Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Accuracy, efficiency and corruption robustness of CPU- constrained infrared bird–drone video classification

Yucheng Wang[1]

1455691502@qq.com

Nanjing University of Science and Technology

Yu Wang[1]

Nanjing University of Science and Technology

Research Article

Keywords: infrared video, bird–drone classification, lightweight CNN, CPU inference, corruption robustness, temporal sampling

Posted Date: November 26th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11551769/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Abstract

Infrared bird–drone classification in low-altitude surveillance requires a practical balance among accuracy, processor demand, and sensitivity to degraded imagery. We evaluate the post-localization classification of 16-frame target regions of interest using two public infrared data sources, with group-disjoint training, development, and held-out evaluation partitions. The three-seed, equal-dataset, source-group-weighted macro-F1 scores are 0.9638 ± 0.0181 for ResNet18 with temporal mean pooling and 0.9727 ± 0.0009 for ShuffleNetV2 with the same pooling rule. A paired group bootstrap gives a ShuffleNetV2-minus-ResNet18 difference of 0.0090 (95% confidence interval: -0.0138 to 0.0278), which does not establish an accuracy advantage. On a single-thread Intel Core i7-12700F, their network-only median inference latencies are 64.15 and 14.85 ms, respectively. However, across ten predefined synthetic corruptions, mean macro-F1 losses are 0.1634 and 0.2046, respectively, with pronounced class-specific failures under severe noise. Separate development-set experiments compare sampling 1–32 frames and alternative spatial and spatiotemporal encoders: increasing frame count yields modest accuracy gains at higher offline inference costs, while R(2 + 1)D-18 improves development-set accuracy at substantially greater CPU cost. The resulting evidence characterizes accuracy–efficiency–robustness trade-offs in localized infrared video classification and informs model selection under CPU constraints.

Introduction

Birds and unmanned aerial vehicles (UAVs) are visually confusable in low-altitude surveillance, especially when distant targets occupy only a small number of infrared pixels. Once a detector or tracker has localized a candidate, a lightweight classifier must interpret a sequence of target-centered image patches under a restricted computational budget. Averaging frame-level features can aggregate appearance evidence cheaply, whereas explicitly spatiotemporal architectures can extract additional temporal structure at greater cost. These alternatives must therefore be compared using both predictive performance and measured inference cost, rather than parameter counts alone.

Existing multisensor and trajectory datasets support research on distinguishing aerial targets [1, 2]. Sequence classifiers, motion and trajectory features, and integrated detection–tracking systems provide complementary approaches [3, 4, 12, 15]. Modern UAV detection systems also demonstrate progress in full-frame localization [13, 14]. However, object detection, trajectory classification, and the classification of already localized regions of interest (ROIs) are different tasks. Results are not directly transferable across these task definitions, datasets, or evaluation protocols.

We investigate the latter task through binary classification of previously localized, 16-frame infrared ROI sequences, addressing three connected research questions. First, how do pretrained lightweight spatial encoders with temporal mean pooling compare with more expensive temporal representations? Second, how do their source-group-aware accuracy, CPU latency, and memory demands interact? Third, how do temporal sampling density and input corruptions alter that trade-off?

The study contributes (i) a source-group-disjoint evaluation across two public infrared datasets, with three-seed held-out performance and paired uncertainty estimates; (ii) a joint analysis of classification accuracy, measured single-thread CPU costs, and sensitivity to ten predefined corruptions, including source–class recall; and (iii) controlled development-set comparisons of frame sampling and alternative video representations. Together, these analyses reveal when gains in computational efficiency are accompanied by changes in recognition reliability.

Related work

Infrared bird–drone discrimination

Svanström et al. [1] provided a multisensor dataset relevant to drone surveillance, while Ke et al. [2] introduced LAT-BirdDrone for low-altitude small-target trajectory classification. Akyon et al. [3] evaluated alternative video sequence models. Luesutthiviboon et al. [4] studied the complementarity of visual and trajectory features, and Kataria and Lall [12] combined tracking with subsequent sequence classification. These studies motivate a focused evaluation of image evidence after localization, while their differing labels, input modalities, and sample units preclude direct score rankings.

Lightweight video representation and corruptions

ResNet [5] offers a robust convolutional baseline, while ShuffleNetV2 [6] emphasizes hardware-related costs beyond nominal floating-point operation counts. MobileNetV3 [7] is another relevant efficient design. Temporal mean pooling discards frame order but aggregates appearance cues at minimal additional model complexity. Explicit spatiotemporal convolutions such as R(2 + 1)D [10] may recover temporal information at higher cost. Common-corruption evaluation [8] and robustness-oriented augmentation [9] underline the need to assess noise, blur, compression, and spatial perturbations rather than relying solely on clean accuracy. The newer YOLO26 classification encoder is considered as an image-classification component, not as a full-frame detector [11].

