2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Lightweight Learning-Based Feature Selection for Real-Time Optical Flow Navigation on a Quadrotor Platform

Ali Abosaad, and Jinjun Shan

Abstract— **Accurate state estimation in GPS-denied environments is critical for autonomous quadrotor navigation. Conventional visual-inertial odometry (VIO) pipelines rely on dense feature extraction and tracking, which increases computational cost and is prone to drift when low-quality features dominate. Although learning-based detectors improve robustness, most are too computationally heavy for embedded deployment. This paper proposes a lightweight learning-based feature selection framework that prunes unreliable features to enable efficient optical flow navigation. A compact Convolutional Neural Network (CNN) is employed, with its pruning threshold adaptively adjusted to maintain a stable number of reliable features. The CNN augments ORB and Lucas–Kanade optical flow in a multithreaded pipeline with rotational false-velocity compensation and EKF fusion. Experiments on the Quanser QDrone2 demonstrate up to 75–80% reduction in position RMSE and approximately 25–30% reduction in computation time compared to the Fourier-based Phase Correlation (FPC) method, while sustaining real-time performance above 120 Hz without reliance on external localization systems.**

I. INTRODUCTION

Autonomous quadrotors require reliable and accurate state estimation to operate in environments where GPS signals are not available or degraded, such as indoors, urban canyons, or under dense foliage. Recent advances in multisensor fusion, particularly the integration of visual, inertial, and LiDAR modalities, have demonstrated robust real-time navigation in GPS-denied environments, achieving sub-meter accuracy even under significant sensor noise and environmental variation [1] [2]. These advances have accelerated the adoption of quadrotors for safety-critical applications, including search and rescue, infrastructure inspection [3], and autonomous racing in cluttered environments [4].

A central bottleneck in these systems lies in the quality of visual features extracted from onboard cameras. Classical detectors such as Shi–Tomasi [5], FAST [6], and ORB [7], often coupled with Lucas–Kanade optical flow [8], remain widely used because of their computational efficiency and compatibility with embedded processors. However, these methods frequently extract redundant or unstable features in low-texture or dynamic scenes, inflating computation and causing drift in long-term navigation.

Learning-based detectors have emerged as powerful alternatives, with approaches such as SuperPoint [9], LF-Net [10], R2D2 [11], and LLF [12] demonstrating significantly improved robustness to illumination changes, motion blur, and viewpoint variation. More recent work has introduced event-based detection for low-latency navigation [17]. Although these methods outperform handcrafted features in accuracy and repeatability, their high computational demands remain prohibitive for real-time onboard deployment on resource-constrained aerial platforms. Visual-Inertial Odometry (VIO) frameworks, such as MSCKF [13] and nonlinear optimization-based methods [14], have become central to quadrotor navigation in GPS-denied environments. Yet, these systems remain heavily dependent on the quality of visual features provided to them. An abundance of low-quality points can quickly destabilize estimation and amplify drift, while overly aggressive filtering risks discarding valuable information. Recent efforts have sought to address this tension, through adaptive thresholding strategies [15], hybrid fusion with UWB or LiDAR sensors [19], or even completely replacing feature tracking with Fourier transform methods.

Among recent methods for velocity estimation, Fourier-based Phase Correlation (FPC) [16] has gained attention. Working in the frequency domain, it avoids explicit keypoint detection and matching by directly estimating UAV velocity from monocular images. This makes it robust to lighting changes and texture variations, and it has shown promising results for navigation without GPS. However, the method depends on global spectral alignment, which can suffer from aliasing and requires careful windowing. In addition, it does not take advantage of long-lived local features, which are often key to reducing drift during longer flights.

Despite these advances, there is still no lightweight and adaptive framework tailored for real-time quadrotor navigation. This gap calls for a different solution. Instead of replacing classical detectors with heavy deep-learning models or purely frequency-domain methods, this work introduces a compact learning-based approach that builds on their strengths. A small convolutional neural network is trained to score patch survival and filter out unstable features, while an adaptive threshold maintains a consistent number of reliable keypoints. Together, these techniques reduce redundancy, keep only trackable features, and maintain low computational cost, enabling robust real-time optical flow navigation on resource-limited quadrotors.

The contributions of this work are as follows:

1) A lightweight CNN-based feature selection strategy trained on patch survival labels, enabling tracking-driven pruning of unreliable features for real-time optical flow navigation.

The authors are with the Department of Earth & Space Science and Engineering, York University, Toronto, ON M3J 1P3, Canada (e-mail: alish888@yorku.ca; jjshan@yorku.ca).

2) An adaptive thresholding mechanism that dynamically regulates the number of detected keypoints, maintaining a stable feature supply across varying environments.

