MoCrop: Training Free Motion Guided Cropping for Efficient Video Action Recognition

Binhua Huang[1], Wendong Yao[1], Shaowu Chen[2], Guoxin Wang[3], Qingyuan Wang[3], Soumyabrata Dev[1,4]
[1]School of Computer Science, University College Dublin
[2]Undergraduate School of Artificial Intelligence, Shenzhen Polytechnic University
[3]School of Electrical and Electronic Engineering, University College Dublin
[4]School of Computer Science and Statistics, Trinity College Dublin

***Abstract*—Standard video action recognition models often process typically resized full frames, suffering from spatial redundancy and high computational costs. To address this, we introduce MoCrop, a motion-aware adaptive cropping module designed for efficient video action recognition in the compressed domain. Leveraging Motion Vectors (MVs) naturally available in H.264 video, MoCrop localizes motion-dense regions to produce adaptive crops at inference without requiring any training or parameter updates. Our lightweight pipeline synergizes three key components: Merge & Denoise (MD) for outlier filtering, Monte Carlo Sampling (MCS) for efficient importance sampling, and Motion Grid Search (MGS) for optimal region localization. This design allows MoCrop to serve as a versatile "plug-and-play" module for diverse backbones. Extensive experiments on UCF101 demonstrate that MoCrop serves as both an accelerator and an enhancer. With ResNet-50, it achieves a +3.5% boost in Top-1 accuracy at equivalent FLOPs (Attention Setting), or a +2.4% accuracy gain with 26.5% fewer FLOPs (Efficiency Setting). When applied to CoViAR, it improves accuracy to 89.2% or reduces computation by roughly 27% (from 11.6 to 8.5 GFLOPs). Consistent gains across MobileNet-V3, EfficientNet-B1, and Swin-B confirm its strong generality and suitability for real-time deployment. Our code and models are available at https://github.com/microa/MoCrop.**

***Index Terms*—Motion-Aware Cropping, Video Action Recognition, Motion Vectors, Computational Efficiency, Plug-and-Play Module**

I. Introduction

Deep learning has fundamentally transformed video action recognition, evolving from hand-crafted features to sophisticated spatiotemporal representations learned via 3D convolutions and Video Transformers [1]–[4]. However, this performance leap comes with substantial computational costs. A primary source of inefficiency is spatial redundancy: standard models process the entire frame—often resized to a fixed resolution (e.g., 224×224)—regardless of whether the action occurs in a localized region or spans the whole scene. While spatial attention mechanisms [5], [6] can mitigate this by highlighting informative areas, they typically require end-to-end training, introduce additional parameters, and are coupled tightly with specific backbone architectures.

To address efficiency without sacrificing accuracy, the compressed domain offers a compelling alternative [7], [8]. Video codecs like H.264 naturally encode motion information via Motion Vectors (MVs) to reduce temporal redundancy. Prior works such as CoViAR [7] and MV2Flow [9] treat these MVs as input modalities to learn features. In contrast, we hypothesize that MVs serve a more primal purpose: they are essentially free saliency maps that can guide the model to focus on Spatially Salient Regions (SSR) before inference begins [10], [11].

Building on this insight, we introduce MoCrop, a lightweight, training-free module that leverages codec MVs for adaptive spatial cropping. Unlike learned attention modules, MoCrop operates as a pre-processor. It addresses the inherent noisiness and sparsity of raw MVs through a novel pipeline: Merge & Denoise (MD) to filter outliers, Monte Carlo Sampling (MCS) to drastically reduce processing overhead, and Motion Grid Search (MGS) to lock onto the optimal action region.

MoCrop offers a versatile plug-and-play capability. In an Efficiency Setting, it allows models to process smaller, cropped resolutions (e.g., 192px), significantly reducing FLOPs. In an Attention Setting, it removes background clutter for standard inputs, boosting accuracy. Crucially, MoCrop requires no retraining of the backbone model, making it immediately applicable to diverse architectures ranging from ResNet to Swin Transformers.

Our main contributions are summarized as follows:

- We propose MoCrop, a training-free, parameter-free adaptive cropping module that utilizes compressed-domain Motion Vectors to guide efficient video recognition.
- We design a high-efficiency pipeline incorporating Monte Carlo Sampling and dual-objective grid search, enabling robust region selection with negligible computational overhead.
- We demonstrate that MoCrop serves as a universal accelerator and enhancer. Experiments on UCF101 show it can improve Top-1 accuracy by up to 3.5% (ResNet-50) or reduce FLOPs by over 26% while maintaining accuracy, generalizing across CNNs and Transformers.

II. Related Work

Video Action Recognition. Early approaches to video action recognition relied on hand-crafted features but struggled with scalability [12]. The emergence of deep learning marked a

paradigm shift, with the Two-Stream Network [13] leveraging pre-computed optical flow to capture motion patterns. However, optical flow computation remains expensive. C3D [2] introduced 3D convolutions to learn spatiotemporal features end-to-end, avoiding explicit optical flow but incurring high computational costs. More recently, video transformers such as Video Swin [3] and TimeSformer [4] achieve state-of-the-art performance by modeling long-range dependencies through self-attention, though at significant computational expense. These methods motivate efficient alternatives that reduce computation while maintaining accuracy.