Materials and methods

Task and data protocol

The binary labels are bird and drone. Each main-experiment window consists of T = 16 grayscale ROI patches of 64 × 64 pixels, represented by a 16 × 1 × 64 × 64 tensor. The evaluation isolates the post-localization classifier by using target boxes available in the source datasets.

Two sources are used: LAT-BirdDrone (LAT) [2] and the infrared subset of the multisensor dataset (External) [1]. Training draws from both sources, and held-out evaluation measures performance on disjoint source groups within those datasets. A manifest-level grouping strategy separates training, development and test partitions by original video or conservative source-group identity, to mitigate

overlap among highly related windows. Table 1 gives the exact window counts. The LAT held-out set has 9 conservative source groups spanning 25 sequences; the External set contains 33 source videos. The LAT original validation split is not part of the counts in Table 1.11

Dataset partition sizes (number of ROI windows).

Partition	LAT	External	Role
Training	2,251	2,427	Model fitting
Development	2,147	514	Model selection
Held-out test	1,334	517	Final evaluation

Spatial encoders and temporal aggregation

For ROI frame x_t, a convolutional encoder f_θ produces feature $z_t = f_\theta(x_t)$. The window representation and the predicted class distribution are

$\bar{z} = (1/T) \sum_{t=1}^{T} f_\theta(x_t)$; $\hat{p} = \mathrm{softmax}(W\bar{z} + b)$.

ResNet18-Mean uses ImageNet-1K V1 weights and a 512-dimensional feature after removal of the original classification layer; ShuffleNetV2-x1.0-Mean similarly uses ImageNet-1K V1 initialization and a 1,024-dimensional feature. Grayscale inputs are replicated to three channels and normalized using the corresponding ImageNet statistics. The model parameter counts, including the binary classifier, are 11,177,538 and 1,255,654, respectively. This representation averages frame appearance information and does not explicitly infer motion direction or speed. Additional preliminary development architectures include a tiny CNN, GRU and LSTM variants, and MobileNetV3-Small.

Study workflow: construction of group-disjoint infrared ROI sequences, feature aggregation, and bird–drone prediction evaluated across accuracy, CPU cost, and synthetic corruptions.

Training, metrics, and uncertainty

Training uses the combined training partitions with balanced sampling between the two classes, and model checkpoints are selected using development scores. The main comparison uses seeds 42, 3407, and 2027. Both final model families are frozen before the held-out test; the three-seed results are reported without retroactively selecting a favorable seed. Comparisons therefore reflect the tested combinations of architecture, initialization, and preprocessing.

Because a source video may produce many overlapping windows, each window in source group g receives weight $w_i = 1/n_g$, where n_g is the group's window count. Weighted true positives, false positives, and false negatives are accumulated for each class c, yielding

$F1_c = 2TP_c / (2TP_c + FP_c + FN_c)$; Macro-F1 = $(F1_{bird} + F1_{drone})/2$.

The primary reported score averages the source-group-weighted macro-F1 from LAT and External equally. Individual source–class recall and worst-source-group recall are additional diagnostics. A paired bootstrap resamples source groups within dataset–class strata 10,000 times and averages the paired difference over the three fixed seeds. Windows and random seeds are not treated as additional independent collection sources. We interpret a confidence interval spanning zero as insufficient evidence to rank the models by accuracy; it does not establish equivalence.

Corruption and computational protocols

The held-out tests include clean inputs and ten fixed synthetic corruptions: contrast scaling $\alpha \in \{0.7, 0.4\}$, Gaussian noise $\sigma \in \{0.04, 0.08\}$, horizontal motion-blur kernels $k \in \{3, 7\}$, JPEG quality $Q \in \{50, 25\}$, and target-scale factors $s \in \{0.75, 0.50\}$. For each model and seed, the mean corruption loss is the mean over these conditions of clean macro-F1 minus corrupted macro-F1. Each corruption uses the same underlying clean test windows; weights and decision settings remain unchanged.

CPU timing uses an Intel Core i7-12700F with PyTorch eager execution, FP32, batch size one, one intra-/inter-op thread, and single-thread OpenCV. For the main comparison, three independent processes each execute 200 tasks stratified by source and class. Network-only forward time, already-materialized input classification, and warm-source ROI decoding and classification are distinguished. Each reported percentile is the arithmetic mean of the per-process percentiles, not a percentile of pooled observations; peak resident set size (RSS) includes runtime and caches. Reciprocal-P50 window rates are derived from the offline timing measurements rather than sustained throughput observations.

For overlapping one-frame-stride windows, temporal means can be updated by reusing cached frame features:

$\bar{z}(t + 1) = \bar{z}(t) + [f\theta(x(t + 1)) - f\theta(x(t - T + 1))]/T$. (T is the window length.)

This exact feature reuse is only valid when the per-frame encoding is independent of its surrounding window. We evaluate numerical agreement and classifier-level computational savings using a 31-frame sequence.