The remainder of this paper is organized as follows: Section II details the proposed methodology. Section III describes the experimental setup, results, and discusses the limitations and insights, while Section IV concludes with directions for future research.

II. Methodology

The proposed framework integrates adaptive feature detection, lightweight feature pruning, and efficient optical flow to achieve accurate velocity estimation without reliance on external ground truth. The pipeline consists of five main stages: (i) CNN-based pruning, (ii) adaptive thresholding, (iii) Lucas–Kanade tracking, (iv) IMU–range compensation, and (v) Extended Kalman Filter. The overall system is shown in Fig. 1.

A. CNN-Based Feature Pruning

To reduce redundancy and suppress unstable points, we employ a lightweight convolutional neural network (CNN) that predicts the trackability of candidate features. Each detected keypoint $\mathbf{u}_i^t$ defines a 32×32 grayscale patch p_i^t, which serves as the CNN input. The network outputs a probability score

$$s_i^t = \sigma(f_\theta(p_i^t)) \in [0,1], \tag{1}$$

where f_θ denotes the CNN with parameters θ, and $\sigma(\cdot)$ is the sigmoid activation. Features with $s_i^t > \tau_{\text{cnn}}$ are retained:

$$\mathcal{F}_t^\star = \{\mathbf{u}_i^t \mid s_i^t > \tau_{\text{cnn}}\}, \tag{2}$$

where τ_{cnn} is an adaptive threshold. This ensures that only trackable, stable points are propagated, while the retained count $|\mathcal{F}_t^\star|$ adapts to scene content and motion dynamics.

Architecture.: As shown in Fig.2, The CNN is deliberately compact to enable real-time execution. It consists of two convolutional blocks followed by fully connected layers:

- Conv1: 3×3 kernels, $1\to16$ channels, ReLU, MaxPool(2)
- Conv2: 3×3 kernels, $16\to32$ channels, ReLU, MaxPool(2)
- FC1: $2048\to64$, ReLU, Dropout(p=0.2)
- FC2: $64\to1$, Sigmoid

The total parameter count is approximately 1.36×10^5.

Training Data.: Labels are automatically generated from the EuRoC MAV dataset [20]. For each candidate feature, a 32×32 patch is extracted in the first frame and tracked using the Lucas–Kanade optical flow method across L=5 consecutive frames. If the feature survives all L frames, the patch is labeled as *trackable* (y=1); otherwise it is labeled *unreliable* (y=0). This procedure produces a balanced dataset of patches with survival-based supervision.

Objective and Training.: The CNN is trained as a binary classifier with binary cross-entropy (BCE) loss:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\Big[y_i \log s_i + (1-y_i)\log(1-s_i)\Big], \tag{3}$$

where N is the number of training patches, $y_i \in \{0,1\}$ is the label, and $s_i = \sigma(f_\theta(p_i))$. Training is performed with the Adam optimizer ($\eta = 10^{-3}$, batch size 64) for 10 epochs. Dropout is used for regularization. The network converges quickly due to the low-dimensional input.

Inference.: At runtime, patches are extracted around candidate features detected by ORB. Each patch is scored by the CNN, and only features exceeding τ_{cnn} are retained for optical flow tracking. The threshold is updated dynamically within $[0.5, 0.8]$ to maintain a sufficient number of features under varying scene conditions.

B. Adaptive Thresholding

In order to balance feature count and stability, the pruning threshold τ_{cnn} is updated dynamically between 0.5 and 0.8. The procedure is outlined in Algorithm 1.

Algorithm 1 Adaptive Threshold Update

Require: Current threshold τ, number of selected features N

$\tau_{\min} \leftarrow 0.50$, $\tau_{\max} \leftarrow 0.80$
if $N \le 10$ **then**
  $\tau \leftarrow \max(\tau_{\min}, \tau - 0.01)$
else if $N \ge 25$ **then**
  $\tau \leftarrow \min(\tau_{\max}, \tau + 0.01)$
end if
return Updated τ

The adjustment step size (±0.01) was selected to provide gradual adaptation of the pruning threshold while avoiding oscillatory behavior in the retained feature count across consecutive frames. A larger step size may cause abrupt fluctuations in the number of selected keypoints, whereas a smaller step would slow responsiveness to scene changes. The hard bounds $[0.5, 0.8]$ constrain the threshold within a stable operating region, preventing degenerate cases under extreme illumination or low-texture conditions. This lightweight update rule requires only the current feature count and runs in constant time, making it suitable for real-time deployment on embedded hardware.

