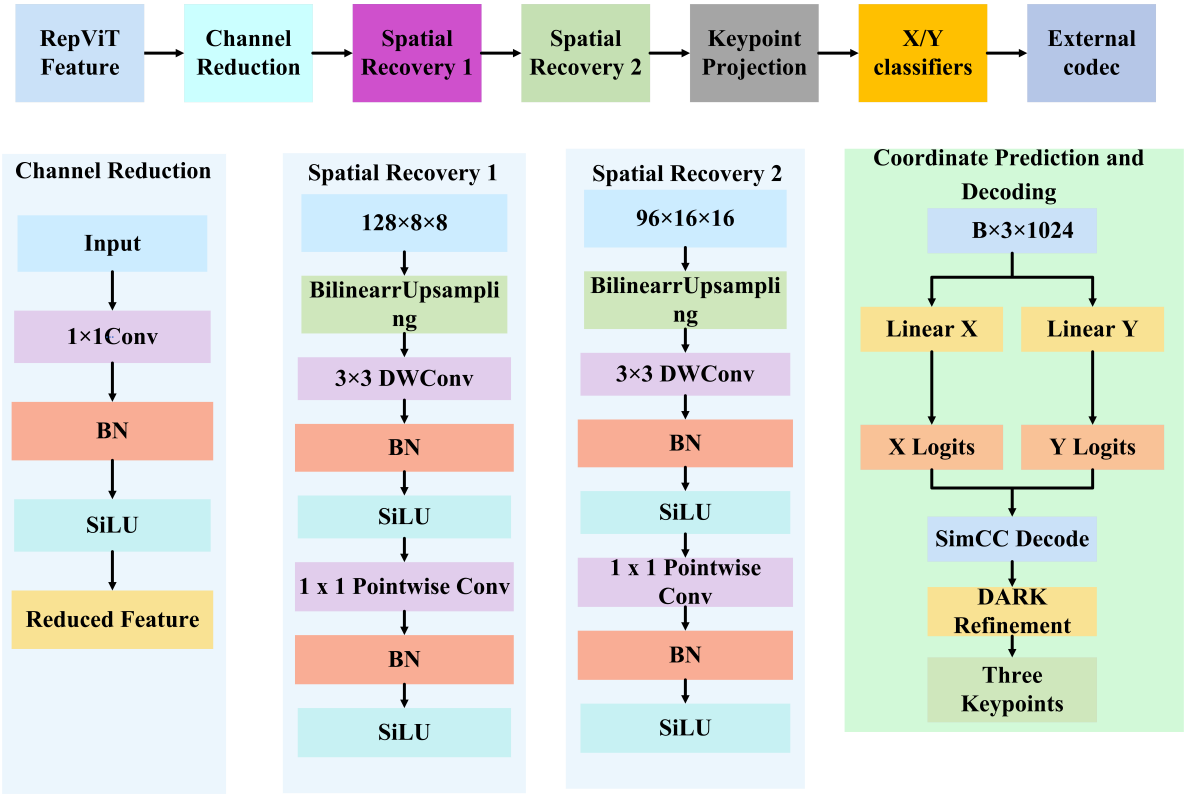


Figure 5. Architecture of LiteFlattenSimCCHead.

Let F_b denote the feature output by RepViT-M0.6. LiteFlattenSimCCHead first obtains a compact feature representation through channel compression and then uses two cascaded lightweight spatial-reconstruction modules to progressively increase spatial resolution. This process can be summarized as

$$F_h = U_2(U_1(\mathcal{R}(F_b))) \tag{4}$$

where R(.) denotes the channel-compression operation, U_1(.) and U_2(.) denote the successive spatial-reconstruction modules, and F_h is the reconstructed localization feature.

Channel compression is first performed at low spatial resolution. The purpose is not merely to reduce network width, but to reduce the scale of subsequent computation before high-resolution feature processing. As spatial dimensions increase, convolution on high-channel-count features rapidly becomes more expensive. A low-dimensional mapping of the backbone output therefore concentrates subsequent computation in a compact feature space more relevant to keypoint localization. This operation does not change the positional arrangement of the original features; instead, it recombines information across channels to form a base representation more suitable for subsequent spatial reconstruction.

$$F_i = P_i(D_i(\mathrm{Up}(F_{i-1}))), \quad i = 1, 2 \tag{5}$$

where $\mathrm{Up}(\cdot)$ denotes bilinear upsampling, $D_i(\cdot)$ denotes depthwise convolution, and $P_i(\cdot)$ denotes pointwise convolution.

Upsampling only increases the number of spatial sampling points and cannot regenerate information already lost during backbone downsampling. This process should therefore not be interpreted as a direct reconstruction of original-image details. Instead, it increases the spatial sampling density of the feature space to construct a position representation more suitable for keypoint-coordinate prediction. Depthwise convolution is introduced after each scale expansion so that neighboring spatial positions can exchange local information, preventing newly interpolated positions from being determined solely by fixed interpolation relationships. Pointwise convolution then integrates information across channels, further fusing local spatial responses with high-level structural information.

This design corresponds well to the feature distribution in line-laser images. Unlike the large textured regions found in natural images, the principal discriminative information in the images considered here is concentrated in the laser stripe and its neighborhood. Keypoints generally correspond to locations at which the stripe shape changes locally. The prediction stage must therefore emphasize subtle response differences among spatial positions rather than continue to expand high-level semantic representation. By progressively increasing spatial sampling density, structural information extracted by the backbone is mapped to a feature scale better suited to keypoint-position discrimination, providing a richer spatial basis for subsequent coordinate prediction.

To prevent spatial reconstruction from eroding the efficiency advantage of the lightweight backbone, local feature fusion combines depthwise and pointwise convolutions. Depthwise convolution primarily models spatial neighborhoods within individual channels, whereas pointwise convolution exchanges information across channels. Separating spatial mixing from channel mixing avoids the continuous use of expensive standard convolutions at higher spatial resolution. For an online robotic vision-guided adhesive-dispensing system, this design keeps prediction-stage computation low while retaining the required local structural modeling capability.

After spatial reconstruction, the shared localization feature is mapped through keypoint projection to feature responses corresponding to the three target keypoints. Let the resulting feature representation for the k-th keypoint be denoted by F_k; then

$$F_k = P_k(F_h) \tag{6}$$

where $P_k(\cdot)$ denotes the keypoint-projection operation. This mapping further transforms the shared feature into position responses corresponding to P_L, P_C, and P_R, providing a basis for separate coordinate discrimination of the three keypoints.

After the keypoint features are formed, global pooling is not used to aggregate the spatial features. Instead, all responses on the two-dimensional grid are retained and converted into one-dimensional position features through flattening:

$$Z_k = \text{Flatten}(F_k) \tag{7}$$

This design is the principal distinction between LiteFlattenSimCCHead and a conventional lightweight classification head. Global pooling effectively compresses features but also weakens absolute spatial-position information, whereas the task considered here is fundamentally one of position determination. In particular, the three keypoints have explicit relative geometric relationships, so spatial responses should not be excessively aggregated before coordinate prediction. Flattening retains the responses at different locations in the feature grid, allowing subsequent coordinate classification to continue using the complete spatial distribution.