Results

Clean held-out classification

Table 2

summarizes the final three-seed held-out results. The equal-dataset macro-F1 is 0.9638 ± 0.0181 for ResNet18-Mean and 0.9727 ± 0.0009 for ShuffleNetV2-Mean. The paired ShuffleNetV2-minus-ResNet18 difference is 0.00898 with a 95% bootstrap interval [−0.01385, 0.02775]. Although the point estimate and across-seed spread favor ShuffleNetV2, the group-level interval crosses zero. Accordingly, superiority in accuracy is not established. Both models perform better on the LAT split than on the External split, particularly in the worst-source recall statistic. Clean held-out macro-F1 and source-specific recall (three-seed mean; SD shown for macro-F1).

Statistic	**ResNet18**	**ShuffleNetV2**
External macro-F1	$.9413 \pm .0284$	$.9581 \pm .0079$
LAT macro-F1	$.9862 \pm .0079$	$.9874 \pm .0086$
Equal-dataset macro-F1	$.9638 \pm .0181$	$.9727 \pm .0009$
External bird recall	.9482	.9439
External drone recall	.9469	.9721
External worst-group recall	.4375	.5000
LAT worst-group recall	.9526	.9566

Held-out equal-dataset macro-F1 for the three fixed seeds. Seed-specific values are not independent source groups.

Across the six model instances and the clean plus ten-corruption conditions, the frozen evaluation comprises 122,166 predictions. Despite high aggregate scores, a subset of source videos exhibits low recall, emphasizing the value of source-level reporting for surveillance applications.

CPU costs and feature reuse

Table 3

reports the main CPU timing boundaries. ShuffleNetV2 reduces network-only P50 from 64.15 to 14.85 ms (about 76.9%) and warm-source P50 from 78.48 to 29.26 ms (about 62.7%) compared with ResNet18. The network parameters decrease by approximately 88.8%. RSS differences are much smaller than the parameter-count difference because the measurements include process and caching overhead. For ShuffleNetV2, the warm-source P95 reaches 132.50 ms, highlighting the latency variability introduced by data preparation and classification together. Main experiment: single-thread x86 CPU latency and peak process RSS. P50/P95 values are in ms.

Model	Network	Warm-source ROI	RSS
Tiny CNN	3.92/4.89	18.19/127.40	684.4
ShuffleNetV2	14.85/16.06	29.26/132.50	695.4
ResNet18	64.15/67.40	78.48/185.78	737.7

Network-only and warm-source processing P50 times in the main CPU protocol.

In a separate 31-frame local sliding-window check, the mean classifier-level accelerations from frame-feature reuse are 6.68×, 9.90×, and 8.49× for Tiny CNN, ShuffleNetV2, and ResNet18, respectively, with unchanged discrete predictions and maximum logit difference below 4.77×10^{-7}. These checks demonstrate numerical feasibility of incremental feature averaging, not end-to-end monitoring throughput.

Sensitivity to synthetic corruption

The ten-corruption mean macro-F1 loss is 0.1634 ± 0.0370 for ResNet18 and 0.2046 ± 0.0215 for ShuffleNetV2. Table 4 illustrates that ranking reverses across corruption categories. Severe low contrast favors ShuffleNetV2, whereas severe additive noise, motion blur, and JPEG degradation favor ResNet18. The bootstrap estimates for the ShuffleNetV2-minus-ResNet18 difference are + 0.0831 (95% CI [0.0492, 0.1149]) at contrast $\alpha = 0.4$, and − 0.1594 (95% CI [− 0.1957, − 0.1243]) at Gaussian noise $\sigma = 0.08$. These per-corruption analyses were not adjusted for multiple comparisons and should be considered descriptive.

Equal-dataset macro-F1 under five severe synthetic corruptions (three-seed means).

Condition	ResNet18	ShuffleNetV2
Contrast $\alpha = 0.4$	.6797	.7626
Gaussian noise $\sigma = 0.08$	.5745	.4153
Motion blur $k = 7$	.9494	.8889
JPEG $Q = 25$	.9537	.9102
Target scale $s = 0.50$	.4860	.5619

The clean-model ranking reverses across five severe synthetic corruptions.

More importantly, the post-hoc source–class analysis reveals severe drone failures. Under Gaussian noise σ = 0.08, ShuffleNetV2 drone recall is about .164 for LAT and .009 for External, although bird recall is approximately .999 and 1.000, respectively. Corresponding ResNet18 drone recalls are about .588 and .099. Synthetic target-scale reduction likewise changes class-conditional behavior. These class-dependent errors show why aggregate macro-F1 should be complemented by class-specific recall, and motivate subsequent verification with real sensor degradations.