C. Optical Flow Tracking

Retained features $\mathcal{F}_t^\star$ are tracked across frames using Lucas–Kanade optical flow [8]:

$$\mathbf{u}_i^{t+1} = \mathbf{u}_i^t + \Delta\mathbf{u}_i, \quad \Delta\mathbf{u}_i = \text{LK}(\mathbf{I}_t, \mathbf{I}_{t+1}, \mathbf{u}_i^t). \tag{4}$$

The mean pixel displacement is then used for velocity estimation.

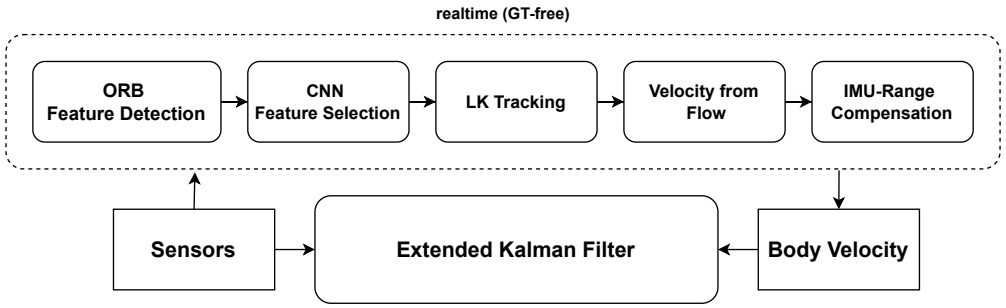

Fig. 1. Overview of the proposed pipeline: Adaptive detection → CNN pruning → LK tracking → IMU–range compensation → EKF.

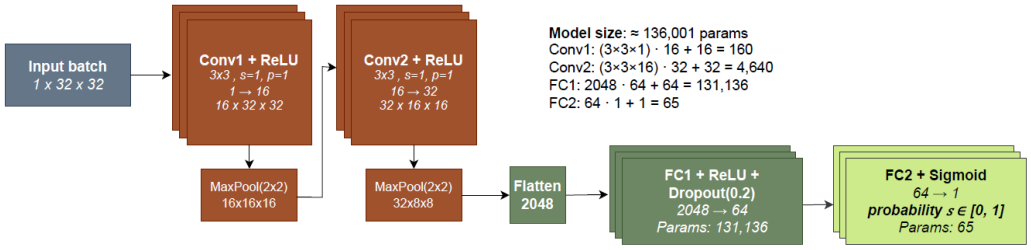

Fig. 2. Lightweight CNN used for feature pruning. Input is a 32×32 grayscale patch; the network outputs a trackability probability used to keep features with $s > \tau_{\text{cnn}}$.

D. Velocity Estimation and Compensation

Let (f_x, f_y) be focal lengths in pixels, h the altitude from the range sensor, and Δt the frame interval. The average translational velocity from optical flow is

$$v_x = \frac{h}{f_x \Delta t} \frac{1}{|\mathcal{F}_t^\star|} \sum_{i \in \mathcal{F}_t^\star} \Delta u_i, \quad v_y = \frac{h}{f_y \Delta t} \frac{1}{|\mathcal{F}_t^\star|} \sum_{i \in \mathcal{F}_t^\star} \Delta v_i. \tag{5}$$

We explicitly correct for rotational false velocities, as they are the dominant structured error source. Other spurious effects (illumination, lens distortion, small-scale feature mismatches) are modeled implicitly within the Kalman filter noise terms, since they are either stochastic or significantly smaller in magnitude compared to the rotational contribution.

Rotationally induced motion is corrected using IMU angular rates $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\top$:

$$\mathbf{v}_{\text{corr}} = \mathbf{v}_{\text{opt}} - \boldsymbol{\omega} \times \begin{bmatrix} 0 \\ 0 \\ h \end{bmatrix} = \begin{bmatrix} v_x - \omega_y h \\ v_y + \omega_x h \end{bmatrix}. \tag{6}$$

E. Extended Kalman Filter for State Estimation

To fuse the heterogeneous sensor streams into a consistent state, we employ an Extended Kalman Filter (EKF). The EKF maintains a 16-dimensional state vector

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} & \mathbf{v} & \mathbf{q} & \mathbf{b}_\omega & \mathbf{b}_a \end{bmatrix}^\top,$$

where $\mathbf{p} \in \mathbb{R}^3$ is position, $\mathbf{v} \in \mathbb{R}^3$ velocity, $\mathbf{q}$ the attitude quaternion, and $\mathbf{b}_\omega, \mathbf{b}_a$ the gyroscope and accelerometer biases.