The resulting position representation Z_k is subsequently fed into the SimCC coordinate classification module. The detailed coordinate classification process and the subsequent distribution-aware coordinate refinement are described in Section 4.4.

In summary, LiteFlattenSimCCHead is not designed around the isolated introduction of a particular convolution operator. Instead, it reorganizes the prediction stage to address the representational mismatch between the lightweight RepViT-M0.6 backbone and the precise line-laser localization task. Low-dimensional feature processing reduces the cost of spatial reconstruction; progressive upsampling and lightweight local convolution enhance positional representation; and keypoint projection and flattening preserve spatial responses associated with coordinate prediction, ultimately forming features suitable for SimCC axial classification. RepViT-M0.6 efficiently extracts hierarchical features of the line-laser contour, while LiteFlattenSimCCHead transforms low-resolution deep features into position-sensitive representations better suited to three-keypoint localization. Together, the two components balance computational efficiency and localization requirements in robotic vision-guided adhesive dispensing.

4.4. SimCC Coordinate Classification and DARK Coordinate Refinement

After processing by LiteFlattenSimCCHead, the network obtains position features for the three keypoints. To convert them into two-dimensional coordinates, SimCC is used to represent keypoint positions through axial classification, decomposing two-dimensional localization into two independent one-dimensional classification problems along the horizontal and vertical directions. For the kth keypoint, the prediction is expressed as

$$S_x^{(k)} = C_x(Z_k) \tag{8}$$

$$S_y^{(k)} = C_y(Z_k) \tag{9}$$

where Z_k denotes the position representation of the k-th keypoint obtained from LiteFlattenSimCCHead, while $C_x(\cdot)$ and $C_y(\cdot)$ denote the classification mappings along the horizontal and vertical directions, respectively. During training, ground-truth keypoint coordinates are encoded as corresponding one-dimensional target distributions, and distribution supervision is used to learn positional responses along the two axes. This approach provides a fine-grained position representation for keypoints on a line-laser stripe while avoiding the cost of high-resolution two-dimensional heatmaps.

During inference, the initial keypoint positions are first obtained from the peaks of the predicted coordinate distributions along the horizontal and vertical directions. Because coordinate classification is defined in a discrete space, direct peak decoding may introduce discretization bias. Therefore, the distribution-aware decoding strategy of DARK is applied to refine each coordinate using the local distribution around the predicted peak. For one coordinate direction, the local correction is expressed as

$$u = u_0 - \frac{f'(u_0)}{f''(u_0)} \tag{10}$$

where u_0 is the initial peak position and $f(\cdot)$ denotes the logarithmic response in the neighborhood of the peak. Correcting the horizontal and vertical positions separately yields the final two-dimensional keypoint coordinates. In the implementation, the initial peak positions are obtained by applying the SimCC maximum-response operation independently to the horizontal distribution `simcc_x` and the vertical distribution `simcc_y`. When DARK is enabled, the decoder calls `refine_simcc_dark` separately for the two one-dimensional distributions. The refinement-window sizes are calculated from the corresponding SimCC standard deviations and adjusted to odd integer values before refinement. After the horizontal and vertical refinements, the resulting coordinates are divided by the SimCC split ratio to convert them from SimCC coordinate space to the input-image coordinate space. For the final model, DARK refinement is enabled in the SimCC decoder. The corresponding codec uses an input size of 256×256, a split ratio of 2.0, and target standard deviations $\sigma = (5.66, 5.66)$. DARK is applied only during inference and does not introduce additional trainable layers, trainable parameters, or a separate training stage. The network ultimately outputs the left point P_L, center point P_C, and right point P_R, completing the prediction from LiteFlattenSimCCHead position features to three-keypoint coordinates. SimCC establishes the axial coordinate distributions, while DARK improves the precision of discrete-distribution decoding; together, they implement coordinate representation and decoding at the network output.

5. Experiments and Results

5.1. Experimental Environment

To ensure a fair and consistent comparison, all experiments were conducted on 64-bit Windows 11 Professional using Python 3.8.20, PyTorch 2.4.1, and CUDA 11.8. Training and testing were performed with an NVIDIA GeForce RTX 4070 GPU and an Intel(R) Core(TM) i5-12400 CPU. Other parameter settings are listed in Table 2.

Table 2. Experimental parameter settings.

Experimental Parameter	Value
Image size	256×256
Batch size	32
Initial learning rate	5×10^{-4}
Optimizer	Adam
Warm-up iterations	500
Maximum epochs	210
SimCC split ratio	2.0
Learning-rate decay	MultiStepLR; milestones at epochs 170 and 200, $\gamma = 0.1$
Loss function for SimCC models	`KLDiscretLoss`
Target weighting for SimCC models	Enabled
Loss parameter for SimCC models	$\beta = 10.0$
Label softmax for SimCC models	Enabled
SimCC target distribution	$\sigma = (5.66, 5.66)$; target normalization disabled
RepViT-M0.6 initialization	`repvit_m0_6_distill_300e.pth`
Translation augmentation	Shift factor 0.06
Rotation augmentation	$\pm 10^\circ$
Scale augmentation	$(0.90, 1.10)$
Model selection	Validation-set metrics; test set excluded

The loss and target-distribution settings listed as "for SimCC models" apply to the RepViT-M0.6 configurations using the SimCC representation. The Lite-HRNet-18 baseline and other heatmap-based comparison models use their corresponding heatmap-based training configurations. The RepViT-M0.6 backbone was initialized with `repvit_m0_6_distill_300e.pth`; this checkpoint was not used as a universal initialization for all comparison models.

The training configuration used Adam with an initial learning rate of 5×10^{-4}. A linear warm-up was applied during the first 500 iterations, with a start factor of 0.001. The learning rate was reduced by a factor of 0.1 at epochs 170 and 200. The maximum training duration was 210 epochs with a batch size of 32. The training augmentation used a translation shift factor of 0.06, a rotation range of $\pm 10^{\circ}$, and a scale range of $(0.90, 1.10)$.

Model selection was performed using the validation subset, and the test subset was not used for checkpoint selection. The configuration saved best checkpoints according to the validation metrics specified for each experiment, including AP, mean pixel error, Arm error, and PCK@5.

The first-stage YOLO26n detector was trained with random seed 42 and deterministic execution enabled. The second-stage MMPose keypoint configurations did not explicitly define a fixed training seed, and the reported comparison and ablation results were obtained from one saved checkpoint for each configuration. Therefore, the table values are single-run results rather than averages over repeated training runs, and no standard deviation across independent training runs is reported.

5.2. Inference Timing Protocol

Inference speed was evaluated using the 567 test instances with a batch size of one. Each timing experiment was repeated three times, resulting in 1701 timed inferences. CUDA synchronization was performed immediately before and after each timed inference to reduce the influence of asynchronous GPU execution. In the model-mode benchmark, image loading, ROI construction, affine transformation, normalization, and device transfer were completed before the timed section. The timed section therefore measured the model inference operation used for keypoint prediction. The mean latency was calculated by averaging the per-instance latency over all timed inferences, and the derived FPS was calculated as $1000/\bar{t}$, where $\bar{t}$ denotes the mean latency in milliseconds. The warm-up count was not retained in the saved benchmark metadata and is therefore not specified in this manuscript.