Frame sampling and modern-encoder trade-offs

Table 5

combines the development-set three-seed classification results with separately measured offline seed-42 CPU timings. On the same 2,549 development windows, 16-frame sampling reaches 0.9277 ± 0.0069 macro-F1 at 15.76 ms fixed cached-ROI classification P50. At 32 frames, the mean macro-F1 increases by about .0067 while P50 rises by 10.07 ms. Because the fixed-ROI timing and the three-seed accuracy estimate have different repetition units, and because the paired accuracy confidence intervals do not establish consistent advantages at all seeds, 16 frames should be viewed only as an empirical cost–performance compromise. The one-frame selection also covers less effective temporal context than the others. Temporal sampling on common development windows: accuracy and separate fixed-ROI CPU test.

Frames	Macro-F1 $\pm$ SD	P50 (ms)	P95 (ms)
1	$.9169 \pm .0123$	4.09	5.11
8	$.9149 \pm .0178$	8.92	10.57
16	$.9277 \pm .0069$	15.76	17.67
32	$.9344 \pm .0099$	25.83	28.70

Development-set macro-F1 against fixed cached-ROI materialization and classification P50 for sampling 1, 8, 16, and 32 frames. Costs are measured offline on cached ROI inputs.

The modern-encoder comparison has different training initializations and internal image resolutions. On the intersection of 2,549 development windows, R(2 + 1)D-18, ShuffleNetV2-Mean, and YOLO26n-cls-Mean obtain macro-F1 scores of 0.9586 ± 0.0062, 0.9277 ± 0.0069, and 0.8780 ± 0.0161, respectively. Their single-thread network-only P50 timings are 1180.32, 16.22, and 112.49 ms (Table 6). Thus R(2 + 1)D-18 shows the highest development-set score in this particular configuration, at approximately 72.75 times the mean network latency of ShuffleNetV2. Because pretraining and internal resolution differ, these development-set results characterize the tested model configurations rather than the isolated effect of network architecture.

Development-only modern-encoder comparison on 2,549 common windows; network-level timing uses seed 42.

Encoder	Macro-F1 $\pm$ SD	CPU P50 (ms)
ShuffleNetV2-Mean	$.9277 \pm .0069$	16.22
YOLO26n-cls-Mean	$.8780 \pm .0161$	112.49
R(2 + 1)D-18	$.9586 \pm .0062$	1180.32

As an exploratory training study using a single development seed, noise-and-scale or mixed-corruption augmentation reduces the mean ten-corruption loss of ShuffleNetV2 from .2531 (unaltered baseline) to .0876 and .0609, respectively. However, mixed corruption also reduces clean macro-F1 from .9594 to .9368, and both augmented variants have source–class recall drawbacks under some severe conditions. The results illustrate the competing objectives of clean-input accuracy, mean corruption tolerance, and worst-case class recall.

Discussion

The results reveal a condition-dependent accuracy–efficiency–robustness trade-off. On the held-out data, the two pretrained mean-pooled encoders attain similar group-weighted macro-F1 point estimates within the uncertainty of available independent source groups. ShuffleNetV2 reduces network and warm-source processing cost on the tested CPU, while ResNet18 loses less macro-F1 on average across the selected synthetic corruptions. Under severe Gaussian noise, the drone-class recall of both can deteriorate to unacceptable levels despite high clean performance. Thus inference speed and aggregate clean accuracy are inadequate stand-alone selection criteria.

The supplementary development experiments further clarify the role of temporal representation. Mean pooling can be competitive with LSTM or GRU variants at the tested resolution: the single-seed development scores of ResNet18-Mean and ResNet18-LSTM are .9756 and .9649, respectively, while Tiny CNN-Mean and Tiny CNN-GRU score .9207 and .9242. Nevertheless, R(2 + 1)D-18 outperforms the tested ShuffleNetV2 configuration on a common development subset, showing that explicit spatiotemporal modeling may still help when its much greater computation is acceptable. The different pretraining, input sizes, and development protocol rule out causal claims that a particular operation alone produces these differences.

These findings should be interpreted in light of the evaluation setting. Because both data sources contribute to training, the held-out sets quantify performance on new source groups within known datasets; generalization to new acquisition environments remains to be examined. The LAT test split contains nine conservative source groups, which limits the precision of group-bootstrap inference. CPU measurements characterize isolated ROI classification on a single-thread x86 processor, whereas integrated surveillance would additionally involve detection, tracking, and stream scheduling. The

cached-ROI and warm-source timing procedures also differ in their preprocessing and software environments, so latency values are compared within each protocol. Finally, the synthetic corruptions reveal sensitivity patterns that merit validation under representative sensor and environmental conditions.

Taken together, the findings suggest that efficient infrared video classification is best assessed through joint consideration of compute cost, input quality, and class-specific error patterns. A promising next step is to examine whether knowledge from computationally intensive spatiotemporal encoders can improve lightweight representations, followed by evaluation with continuous video and integrated target localization.