The filter prediction step is driven by the IMU propagation model. At each timestep k, the state and covariance are propagated using

$$\mathbf{x}_{k|k-1} = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}), \quad \mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^\top + \mathbf{Q},$$

where $\mathbf{P}$ is the prediction covariance, $\mathbf{F}$ is the Jacobian of the motion model and $\mathbf{Q}$ is the process noise covariance, it is tuned according to IMU noise densities and bias random walks .

The update step integrates exteroceptive sensor measurements. The downward-facing camera provides velocity measurements in the horizontal plane,

$$\mathbf{z}_{\text{cam}} = \begin{bmatrix} v_x \\ v_y \end{bmatrix},$$

while the range sensor constrains the vertical velocity

$$\mathbf{z}_{\text{range}} = v_z,$$

and the IMU gyroscope provides angular velocity measurements

$$\mathbf{z}_{\text{imu}} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}.$$

The measurement update follows the standard EKF formulation:

$$\mathbf{K} = \mathbf{P}_{k|k-1}\mathbf{H}^\top \left(\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^\top + \mathbf{R}\right)^{-1},$$

$$\mathbf{x}_k = \mathbf{x}_{k|k-1} + \mathbf{K}(\mathbf{z}_k - h(\mathbf{x}_{k|k-1})), \quad \mathbf{P}_k = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}_{k|k-1},$$

where $\mathbf{H}$ is the measurement Jacobian, and $\mathbf{R}$ is the measurement noise covariance, derived from sensor calibration statistics (optical flow variance and range sensor precision).

This formulation allows direct fusion of camera-based velocity estimates with inertial and range data, providing

a consistent real-time estimate of the full quadrotor state. The EKF serves as the backbone estimator, ensuring that the proposed feature-selection pipeline leads to improvements at the state-estimation level.

F. Runtime Execution

The full pipeline runs in real time. To ensure throughput $> 120\,\text{Hz}$, two threads are used: (i) a capture thread for frame acquisition, and (ii) a processing thread for feature detection, pruning, flow tracking, and velocity estimation. This decoupling prevents frame drops and maintains deterministic latency.

G. Validation and Evaluation

The proposed framework is validated using both dataset-based experiments and hardware-in-the-loop testing. Specifically, the EuRoC MAV dataset [20] is employed to train and evaluate the CNN pruning and adaptive thresholding under controlled conditions, while indoor hardware tests on the QDrone2 are performed with OptiTrack ground truth for evaluation only (not used during runtime).

Performance is assessed along three dimensions:

- **Accuracy**: Root Mean Square Error (RMSE) of velocity and position compared against ground truth.
- **Efficiency**: Frames per second (FPS) and average processing latency.
- **Robustness**: Stability of retained features across varying environments.

III. EXPERIMENTAL STUDY

This section presents the implementation and validation of the proposed pipeline. It begins with a description of the hardware platform and the ROS 2 software framework, followed by the experimental setup and the obtained results. The section also includes a discussion highlighting the key insights and limitations observed.

A. *Experimental setup*

Experiments are conducted on the Quanser QDrone2 platform shown in Fig. 3. The vehicle is equipped with:

- A downward-facing global omnivision shutter grayscale camera (up to 210 fps),
- Inertial Measurement Units (IMUs) providing angular rates and linear accelerations,
- A downward-facing range sensor for altitude estimation.

The onboard computer runs ROS 2 Eloquent, where the camera publisher node executes, while a ground station running ROS 2 Humble hosts the feature detection and selection node.

ROS 2 Software framework as shown in Fig. 4 is implemented as a set of ROS 2 nodes:

1) **Camera Publisher**: Publishes downward-facing camera frames,
2) **Feature Detection & Selection Node**: Performs ORB detection, adaptive CNN-based pruning, Tracks retained features using Lucas–Kanade and estimates velocities, and corrects optical flow velocities using IMU angular rates and range altitude,

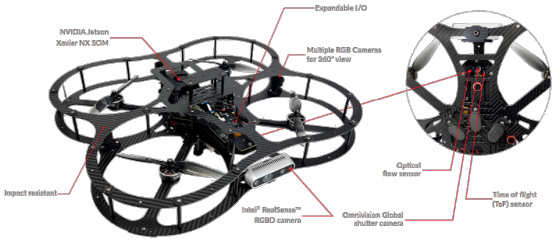

Fig. 3. QDrone2 hardware platform with equipped sensors.