The reported GPU and CPU speeds correspond to the second-stage single-ROI keypoint-model inference under the specified timing protocol. The timed operation excludes image loading, first-stage YOLO26n detection, confidence filtering, ROI construction, affine transformation, normalization, and device transfer. Therefore, the reported 116.34 GPU FPS and 49.32 CPU FPS characterize the efficiency of the keypoint-recognition stage only and should not be interpreted as the end-to-end throughput of the complete two-stage vision pipeline.

5.3. Evaluation Metrics

To comprehensively evaluate localization accuracy, geometric-structure preservation, and inference efficiency for three-keypoint recognition in line-laser adhesive dispensing, Average Precision (AP), mean pixel error (Mean), root mean square pixel error (RMSE), center-keypoint error (Center), contour-slope error (Slope), side-arm angle error (Arm), Percentage of Correct Keypoints at 5 pixels (PCK@5), and inference speed (FPS) are used. AP is calculated using the COCO-style keypoint evaluation implemented by MMPose, following the OKS-based keypoint evaluation procedure. The dataset metainfo defines three key-

points in the order center, left, and right, with equal OKS sigma values of $(0.05, 0.05, 0.05)$. The OKS scale is determined from the annotated instance area according to the COCO-style evaluation procedure. The AP is averaged over the default OKS thresholds from 0.50 to 0.95 with an interval of 0.05. The custom pixel and geometric metrics are calculated separately from the matched prediction–ground-truth pairs and are not used to calculate AP. RMSE reflects sensitivity to relatively large localization errors.

The SimCC configuration uses an input size of 256 × 256 pixels, a split ratio of 2.0, and target-distribution standard deviations of $(5.66, 5.66)$ for the horizontal and vertical coordinate distributions. These SimCC target-distribution parameters are distinct from the OKS sigma values used by the AP evaluator. DARK is enabled only for the final decoding configuration.

The detector-generated ROI is transformed to the fixed 256 × 256 network input using the MMPose top-down affine transformation. The keypoint model predicts coordinates in the transformed ROI coordinate system. These coordinates are then mapped back to the original image coordinate system using the inverse transformation before the custom localization metrics are calculated. Therefore, the reported pixel errors are measured in original-image coordinates rather than directly in the resized network-input coordinates.

The principal metrics are calculated as follows.

Assume that N valid instances are obtained after one-to-one matching between detected and ground-truth boxes, with each instance containing the left keypoint P_L, center keypoint P_C, and right keypoint P_R. For the kth keypoint of the ith instance, let $\mathbf{p}_{i,k}$ and $\hat{\mathbf{p}}_{i,k}$ denote the ground-truth and predicted coordinates, respectively. The pixel localization error of one keypoint is

$$e_{i,k} = \left\| \hat{\mathbf{p}}_{i,k} - \mathbf{p}_{i,k} \right\|_2 = \sqrt{(\hat{x}_{i,k} - x_{i,k})^2 + (\hat{y}_{i,k} - y_{i,k})^2} \tag{11}$$

The mean pixel error over all three keypoints is defined as

$$E_{\text{Mean}} = \frac{1}{3N} \sum_{i=1}^{N} (e_{i,L} + e_{i,C} + e_{i,R}) \tag{12}$$

The root mean square pixel error is defined as

$$E_{\text{RMSE}} = \sqrt{\frac{1}{3N} \sum_{i=1}^{N} \sum_{k \in \{L,C,R\}} e_{i,k}^2}, \tag{13}$$

where $e_{i,k}$ denotes the Euclidean localization error of the k-th keypoint in the i-th successfully matched instance.

The center-keypoint error is defined as

$$E_{\text{Center}} = \frac{1}{N} \sum_{i=1}^{N} e_{i,C} \tag{14}$$

To quantify the local contour orientation preserved by the predicted keypoints, the ground-truth and predicted contour angles are calculated from the left and right keypoints, respectively:

$$\theta_i = \text{atan2}(y_{i,R} - y_{i,L}, x_{i,R} - x_{i,L}), \quad \hat{\theta}_i = \text{atan2}(\hat{y}_{i,R} - \hat{y}_{i,L}, \hat{x}_{i,R} - \hat{x}_{i,L}) \tag{15}$$

Considering angular periodicity, Slope is defined as

$$E_{\text{Slope}} = \frac{1}{N}\sum_{i=1}^{N}\left|\left((\hat{\theta}_i - \theta_i + 180^\circ) \bmod 360^\circ\right) - 180^\circ\right| \tag{16}$$

To quantify the angular consistency of the two side arms formed by the left, center, and right keypoints, the directions from the left and right keypoints toward the center point are calculated separately. For the i-th target instance, the ground-truth left- and right-arm angles are defined as

$$\alpha_{i,L} = \text{atan2}(y_{i,C} - y_{i,L}, x_{i,C} - x_{i,L}), \qquad \alpha_{i,R} = \text{atan2}(y_{i,C} - y_{i,R}, x_{i,C} - x_{i,R}). \tag{17}$$

The corresponding predicted arm angles are

$$\hat{\alpha}_{i,L} = \text{atan2}(\hat{y}_{i,C} - \hat{y}_{i,L}, \hat{x}_{i,C} - \hat{x}_{i,L}), \qquad \hat{\alpha}_{i,R} = \text{atan2}(\hat{y}_{i,C} - \hat{y}_{i,R}, \hat{x}_{i,C} - \hat{x}_{i,R}). \tag{18}$$

To account for the periodicity of angular coordinates, the angular difference is defined as

$$d_\theta(a, b) = |(a - b + 180^\circ) \bmod 360^\circ - 180^\circ|. \tag{19}$$

The Arm error is then defined as the mean angular error of the two side arms:

$$E_{\text{Arm}} = \frac{1}{2N}\sum_{i=1}^{N}[d_\theta(\hat{\alpha}_{i,L}, \alpha_{i,L}) + d_\theta(\hat{\alpha}_{i,R}, \alpha_{i,R})]. \tag{20}$$

Only target instances for which the left, center, and right keypoints are all valid are included in the Arm calculation. The Arm error is reported in degrees.

PCK@5 represents the proportion of keypoints whose localization error does not exceed 5 pixels. In the implementation, the PCK threshold is explicitly set to 5.0 pixels. Although YOLO26n-generated ROIs are resized to 256 × 256 pixels before being processed by the second-stage keypoint network, MMPose maps the predicted keypoints back to the original-image coordinate system before the localization metrics are calculated. Therefore, the 5-pixel threshold is applied in the original-image coordinate system, rather than directly in the resized ROI coordinate system. The 256 × 256 ROI is used to standardize the network input, while the same 5-pixel threshold is used consistently for all compared models. The same threshold was applied to all methods and does not represent a calibrated physical dispensing tolerance.

$$\text{PCK@5} = \frac{1}{3N}\sum_{i=1}^{N}\sum_{k\in\{L,C,R\}}\mathbb{I}(e_{i,k} \le 5) \tag{21}$$

The experiments in this study evaluate two-dimensional keypoint localization within a single detected ROI. The reported pixel errors, geometric errors, and PCK@5 are calculated after the predicted keypoints are mapped from the resized ROI back to the original-image coordinate system. Thus, 256 × 256 defines the input resolution of the second-stage keypoint network, whereas the original image defines the coordinate system used for the reported pixel-based metrics. Therefore, these results measure the performance of the keypoint localization stage and should not be interpreted as complete-system robotic dispensing accuracy or physical robot-positioning accuracy.