Conclusion

This study compares infrared bird–drone ROI video classifiers under explicit CPU and image-degradation constraints. Held-out three-seed group-weighted macro-F1 scores are 0.9638 for ResNet18-Mean and 0.9727 for ShuffleNetV2-Mean; the paired difference does not show a statistically supported accuracy ranking. ShuffleNetV2 offers lower single-thread inference costs on the evaluated x86 configuration, whereas ResNet18 is less sensitive on average to the predefined corruptions. The supplementary sampling and spatiotemporal-encoder results further illustrate how classification gains depend on computational cost. For resource-limited infrared monitoring, encoder selection should therefore consider the processing budget, likely acquisition degradations, and the consequences of class-specific errors.

Declarations

Competing interests.

The authors declare that they have no competing interests.

AI-assisted manuscript preparation.

OpenAI ChatGPT (GPT-6; October 2026) was used to assist with translating the authors' Chinese manuscript into English, editorial restructuring, and language revision. AI-assisted tools also supported LaTeX formatting and the programmatic redrawing of figures from the pre-existing method description and reported numerical results. No new experimental measurements were generated as part of manuscript preparation. The authors remain responsible for verifying the accuracy and integrity of the final submitted manuscript.

Funding.

No funding was received for this research.

Author Contribution

□□□□Yucheng Wang□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Yu Wang□ □□□□□□□□□□□□□□□□

Data Availability

The datasets analyzed in this study were obtained from publicly available research datasets, including LAT-BirdDrone and the infrared subset of the multi-sensor drone detection dataset. The experimental results and supporting analysis materials generated during this study are available from the corresponding author upon reasonable request, subject to applicable dataset licensing restrictions.

References

1. Svanström, F., Alonso-Fernandez, F., Englund, C.: A dataset for multi-sensor drone detection. Data Brief. **39**, 107521 (2021). https://doi.org/10.1016/j.dib.2021.107521
2. Ke, M., Kang, X., Zhang, Z., et al.: LAT-BirdDrone: A dedicated dataset for high-precision classification of low-altitude small target trajectories enhanced by hybrid neural networks. Data Brief. **64**, 112333 (2026). https://doi.org/10.1016/j.dib.2025.112333
3. Akyon, F.C., Akagunduz, E., Altinuc, S.O., Temizel, A.: Sequence models for drone vs bird classification. In: Proceedings of SPIE, Sixteenth International Conference on Machine Vision, vol. 13072 (2023). https://doi.org/10.1117/12.3023457
4. Luesutthiviboon, S., de Croon, G.C.H.E., Altena, A., et al.: Improving visual differentiation of drones and birds in aerial surveillance using trajectory features. Neural Comput. Appl. **38**, 477 (2026). https://doi.org/10.1007/s00521-026-12080-5
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
6. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In: ECCV, pp. 116–131 (2018). https://doi.org/10.1007/978-3-030-01264-9_8
7. Howard, A., Sandler, M., Chu, G., et al.: Searching for MobileNetV3. In: ICCV, pp. 1314–1324 (2019). https://doi.org/10.1109/ICCV.2019.00140
8. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: ICLR (2019). https://openreview.net/forum?id=HJz6tiCqYm
9. Hendrycks, D., Mu, N., Cubuk, E.D., et al.: AugMix: A simple data processing method to improve robustness and uncertainty. In: ICLR (2020). https://openreview.net/forum?id=S1gmrxHFvB
10. Tran, D., Wang, H., Torresani, L., et al.: A closer look at spatiotemporal convolutions for action recognition. In: CVPR, pp. 6450–6459 (2018). https://doi.org/10.1109/CVPR.2018.00675

11. Ultralytics: YOLO26 model and classification documentation. https://docs.ultralytics.com/models/yolo26/; https://docs.ultralytics.com/tasks/classify/ (accessed 17 Oct 2026)
12. Kataria, M., Lall, B.: Tracking aided drone bird classification using YOLO and LSTM. In: ICIPCW (2023). https://doi.org/10.1109/ICIPC59416.2023.10328340
13. Zhu, J., Rong, J., Kou, W., et al.: Accurate recognition of UAVs on multi-scenario perception with YOLOv9-CAG. Sci. Rep. **15**, 27755 (2025). https://doi.org/10.1038/s41598-025-12670-8
14. Zhu, J., Rong, J., Kou, W., et al.: Robust real-time recognition of drones and birds in complex scenarios: A multimodal data fusion recognize approach. Complex. Intell. Syst. **12**, 8 (2026). https://doi.org/10.1007/s40747-025-02117-y
15. Nesteruk, A., Nikitin, V., Albrekht, Y., et al.: Distinguishing a drone from birds based on trajectory movement and deep learning. Sensors. **26**(3), 755 (2026). https://doi.org/10.3390/s26030755