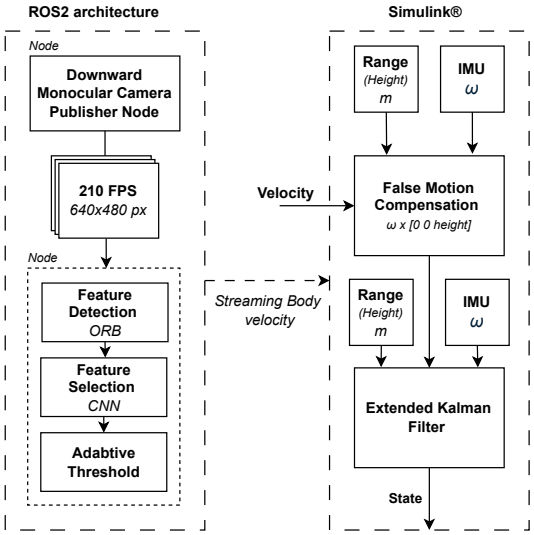

Fig. 4. System architecture. Shows ROS Architecture on the left and data getting passed from the camera publisher to the feature selector, and then velocity computed and streamed to Simulink® for compensating the false motion and estimating the state.

The computed velocity is streamed to a Simulink® model, where false-velocity compensation is applied. The velocity, computed using IMU angular rates and range sensor altitude, is fused within an EKF to produce the full state estimate.

B. *Experimental Results*

The evaluation focuses on five key aspects: (i) feature selection behavior, (ii) adaptive thresholding effectiveness, (iii) position estimation accuracy, (iv) computational efficiency, and (v) end-to-end onboard performance.

The evaluation of the feature selection should not only verify that a sufficient number of keypoints is detected, but also ensure that these retained keypoints are of high quality and reliability. Fig. 5 shows the per-frame number of ORB candidates N_t and the CNN-retained features $|\mathcal{F}_t^\star|$, where pruning effectively reduces redundancy while maintaining a stable subset. Beyond quantity, the distribution of feature lifetimes in Fig. 6 highlights that the selected keypoints persist significantly longer than the overall candidate set, indicating that the retained features are more stable and reliable. This confirms that the proposed pruning strategy not

only controls feature counts but also prioritizes long-lasting keypoints that strengthen downstream state estimation.

The evaluation of the adaptive threshold focuses on its ability to regulate the number of retained keypoints across frames. As shown in Fig. 7, the curve illustrates both the retained keypoint count and the corresponding threshold updates applied by the adaptive rule. This mechanism dynamically adjusts the threshold upward when too many features are passed and downward when too few are retained, thereby keeping the feature count within the desired bounds. Maintaining this balance is critical: if the number of retained keypoints is too low, the estimation becomes overly sensitive to noise and outliers; conversely, if it is too high, redundant and unstable features inflate computation and degrade accuracy. The adaptive update thus ensures a sufficient, stable, and reliable feature set across varying scenes, contributing directly to robust downstream state estimation.

As shown in Fig. 8, the proposed method's estimated trajectory aligns closely with the ground truth, whereas the Fourier-based phase correlation (FPC) [17] exhibits larger deviations and noticeable drift under varying lighting and motion conditions. This qualitative advantage is further reflected in Table I, where our method consistently achieves RMSE values around 0.05–0.06 m, in contrast to the FPC baseline, which exceeds 0.2 m and reaches up to 0.3 m in more challenging scenarios. Complementing the trajectory analysis, Fig. 9 presents a velocity comparison along a simple back-and-forth motion on the x-axis. In this controlled setting, our method tracks the OptiTrack reference closely, capturing both the direction changes and magnitude with improved consistency across direction changes. By comparison, the FPC approach follows the overall trend but shows greater fluctuations and lag. Together, these results demonstrate that adaptive feature selection and CNN pruning not only stabilize the feature set but also provide long-lived, reliable keypoints, ultimately enhancing both position and velocity estimation accuracy during flight.

The computational efficiency of the proposed method is summarized in Table I which compares our method with FPC in terms of RMSE, computation time, and processing frequency across different lighting and trajectory conditions. The proposed pipeline consistently achieves lower error while maintaining an average computation time of around 8 ms per frame, corresponding to over 120 Hz. In contrast, FPC operates at significantly higher latency and lower frequency. These results confirm that our approach delivers real-time performance with significantly improved accuracy and approximately 25–30% reduction in computation time. All experiments were conducted on an Intel® Core™ i7-8700 CPU @ 3.20 GHz × 12 with 16 GB RAM.

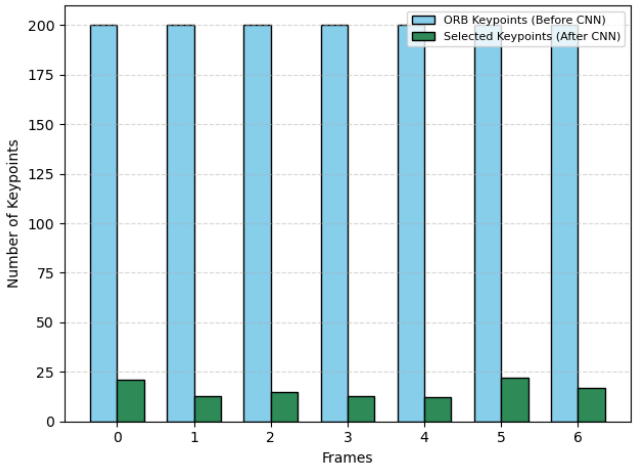

Fig. 5. Candidates vs. retained features per frame: ORB candidates N_t (blue) and CNN-retained $|\mathcal{F}_t^\star|$ (green).