5.4. Ablation Study

To examine the contributions of the prediction head and coordinate decoding, the ablation study is organized as a sequence of controlled comparisons. The Baseline uses Lite-HRNet-18 with HeatmapHead and MSRAHeatmap encoding and decoding. Model A uses RepViT-M0.6 with the standard SimCCHead and standard SimCC decoding. Model

B retains the RepViT-M0.6 backbone and replaces only the standard SimCCHead with LiteFlattenSimCCHead, while keeping the input size, SimCC coordinate representation, loss configuration, and standard decoding unchanged. Model C uses the same trained network weights as Model B and enables DARK only during inference-time decoding. Thus, the Model A–Model B comparison evaluates the head replacement, and the Model B–Model C comparison evaluates DARK. The Baseline–Model A comparison changes the backbone, prediction head, and coordinate representation/decoding scheme simultaneously; it is therefore a comparison between complete configurations and cannot isolate the contribution of RepViT-M0.6 alone. The results are presented in Table 3.

Compared with the Lite-HRNet-18 baseline, Model C (Ours) increases AP from 0.9152 to 0.9532, an improvement of 3.80 percentage points, and increases PCK@5 from 0.7337 to 0.7842, an improvement of 5.05 percentage points. Mean, RMSE, Center, Slope, and Arm errors decrease by 16.51%, 20.12%, 20.14%, 24.08%, and 14.91%, respectively. The second-stage single-ROI inference speed increases from 23.78 to 116.34 FPS on the GPU and from 16.10 to 49.32 FPS on the CPU. These values compare the complete baseline and proposed configurations; because the backbone, head, coordinate representation, and decoding scheme differ simultaneously, the improvements cannot be attributed solely to RepViT-M0.6.

The controlled Model A–Model B comparison evaluates the replacement of Standard SimCCHead with LiteFlattenSimCCHead while retaining the RepViT-M0.6 backbone and standard SimCC decoding. AP increases from 0.8521 to 0.9461, Mean decreases from 5.2306 to 3.8415 px, RMSE decreases from 7.0058 to 5.0469 px, and PCK@5 increases from 0.6537 to 0.7678. Center, Slope, and Arm errors also decrease from 5.1999 to 4.1764 px, from 3.8549 to 2.2790 degrees, and from 4.6229 to 3.9554 degrees, respectively. This comparison supports the effectiveness of LiteFlattenSimCCHead under the evaluated RepViT-M0.6 configuration.

The Model B–Model C comparison isolates the decoding change: Model C retains the trained Model B network weights and enables DARK only during inference. AP increases from 0.9461 to 0.9532, Mean decreases from 3.8415 to 3.6385 px, RMSE decreases from 5.0469 to 4.8272 px, and PCK@5 increases from 0.7678 to 0.7842. Center, Slope, and Arm errors decrease from 4.1764 to 3.9802 px, from 2.2790 to 2.1432 degrees, and from 3.9554 to 3.5906 degrees, respectively. Under the unified timing protocol, derived GPU FPS decreases from 120.49 to 116.34, while CPU FPS decreases from 49.40 to 49.32. These results indicate improved localization with a measurable decoding overhead. Overall, the A–B comparison indicates that LiteFlattenSimCCHead provides the primary contribution to the accuracy improvement, whereas DARK provides a smaller but consistent additional gain through inference-time coordinate refinement.

Effectiveness and Efficiency of LiteFlattenSimCCHead

To verify the lightweight characteristics of LiteFlattenSimCCHead, the RepViT-M0.6 backbone, input size, SimCC coordinate encoding, loss function, and standard decoding method are held fixed, and only Standard SimCCHead is replaced with LiteFlattenSimCCHead. Under identical hardware, software, and timing protocols, the complete models are compared in terms of parameter count, theoretical computation, and GPU and CPU inference efficiency.

Table 3. Evolution of the main model architecture and comparison of overall performance.

Model	Backbone	Keypoint Head	Decoder	AP	Mean/px	RMSE/px	Center/px	Slope/deg	Arm/deg	PCK@5	GPU FPS	CPU FPS
Baseline	Lite-HRNet-18	HeatmapHead	MSRAHeatmap	0.9152	4.3580	6.0430	4.9840	2.8230	4.2200	0.7337	23.78	16.10
Model A	RepViT-M0.6	Standard SimCCHead	Standard SimCC	0.8521	5.2306	7.0058	5.1999	3.8549	4.6229	0.6537	93.57	29.90
Model B	RepViT-M0.6	LiteFlattenSimCCHead	Standard SimCC	0.9461	3.8415	5.0469	4.1764	2.2790	3.9554	0.7678	120.49	49.40
Model C (Ours)	RepViT-M0.6	LiteFlattenSimCCHead	DARK	0.9532	3.6385	4.8272	3.9802	2.1432	3.5906	0.7842	116.34	49.32

Structurally, the Standard SimCCHead receives the $320 \times 8 \times 8$ RepViT feature map and applies three transposed-convolution layers, each with 256 output channels, to progressively increase the spatial resolution to 64×64. A final 1×1 convolution produces three keypoint feature maps. Each map is flattened and passed to separate fully connected projections for the horizontal and vertical SimCC distributions. In contrast, LiteFlattenSimCCHead first reduces the channel dimension from 320 to 128 using a 1×1 convolution. It then performs two $2\times$ bilinear upsampling stages, each followed by a depthwise-separable convolution block, reducing the channels from 128 to 96 and then to 64. A 1×1 keypoint projection produces three 32×32 maps, which are flattened and projected into the horizontal and vertical SimCC distributions. Thus, the head replacement reduces both the channel width of the spatial-processing stages and the spatial size of the flattened representation while retaining separate one-dimensional SimCC coordinate prediction.

To complement the quantitative ablation results, the final coordinate responses of Standard SimCCHead and LiteFlattenSimCCHead are visualized in Figure 6. Since SimCC predicts independent one-dimensional response distributions along the horizontal and vertical axes rather than a conventional two-dimensional heatmap, the response map of the k-th keypoint is reconstructed only for visualization as follows.

$$H_k(x, y) = \widetilde{p}_k^x(x)\, \widetilde{p}_k^y(y), \tag{22}$$

Here, $\widetilde{p}_k^x(x)$ and $\widetilde{p}_k^y(y)$ denote the independently normalized horizontal and vertical SimCC responses, respectively. The reconstructed response maps are used only for qualitative analysis and do not participate in model training or coordinate decoding.