Figures

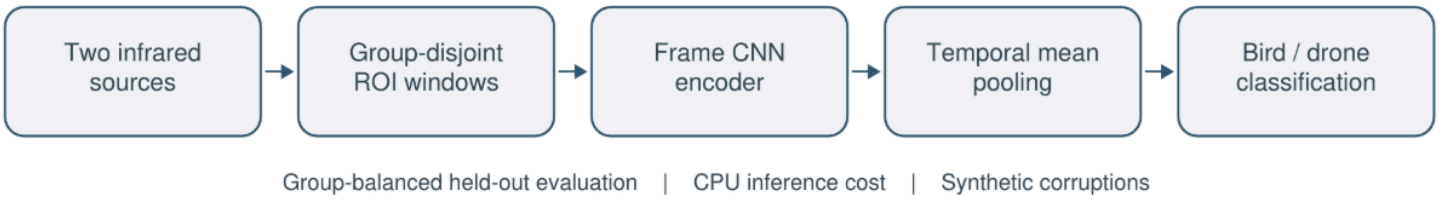


Figure 1

Legend not included with this version.

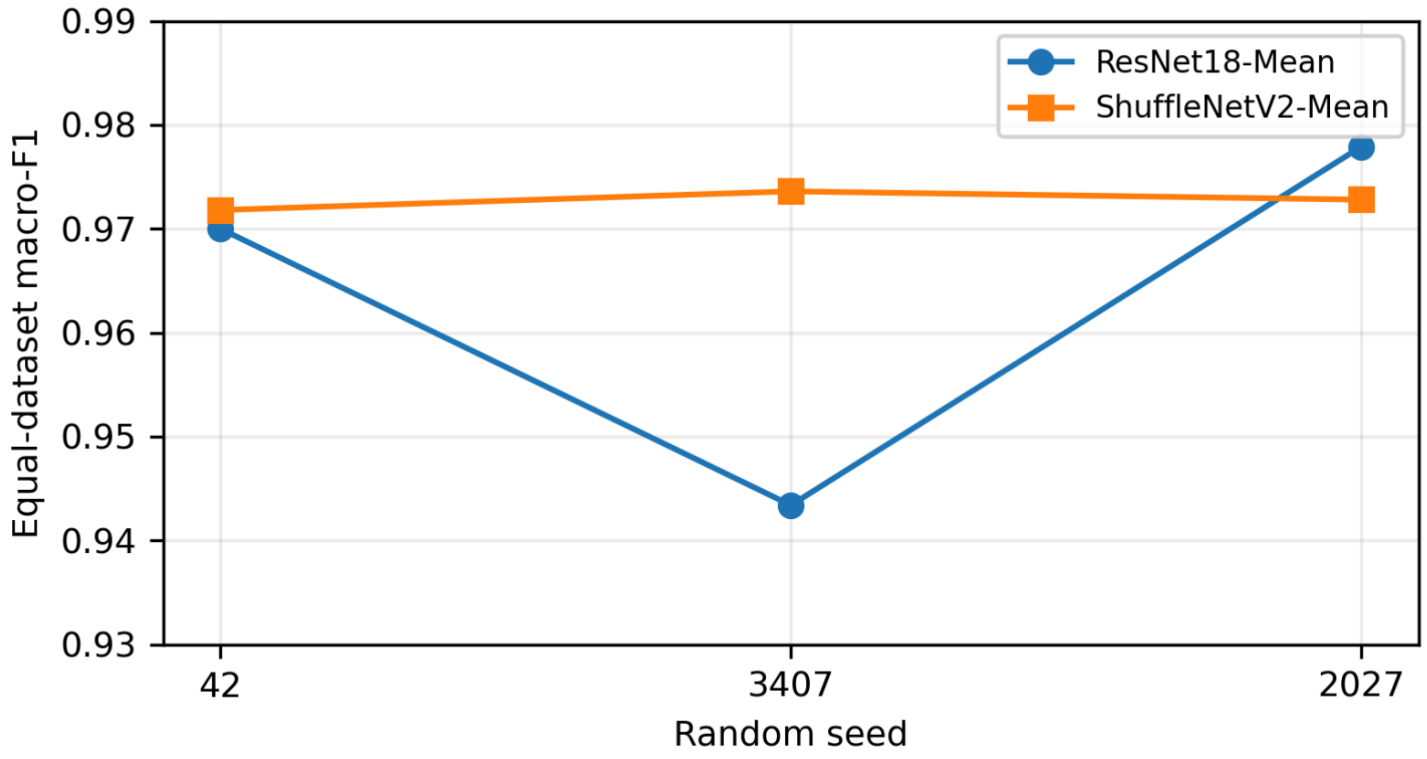


Figure 2

Legend not included with this version.