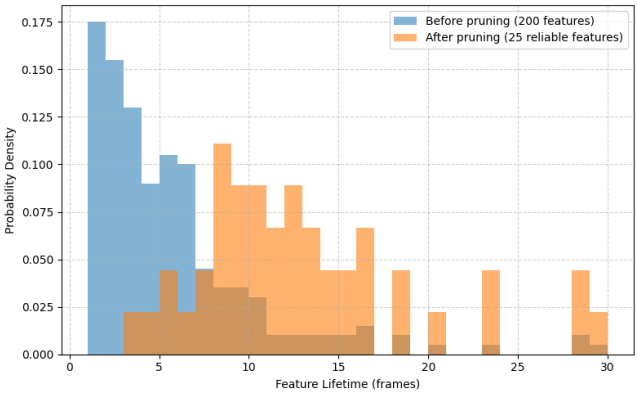

Fig. 6. Feature lifetime distribution (in frames). Pruning reduces the dominance of short-lived features while promoting long-lived, stable tracks that strengthen position estimation.

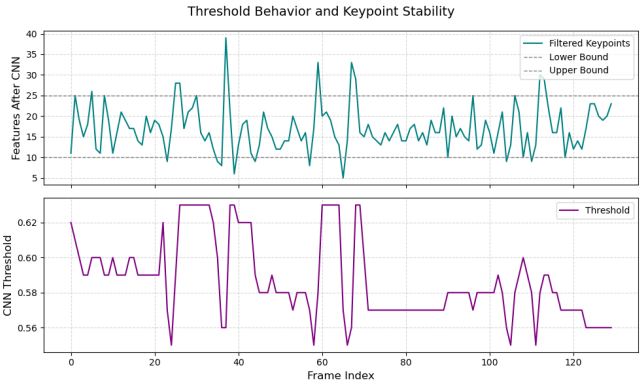

Fig. 7. ORB candidate count N_t with fixed vs. adaptive threshold. The adaptive rule maintains a stable supply near N_{target}.

C. *Ablation Studies*

We evaluate the contribution of each module by progressively enabling (i) adaptive thresholding (AT) and (ii) CNN-based feature pruning (CNN). Results are averaged over validation flights under bright, medium, and low-light conditions. We report position accuracy as RMSE (m), along with end-to-end latency and the corresponding processing frequency.

Table II shows that adding the CNN-based pruning module

TABLE I

COMPARISON BETWEEN OUR METHOD AND FOURIER-BASED PHASE CORRELATION METHOD UNDER DIFFERENT LIGHTING AND TRAJECTORIES

Lighting / Trajectory	Method	RMSE (m)	Computation Time (ms)	Frequency (Hz)
Bright / Straight	FPC	0.197	11.23	89.0
	Ours	**0.048**	**8.21**	**121.8**
Medium / Curved	FPC	0.213	12.05	83.0
	Ours	**0.058**	**8.11**	**123.4**
Low-light / Complex	FPC	0.304	10.78	92.7
	Ours	**0.062**	**8.23**	**121.5**

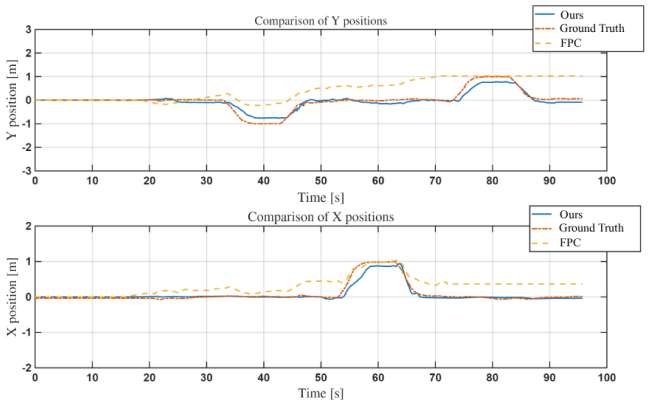

Fig. 8. Position trajectory comparison under indoor flight conditions. The proposed method (CNN-based feature pruning with adaptive threshold) is compared against the Fourier-based Phase Correlation (FPC) approach with respect to OptiTrack ground truth. The proposed method achieves a position RMSE of 0.058 m, while FPC results in a larger error of 0.213 m, demonstrating reduced drift and improved trajectory consistency.