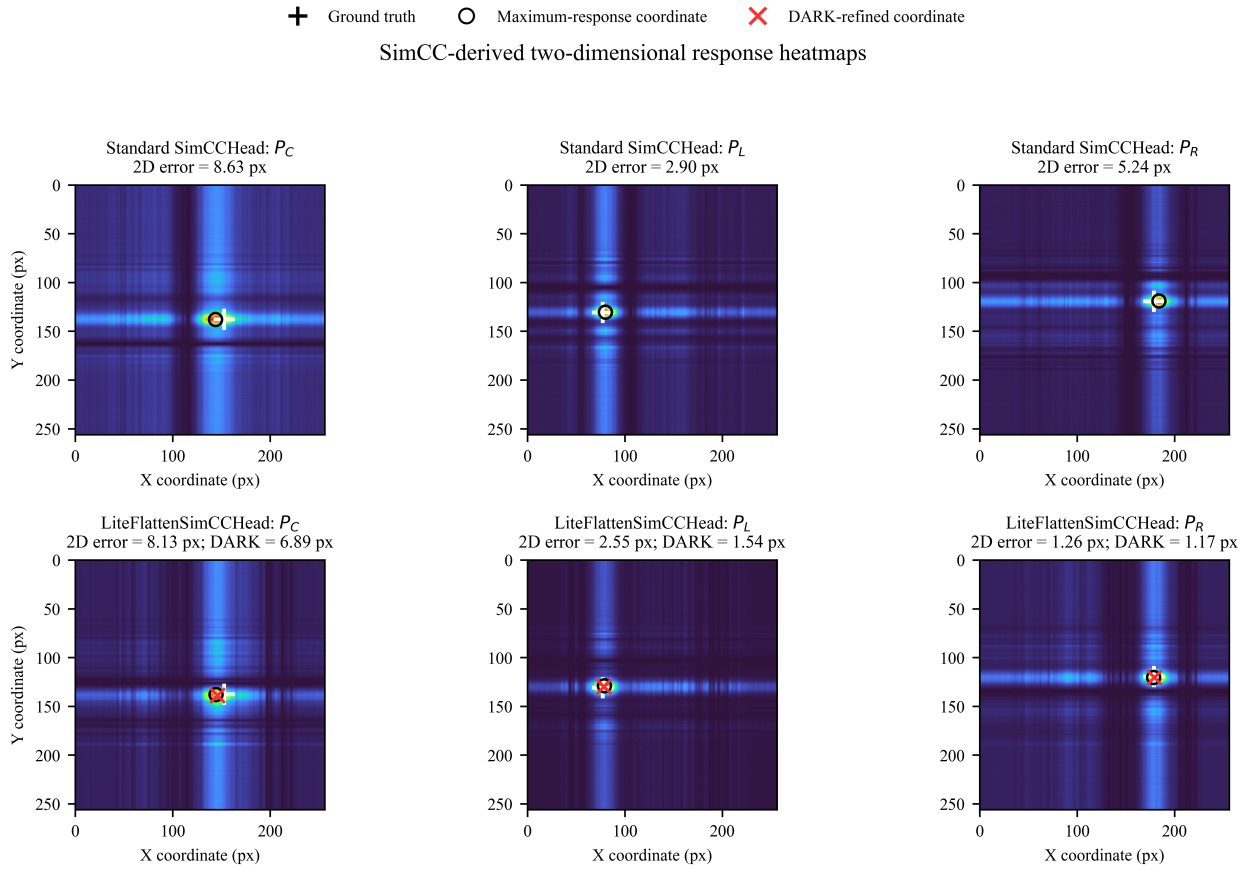


Figure 6. SimCC-derived response heatmaps for Standard SimCCHead and LiteFlattenSimCCHead. The columns represent P_C, P_L, and P_R. The plus sign and circle denote the ground truth and maximum-response coordinate, respectively; the cross in the lower row denotes the DARK-refined coordinate.

The principal visual difference lies in the displacement between the maximum-response coordinate and the ground-truth position rather than solely in the shape or color intensity of the reconstructed response map. For the illustrated test instance, replacing Standard SimCCHead with LiteFlattenSimCCHead reduces the two-dimensional ROI-space errors of P_C, P_L, and P_R from 8.63, 2.90, and 5.24 px to 8.13, 2.55, and 1.26 px, respectively. DARK refinement further reduces the corresponding errors to 6.89, 1.54, and 1.17 px. The most evident improvement appears for P_R, whose maximum-response coordinate is substantially closer to the ground-truth position.

The efficiency results are shown in Table 4.

Table 4. Efficiency ablation of LiteFlattenSimCCHead.

Structure	GPU FPS	GPU Latency/ms	CPU FPS	CPU Latency/ms	Params/M	GFLOPs
Standard SimCCHead	93.57	10.687	29.90	33.446	9.771	1.959
LiteFlattenSimCCHead	120.49	8.300	49.40	20.245	3.278	0.532

At an input size of 256 × 256, the complete model with Standard SimCCHead has 9.771 M parameters and requires 1.959 GFLOPs. Replacing it with LiteFlattenSimCCHead reduces the parameter count to 3.278 M, a decrease of 66.45%, and the theoretical computation to 0.532 GFLOPs, a decrease of 72.84%. These results show that LiteFlattenSimCCHead substantially compresses the model and reduces the theoretical computational cost of keypoint prediction.

Under the unified timing protocol, the Derived FPS on the GPU increases from 93.57 to 120.49 FPS, while the corresponding mean latency decreases from 10.687 to 8.300 ms, representing an FPS improvement of 28.77% and a latency reduction of 22.34%. On the CPU, the Derived FPS increases from 29.90 to 49.40 FPS, while mean latency decreases from 33.446 to 20.245 ms, corresponding to an FPS improvement of 65.18% and a latency reduction of 39.47%. These results show that LiteFlattenSimCCHead reduces the inference latency of the complete second-stage keypoint model on both platforms, with a particularly clear benefit on the CPU platform.

Overall, with the RepViT-M0.6 backbone and SimCC prediction pipeline unchanged, LiteFlattenSimCCHead substantially reduces the parameter count and theoretical computation of the complete model and improves second-stage keypoint inference efficiency on both GPU and CPU platforms. Together with the localization-accuracy results, the head provides a more favorable balance between model accuracy and computational efficiency and establishes a lightweight implementation basis for online three-keypoint recognition in vision-guided adhesive dispensing.

5.5. Comparison Experiments

To verify the effectiveness of the proposed method for keypoint localization in line-laser adhesive dispensing, it is compared with RLE-ResNet50 [17], Integral-ResNet50 [18], Hourglass-52 [19], SimpleBaseline-ResNet50 [20], HRNet-W32 [21], and Lite-HRNet-18 [13]. The evaluation considers keypoint-localization accuracy, local geometric errors, and inference efficiency. The results are shown in Table 5.

As shown in Table 5, the proposed method achieves an AP of 0.9532, which is 3.80 percentage points higher than that of Lite-HRNet-18. Its Mean and RMSE are 3.6385 and 4.8272 px, respectively, representing reductions of 16.5% and 20.1% relative to Lite-HRNet-18, while PCK@5 increases from 0.7337 to 0.7842. The Center, Slope, and Arm errors are 3.9802 px, 2.1432°, and 3.5906°, respectively. For second-stage single-ROI inference, GPU and CPU speeds reach 116.34 and 49.32 FPS. Relative to RLE-ResNet50, the fastest compari-

son model on the GPU at 89.52 FPS, the GPU speed improves by approximately 30.0%; the CPU speed is approximately 3.06 times that of Lite-HRNet-18 (16.10 FPS).