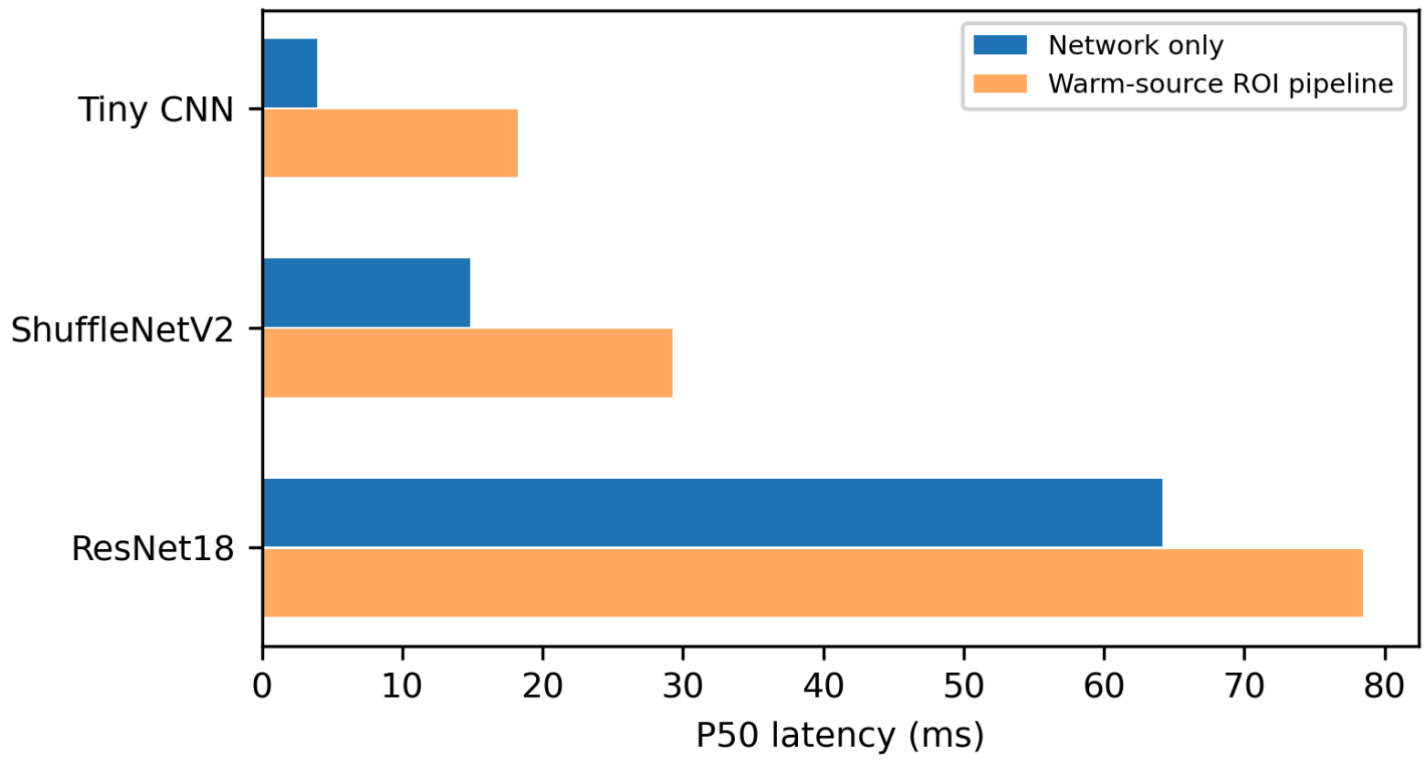


Figure 3

Legend not included with this version.