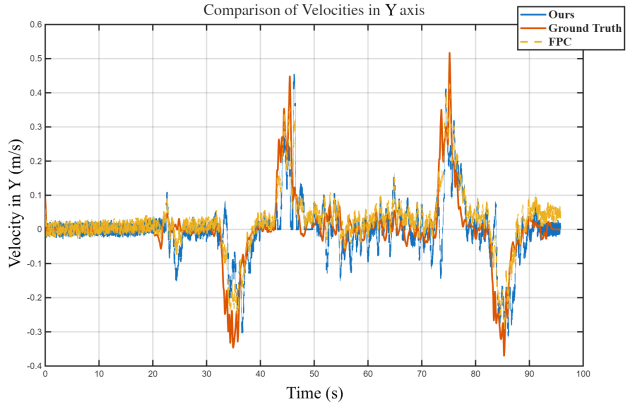

Fig. 9. Velocity estimation comparison along the x-axis during a back-and-forth motion. The proposed method is compared against the Fourier-based Phase Correlation (FPC) approach with respect to ground truth. The proposed method tracks direction changes and velocity magnitude more consistently, exhibiting reduced bias and improved temporal alignment.

substantially improves position accuracy, reducing RMSE from 0.519 m to 0.09 m (**82.7% reduction**) while maintaining real-time operation at approximately 122 Hz. Incorporating the full system with adaptive thresholding further refines the feature set, lowering the error to 0.058 m (**88.8% reduction** relative to baseline) while sustaining a processing frequency near 121.5 Hz (≈8.23 ms per frame).

TABLE II

ABLATION OF PIPELINE COMPONENTS ON POSITION ACCURACY AND FREQUENCY

Variant	**RMSE (m)**	**Freq. (Hz)**
Baseline (fixed ORB, LK only)	0.519	130
+ CNN	0.09	122.1
+ AT + CNN (**Ours**)	**0.058**	**121.5**

Overall, the complete system (**Ours**) achieves nearly a **88% reduction** in position error (0.519 m → 0.058 m) compared to the baseline, with only a modest decrease in frequency (130 Hz → 121.5 Hz). These results confirm that each module contributes positively: CNN pruning dramatically improves feature reliability, while adaptive thresholding stabilizes feature supply, culminating in a pipeline that is significantly more accurate while preserving real-time performance for onboard state estimation.

D. *Discussion*

a) Effect of Feature Regulation and Pruning.: The ablation results highlight the importance of tailoring the feature extraction process for onboard state estimation. While the baseline pipeline achieves only moderate accuracy at 0.519 m RMSE, it also suffers from unstable feature counts that translate into noisy trajectories. Adaptive thresholding addresses this by regulating the number of tracked features, leading to both improved accuracy and reduced latency. Incorporating CNN-based pruning delivers the most significant gain, cutting the error to 0.058 m while maintaining real-time execution above 120 Hz.

b) Accuracy–Efficiency Tradeoff.: These findings suggest two key insights. First, accuracy and efficiency are not mutually exclusive: carefully designed feature selection can simultaneously lower computational load and improve estimation quality. Second, lightweight learned components, when integrated with classical optical flow tracking, provide a balance between robustness and real-time performance on resource-limited platforms platforms. This validates the goal of our method, achieving high frequency, low latency state estimation without sacrificing reliability under varying flight conditions.

c) Additional Baseline Considerations.: Although the primary quantitative comparison emphasizes the Fourier-based Phase Correlation (FPC) method due to its suitability for monocular velocity estimation without explicit feature

tracking, additional evaluations were conducted using classical ORB-only and learned SuperPoint-based pipelines on the QDrone2 platform. While absolute performance values differ across hardware platforms, the observed trends remain consistent: ORB exhibits higher drift due to unstable feature retention, whereas SuperPoint improves feature robustness at the cost of substantially increased computational load. The proposed hybrid approach achieves a more favorable accuracy–latency tradeoff by combining lightweight learned pruning with classical tracking, reinforcing its suitability for real-time deployment on resource-constrained aerial systems.

d) Contextualization with Lightweight VIO Systems.: While the proposed framework focuses on optical-flow-based velocity estimation fused within an EKF, rather than full nonlinear optimization or bundle-adjustment visual–inertial odometry (VIO), the achieved indoor positioning RMSE falls within the range reported by lightweight VIO systems under comparable controlled conditions (e.g., OpenVINS [18]). A direct numerical comparison is not strictly equivalent due to differences in sensor configurations, trajectories, and evaluation protocols. Nevertheless, the presented results highlight a favorable accuracy–latency tradeoff, demonstrating that high-frequency optical-flow fusion can achieve competitive accuracy while maintaining substantially lower computational complexity for resource-limited aerial platforms.

e) Generalization and Label Definition.: Although the CNN is trained on EuRoC, the learned notion of *trackability* is largely driven by local patch structure (e.g., cornerness/gradient distribution) and its temporal stability under inter-frame motion, rather than scene specific appearance. As a result, the classifier transfers well to our indoor OptiTrack environment: patches that remain reliably trackable over multiple consecutive frames tend to share the same low-level characteristics across domains.