Table 5. Comparison of different keypoint-recognition methods.

Method	AP	Mean/px	RMSE/px	Center/px	Slope/deg	Arm/deg	PCK@5	GPU FPS	CPU FPS
Lite-HRNet-18	0.9152	4.3580	6.0430	4.9840	2.8230	4.2200	0.7337	23.78	16.10
RLE-ResNet50	0.8081	7.0139	8.0188	5.1389	4.4716	5.5234	0.3451	89.52	9.57
Integral-ResNet50	0.8973	4.5244	6.1750	4.8587	2.8029	4.2647	0.7113	78.84	7.83
Hourglass-52	0.8762	4.8040	6.7367	5.1158	2.9707	4.2351	0.6931	80.68	3.90
SimpleBaseline-ResNet50	0.8363	5.1139	7.3388	4.9064	3.9813	4.0516	0.6872	77.78	7.72
HRNet-W32	0.8501	5.0893	7.0712	4.7175	3.5691	4.0330	0.6743	20.08	6.93
Ours	0.9532	3.6385	4.8272	3.9802	2.1432	3.5906	0.7842	116.34	49.32

5.6. Visual Comparison and Error Analysis

To further compare localization performance among models for three-keypoint recognition in line-laser adhesive dispensing, test images with different contour shapes, brightness distributions, and noise interference are selected for qualitative analysis, as shown in Figure 7. White rings indicate ground-truth keypoints; green, red, and blue markers indicate the predicted left, center, and right keypoints, respectively; and yellow line segments represent the local geometric contour formed by the three predicted keypoints.

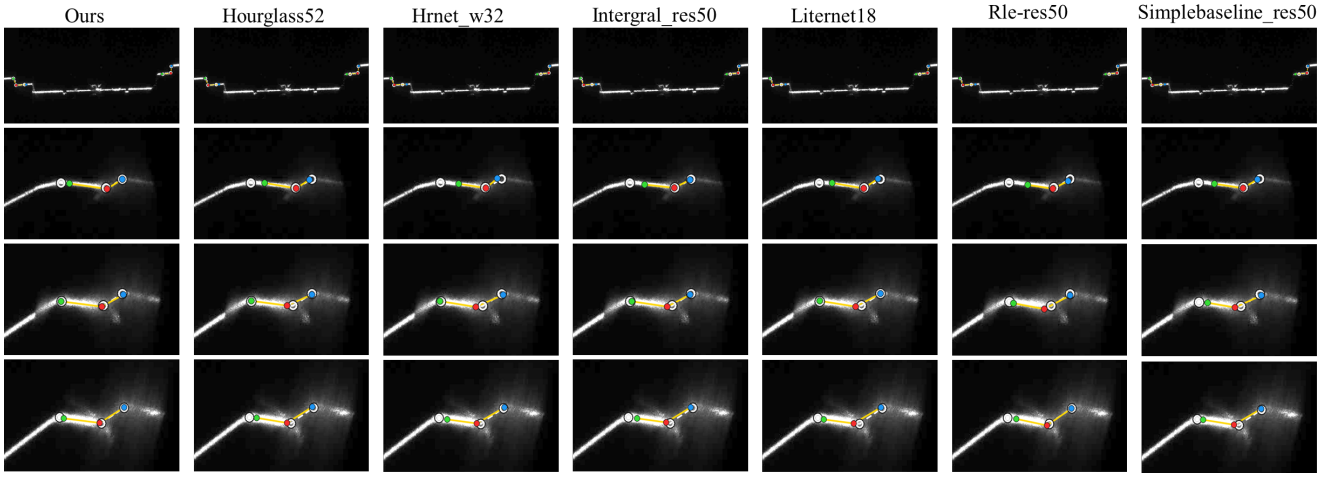


Figure 7. Visual comparison of three-keypoint localization by different models.

The visual results show that all models can locate the approximate keypoint positions when the laser stripe is continuous and its edges are clear. However, under local specular reflection, stripe broadening, nonuniform brightness, or weakened contour-bend features, several comparison models exhibit localization offsets of varying magnitude. The center keypoint is susceptible to interference from bright regions and can deviate from the true bend location. The left and right keypoints may also drift along the laser-stripe direction, causing deviations between the predicted and true contour slopes and side-arm angles.

The three keypoints predicted by the proposed method overlap more closely with the ground-truth annotations. In particular, under blurred stripe boundaries and local grayscale diffusion, the improved network still determines the center bend position accurately and maintains a stable spatial order among the left, center, and right keypoints. The predicted connecting lines are more consistent with changes in the actual laser contour, indicating that the model preserves the geometric relationships among the three keypoints while localizing each individual point.

The proposed method produces more stable keypoint-localization results across different laser-stripe shapes and imaging interference conditions. These qualitative observations are consistent with the quantitative trends in AP, Mean, Center, Slope, Arm, and PCK@5 over the complete test set, further demonstrating that the proposed model reduces keypoint drift under complex imaging conditions and improves the accuracy and geometric consistency of line-laser keypoint recognition for adhesive dispensing.

5.7. Analysis of PCK Curves at Different Error Thresholds

To further evaluate keypoint-recognition performance under different localization-error tolerances, PCK (Percentage of Correct Keypoints) curves are computed for the three keypoints in the test set over a range of error thresholds. Using the definition given in Equation (21), the PCK value is evaluated by varying the localization-error threshold τ.

As shown in Figure 8, the PCK of every model generally increases as the error threshold rises from 0, but clear differences appear within the strict small-error range. The proposed method maintains a higher curve over approximately 3–10 px. At a threshold of 5 px, the proposed method achieves a PCK@5 of 0.7842, meaning that 78.42% of the target keypoints have localization errors of no greater than 5 pixels.

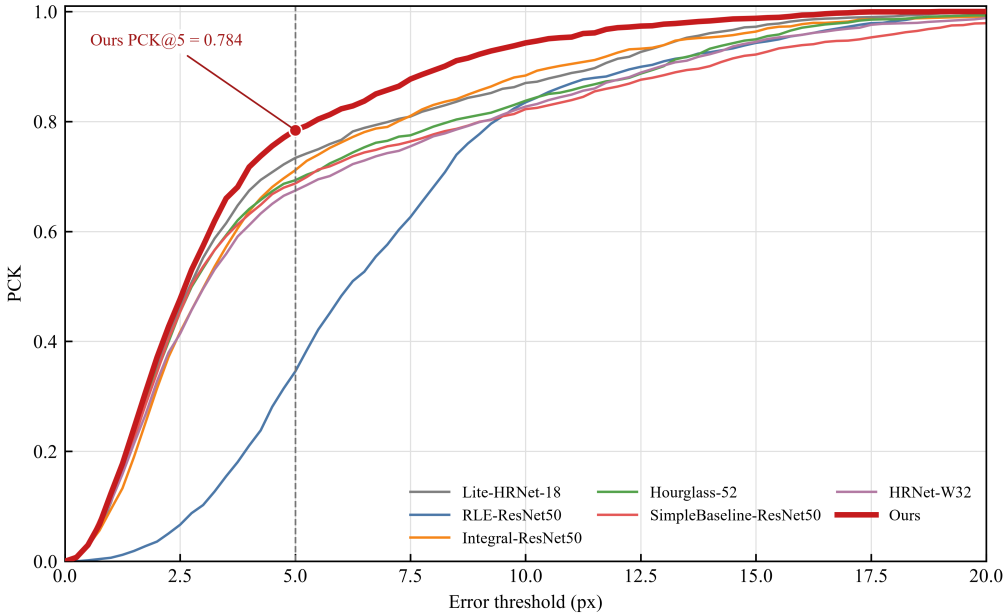


Figure 8. PCK curves at different error thresholds.