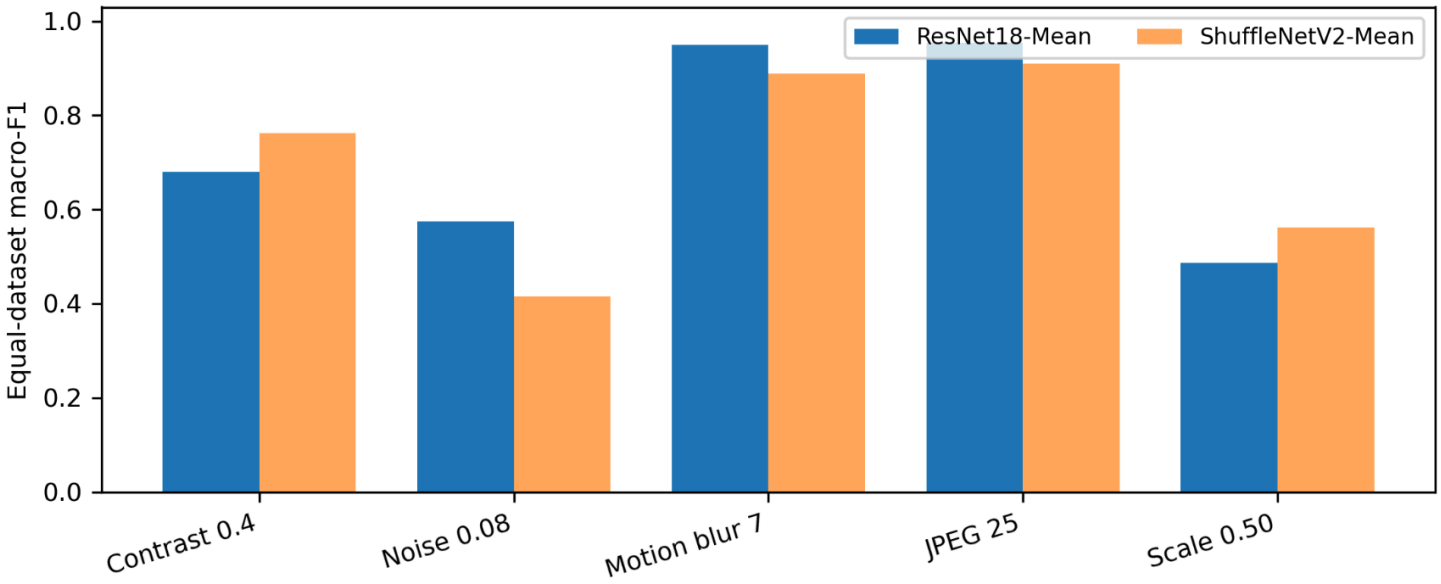


Figure 4

Legend not included with this version.

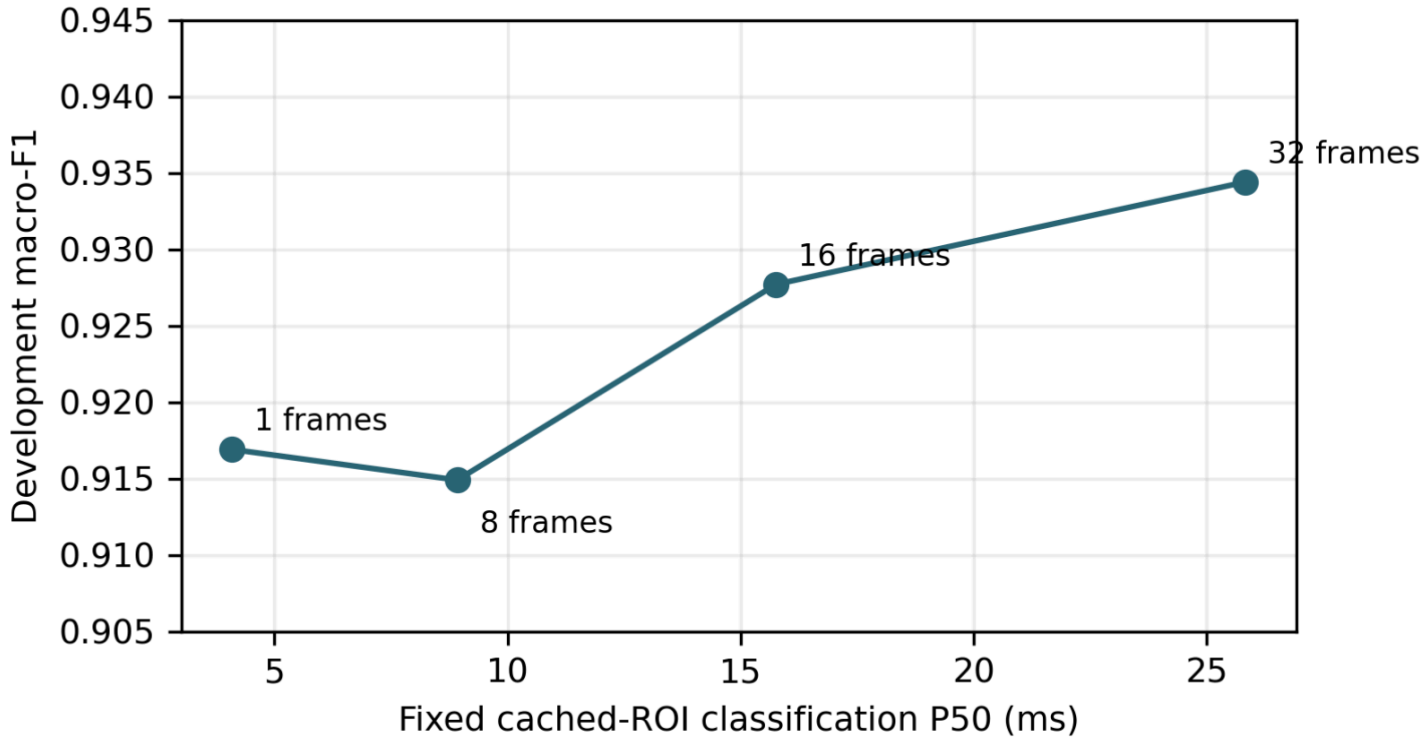


Figure 5

Legend not included with this version.