Trackability labels are generated using a survival criterion: a keypoint is labeled positive ($y = 1$) only if it can be successfully tracked for *at least* L consecutive frames (here $L = 5$); otherwise it is labeled negative ($y = 0$). This choice controls the strictness of supervision. Smaller values (e.g., $L = 3$) yield looser labels that admit short-lived features and can increase the false-positive rate, while larger values (e.g., $L > 5$) over-constrain the positives by favoring only very long-lived tracks, which can reduce the available feature supply under fast or aggressive motion. We select $L = 5$ as a practical tradeoff that emphasizes temporal consistency without excessively shrinking the positive set.

IV. CONCLUSIONS

This paper presented a lightweight learning-based feature selection framework for real-time quadrotor navigation in GPS-denied environments. By integrating CNN-based pruning with adaptive threshold regulation and classical optical flow tracking, the proposed pipeline improves feature reliability while preserving computational efficiency. Compared to the Fourier-based Phase Correlation (FPC) method, the approach achieves up to **75–80% reduction in position RMSE** and approximately **25–30% reduction in computation time**, while sustaining execution above 120 Hz on embedded hardware. Relative to a classical ORB-based baseline, the proposed method reduces position error by approximately **88%**. These results demonstrate that lightweight learned feature regulation can substantially enhance optical-flow-based state estimation while preserving deterministic real-time performance on resource-constrained aerial platforms.

REFERENCES

[1] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2015.

[2] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *Int. J. Robotics Research*, vol. 34, no. 3, 2015.

[3] A. Loquercio, A. I. Maqueda, C. R. del-Blanco, and D. Scaramuzza, "Dronet: Learning to fly by driving," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, 2018.

[4] R. Penicka, Y. Song, M. Wulfmeier, and D. Scaramuzza, "Learning minimum-time flight in cluttered environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5719–5726, Apr. 2022.

[5] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 1994.

[6] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conf. Computer Vision (ECCV)*, 2006.

[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *IEEE Int. Conf. Computer Vision (ICCV)*, 2011.

[8] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Int. Joint Conf. Artificial Intelligence (IJCAI)*, 1981.

[9] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *IEEE Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.

[10] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "LF-Net: Learning local features from images," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[11] J. Revaud, P. Weinzaepfel, C. de Souza, M. Humenberger, and G. Puy, "R2D2: Reliable and repeatable detector and descriptor," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[12] H. Luo, Z. Luo, K. Yang, *et al.*, "LLF: Learning of Low-Level Feature Keypoints for robust image matching," in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020.

[13] A. I. Mourikis and S. I. Roumeliotis, "A Multi-State Constraint Kalman Filter for vision-aided inertial navigation," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2007, pp. 3565–3572.

[14] A. Batool, F. Batool, R. A. Khan, M. A. Mustafa, A. Fedoseev, and D. Tsetserukou, "NMPC-Lander: Nonlinear MPC with barrier function for UAV landing on a mobile platform," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2014.

[15] J. Zhang, S. Singh, and D. Scaramuzza, "Adaptive feature selection for visual odometry in dynamic environments," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2021.

[16] H. Deng, D. Li, B. Shen, Z. Zhao, and U. Arif, "Absolute velocity estimation of UAVs based on phase correlation and monocular vision in unknown GNSS-denied environments," *IET Image Processing*, vol. 18, no. 9, pp. 3218–3230, 2024, doi: 10.1049/ipr2.13167.

[17] D. Gehrig, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza, "Event-based visual odometry with learned feature tracking," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, 2021.

[18] J. Geneva, K. Eckenhoff, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2020.

[19] Z. Yang, Y. Yuan, S. Shen, and F. Zhang, "Multi-sensor fusion for robust autonomous flight in cluttered environments," *IEEE Trans. Robotics*, vol. 38, no. 3, pp. 541–561, 2022.

[20] M. Burri, J. Nikolic, P. Gohl, *et al.*, "The EuRoC micro aerial vehicle datasets," *Int. J. Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.