The overall curves show that the proposed method not only achieves a high proportion of correctly localized keypoints at a single threshold but also maintains favorable localization performance over a broad range of error thresholds. As the threshold increases further, the curves gradually approach 1 and the differences among models diminish. This indicates that the principal advantage of the proposed method lies in controlling small localization errors and producing high-precision keypoint predictions. This characteristic is consistent with the local-position accuracy required by robotic line-laser adhesive dispensing and can provide more stable visual input for subsequent calculation of local position and orientation.

5.8. Pixel-Error Distributions for Different Keypoints

To further analyze localization stability for different keypoints, the pixel localization errors of the left point P_L, center point P_C, and right point P_R are calculated separately. The error of the kth keypoint in the ith target is defined as

$$e_{i,k} = \sqrt{(\hat{x}_{i,k} - x_{i,k})^2 + (\hat{y}_{i,k} - y_{i,k})^2} \tag{23}$$

where $(\hat{x}_{i,k}, \hat{y}_{i,k})$ and $(x_{i,k}, y_{i,k})$ denote the predicted and ground-truth coordinates, respectively. Figure 9 shows the error distributions of the three keypoints for different models. In each box plot, the box denotes the interquartile range, the center line denotes the median, and the diamond denotes the mean.

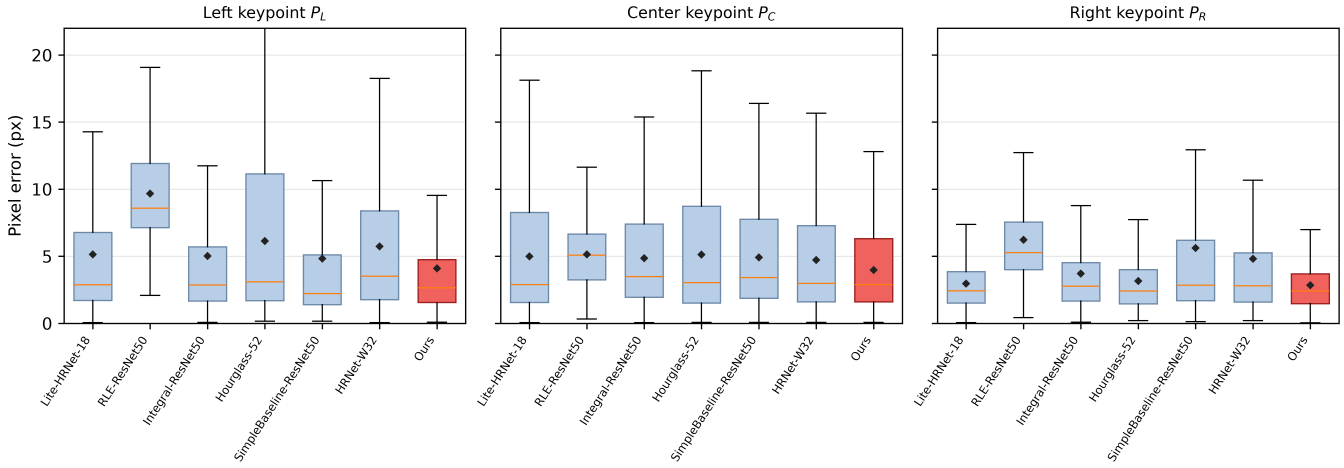


Figure 9. Error distributions of the three keypoints for different models.

For the proposed method, the mean errors of the left point P_L, center point P_C, and right point P_R are 4.0928, 3.9802, and 2.8425 px, respectively, and the corresponding median errors are 2.6440, 2.8868, and 2.4036 px. The mean error of the right point is approximately 30.5% lower than that of the left point. The left-point error distribution is relatively wide, with a 95th-percentile error of 13.6814 px; the 95th-percentile errors of the center and right points are 9.9713 and 6.7469 px, respectively. These results indicate that localization difficulty differs among the three keypoints.

Compared with the other models, the proposed method has box plots in lower error ranges for all three keypoints, with particularly concentrated error distributions for the center and right points. This finding indicates that the proposed method reduces overall mean localization error and improves the stability of predictions for different keypoints. Together with the Mean, RMSE, and PCK results, it shows that the performance improvement is not attributable to only a small number of samples; rather, favorable localization is achieved across multiple keypoints and different error ranges.

6. Conclusions

This study investigated a three-keypoint recognition method that balances localization accuracy and computational efficiency for vision-guided adhesive dispensing using a line-laser sensor mounted on a robotic arm. To address the narrow structure of the laser stripe and the low spatial resolution of deep features from a lightweight network, RepViT-M0.6 was used as the lightweight feature extraction backbone and LiteFlattenSimCCHead was designed. Through progressive spatial reconstruction and lightweight local feature fusion, the proposed head enhances the ability of low-resolution deep features to represent changes

in keypoint position. SimCC and DARK are then combined to predict the coordinates of the three keypoints.

Compared with the Lite-HRNet-18 baseline, Model A achieves higher inference speed but lower localization performance; however, because the backbone, prediction head, and coordinate representation/decoding scheme change simultaneously, this comparison does not isolate the effect of RepViT-M0.6.Introducing LiteFlattenSimCCHead increases AP from 0.8521 to 0.9461, reduces Mean Error from 5.2306 to 3.8415 px, and increases PCK@5 from 0.6537 to 0.7678, demonstrating that the proposed prediction head improves the compatibility between a lightweight backbone and precise keypoint localization. After further coordinate refinement using DARK, the final model achieves an AP of 0.9532, while Mean Error and RMSE decrease to 3.6385 and 4.8272 px, respectively, and Slope Error and Arm Error are 2.1432 and 3.5906°, respectively.

Compared with keypoint-recognition methods including RLE, Integral Regression, Hourglass, SimpleBaseline, HRNet, and Lite-HRNet, the proposed method achieves superior results for AP, Mean, RMSE, Center, Slope, Arm, and PCK@5. Relative to Lite-HRNet-18, AP improves by 3.80 percentage points, Mean Error and RMSE decrease by 16.5% and 20.1%, respectively, and PCK@5 improves by 5.05 percentage points. PCK curves over different error thresholds and the error distributions of the three keypoints further show that the method's advantages are concentrated in keypoint localization under strict error tolerances and in the stability of predictions across different keypoints.

In summary, the proposed method achieves a favorable balance between model lightweighting and precise localization of line-laser keypoints and can provide visual information for subsequent calculation of the target dispensing region's position, local orientation, and robot motion parameters. Therefore, the current results demonstrate the performance of the vision and keypoint-recognition method, but do not yet establish the positioning accuracy of the complete robotic dispensing system.

7. Future Work

The present study primarily validates three-keypoint recognition in two-dimensional image space within the second-stage single-ROI setting. The reported results therefore do not directly represent physical robot positioning accuracy or the performance of the complete online dispensing system. Because camera calibration and image-to-robot coordinate transformation have not yet been completed, the reported pixel errors cannot currently be converted reliably into physical errors in millimeters. Future work will combine camera calibration and robot coordinate transformation to validate actual adhesive-dispensing trajectories and physical placement accuracy. Model robustness under complex imaging conditions, including specular reflection, weak laser intensity, local occlusion, broadened laser stripes, and rounded geometric transitions, will also be investigated. In addition, the complete two-stage vision pipeline comprising object detection and keypoint localization will be further optimized and evaluated to improve its applicability to practical online robotic adhesive dispensing.

Author Contributions: Conceptualization, C.C. (Chuanyi Cui) and F.D.; methodology, C.C. (Chuanyi Cui); software, C.C. (Chuanyi Cui) and C.D.; validation, C.C. (Chuanyi Cui), C.D. and W.L.; formal analysis, C.C. (Chuanyi Cui) and C.D.; investigation, C.C. (Chuanyi Cui) and W.L.; resources, C.C. (Chao Chen) and F.D.; data curation, C.C. (Chuanyi Cui) and W.L.; writing—original draft preparation, C.C. (Chuanyi Cui); writing—review and editing, C.C. (Chao Chen), F.D. and C.D.; visualization, C.C. (Chuanyi Cui); supervision, C.C. (Chao Chen) and F.D.; project administration, C.C. (Chao Chen); funding acquisition, C.C. (Chao Chen) and F.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the collaborative research project "Research on Automatic Tracking and Coating Technology for High-Viscosity Sealants" (Project No. 31360011002572), funded by the AVIC Research Institute For Special Structures of Aeronautical Composites.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Davidor, Y.; Davies, B.L. A vision system for the on-line quality control of adhesive beads. *Int. J. Adhes. Adhes.* **1988**, *8*, 33–38. [CrossRef]
2. Sun, C.-M.; Sezgin, O.S.; Forrest, A.K.; Davies, B.L. Use of vision in dispensing control and inspection of adhesives. *Robotica* **1991**, *9*, 371–377. [CrossRef]
3. Davies, B.; Harris, S.; Razban, A.; Efstathiou, J. Application experience of a robotic cell for automated adhesive dispensing. *Math. Comput. Simul.* **1996**, *41*, 419–427. [CrossRef]
4. Maiolino, P.; Woolley, R.; Branson, D.; Benardos, P.; Popov, A.; Ratchev, S. Flexible robot sealant dispensing cell using RGB-D sensor and off-line programming. *Robot. Comput.-Integr. Manuf.* **2017**, *48*, 188–195. [CrossRef]
5. Hanh, L.D.; Thien, C.V. Visual guidance of a sealant dispensing robot for online detection of complex 3D-curve seams. *Int. J. Interact. Des. Manuf.* **2022**, *16*, 1525–1532. [CrossRef]
6. Pereira, F.R.; Rodrigues, C.D.; Souza, H.S.; Cruz Neto, J.O.; Rocha, M.C.; Barbosa, G.F.; Shiki, S.B.; Inoue, R.S. Force and vision-based system for robotic sealing monitoring. *Int. J. Adv. Manuf. Technol.* **2023**, *126*, 391–403. [CrossRef]
7. Prezas, L.; Arkouli, Z.; Michalos, G.; Angelakis, P.; Gkournelos, C.; Makris, S. A multi-purpose robot perception system enabling closed-loop control for zero defect manufacturing in gluing processes of large parts. *Robot. Auton. Syst.* **2024**, *181*, 104778. [CrossRef]
8. Qiao, Y.; Ye, Y.; Su, F.; Sun, G. Automated high-resolution 3D inspection methods for sealant applications in aerospace based on line structured light. *Rev. Sci. Instrum.* **2025**, *96*, 015101. [CrossRef]
9. Sarti, G.; Kawano, F.L.F.; Shiki, S.B.; Malavolta, A.T.; Barbosa, G.F. Data technologies in robotic sealing: A systematic review on inspection and quality control. *Int. J. Adv. Manuf. Technol.* **2026**, *144*, 6461–6479. [CrossRef]
10. Wang, N.; Zhong, K.; Shi, X.; Zhang, X. A robust weld seam recognition method under heavy noise based on structured-light vision. *Robot. Comput.-Integr. Manuf.* **2020**, *61*, 101821. [CrossRef]
11. Chen, S.; Liu, J.; Chen, B.; Suo, X. Universal fillet weld joint recognition and positioning for robot welding using structured light. *Robot. Comput.-Integr. Manuf.* **2022**, *74*, 102279. [CrossRef]
12. Chen, S.; Li, W.; Li, W.; Gao, X.; Gao, Z.; Li, Y. Fast extraction of weld seam features using a laser stripe contour in line-structured light vision. *Appl. Opt.* **2026**, *65*, 1828–1840. [CrossRef]
13. Yu, C.; Xiao, B.; Gao, C.; Yuan, L.; Zhang, L.; Sang, N.; Wang, J. Lite-HRNet: A lightweight high-resolution network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021*; IEEE: New York, NY, USA , 2021; pp. 10435–10445.
14. Wang, A.; Chen, H.; Lin, Z.; Han, J.; Ding, G. RepViT: Revisiting Mobile CNN From ViT Perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 11–15 June 2025*; IEEE: New York, NY, USA, 2024; pp. 15909–15920.
15. Li, Y.; Yang, S.; Liu, P.; Zhang, S.; Wang, Y.; Wang, Z.; Yang, W.; Xia, S.-T. SimCC: A simple coordinate classification perspective for human pose estimation. In *European Conference on Computer Vision (ECCV)*; Springer: Cham, Switzerland, 2022; pp. 89–106.
16. Zhang, F.; Zhu, X.; Dai, H.; Ye, M.; Zhu, C. Distribution-Aware Coordinate Representation for Human Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 14–19 June 2020*; IEEE: New York, NY, USA, 2020; pp. 7091–7100.
17. Li, J.; Bian, S.; Zeng, A.; Wang, C.; Pang, B.; Liu, W.; Lu, C. Human Pose Regression with Residual Log-Likelihood Estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 11–17 October 2021*; IEEE: New York, NY, USA, 2021; pp. 11005–11014.
18. Sun, X.; Xiao, B.; Wei, F.; Liang, S.; Wei, Y. Integral Human Pose Regression. In *European Conference on Computer Vision (ECCV)*; Springer: Cham, Switzerland, 2018; pp. 536–553.

19. Newell, A.; Yang, K.; Deng, J. Stacked Hourglass Networks for Human Pose Estimation. In *European Conference on Computer Vision (ECCV)*; Springer: Cham, Switzerland, 2016; pp. 483–499.
20. Xiao, B.; Wu, H.; Wei, Y. Simple Baselines for Human Pose Estimation and Tracking. In *European Conference on Computer Vision (ECCV)*; Springer: Cham, Switzerland, 2018; pp. 472–487.
21. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep High-Resolution Representation Learning for Human Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019*; IEEE: New York, NY, USA, 2019; pp. 5686–5696.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.