Compressed-Domain Video Understanding. Compressed video formats like H.264 encode videos using I-frames (independently coded), P-frames (predicted from previous frames), and B-frames (bi-directionally predicted), with MVs and residuals encoding inter-frame differences [8]. Leveraging these codec-provided features offers computational advantages over processing raw pixels. CoViAR [7] pioneered compressed-domain action recognition by processing I-frames, MVs, and residuals in separate streams. MV2Flow [9] refines MVs to approximate optical flow, bridging efficiency and accuracy. Recent works [11], [14] further exploit MVs to guide temporal frame selection and spatial attention, demonstrating that compressed-domain cues can enhance both efficiency and recognition performance.

Spatial Attention and Adaptive Cropping. Spatial attention mechanisms identify and emphasize informative regions within frames. Early convolutional approaches like CBAM [6] and SE-Net [15] apply channel and spatial attention modules trained end-to-end. Transformer-based methods [4], [16] use self-attention to model spatial dependencies but require substantial computation and parameter overhead. In the context of video understanding, adaptive cropping has been explored to focus on action-relevant regions. Traditional methods include center cropping, random cropping during training, and saliency-based ROI extraction [5]. Recent work by Shahabinejad et al. [11] integrates MVs into learned spatial attention for compressed video, achieving improved efficiency. However, most attention and cropping methods require training or fine-tuning for each backbone and dataset.

Our Approach. Unlike prior attention mechanisms that are learned end-to-end or require retraining, MoCrop is a training-free, plug-and-play module that operates solely at inference time. By aggregating MVs into a motion-density map and performing a lightweight spatial search, MoCrop adaptively crops to motion-salient regions without modifying model parameters or training protocols. This design makes it compatible with diverse backbones and datasets, offering a practical path to efficiency and accuracy improvements in compressed-domain video action recognition.

III. Method

MoCrop is a preprocessing module that uses MVs from compressed streams to locate SSR and compute adaptive crops, focusing inference on informative regions. This reduces input resolution and FLOPs and often improves accuracy by suppressing background distraction.

A. Overview

Fig. 1 illustrates the complete MoCrop pipeline. MoCrop comprises three processing modules: **Merge & Denoise (MD)**, **Monte Carlo Sampling (MCS)**, and **Motion Grid Search (MGS)**. MD retains top-k MVs by magnitude; MCS applies weighted importance sampling to downsample filtered MVs and builds a motion-density map; MGS searches for a high-score rectangle via dual-objective optimization to determine the cropping region.

B. Problem Formulation

Given decoded I-frames and raw MVs, let $V_{\text{raw}} = \{(p_k, \Delta p_k)\}_{k=1}^{T}$ with $p_k = (x_k, y_k)$ the origin and Δp_k the displacement. We seek a function f that maps MVs to a bounding box B^*:

$$B^* = f(V_{\text{raw}}),$$

and each I-frame I_i is then cropped by

$$I_i' = g(B^*, I_i).$$

C. Detailed Algorithm

We now describe each stage in detail.

1) Stage 1: Merge & Denoise (MD): **Extraction and Filtering.** Let $V_{\text{raw}} = \{(p_k, \Delta p_k)\}_{k=1}^{T}$ denote raw MVs across frames. We retain only the top-k vectors by motion magnitude:

$$V_{\text{filtered}} = \text{top-}k(V_{\text{raw}}, \alpha), \quad k = \lfloor \alpha |V_{\text{raw}}| \rfloor,$$

where $\alpha \in (0, 1]$ controls the denoising strength (e.g., α=0.01 keeps the top 1%).

2) Stage 2: Monte Carlo Sampling (MCS): For efficiency when $|V_{\text{filtered}}|$ is large, we apply weighted importance sampling. Each vector is sampled with probability proportional to its motion magnitude raised to power β:

$$p_i \propto \|\Delta p_i\|_2^{\beta},$$
$$V_{\text{sampled}} \sim \text{Multinomial}(V_{\text{filtered}}, N, \{p_i\}),$$

where N=$\lfloor \gamma |V_{\text{filtered}}| \rfloor$, $\gamma \in (0, 1]$ is the sampling ratio (e.g., γ=0.1 samples 10%), and β>0 (e.g., β=4.0) biases toward high-magnitude motion.

3) Stage 3: Motion Grid Search (MGS): **Grid Quantization.** We discretize the frame into an $h \times w$ grid and map each MV to its corresponding cell:

$$M_{ij} = \sum_{(p_k, \cdot) \in V_{\text{sampled}}} \mathbb{I}(p_k \in C_{ij}),$$

where C_{ij} spans $[j\frac{W}{w}, (j+1)\frac{W}{w}) \times [i\frac{H}{h}, (i+1)\frac{H}{h})$. This motion-density map M guides the subsequent search.

Submatrix Search. We identify the grid region R with maximum weighted score:

$$R^* = \arg\max_{R \in \mathcal{R}_\rho} S(R),$$
$$S(R) = w_{\text{sum}} \cdot \sum_{(i,j) \in R} M_{ij} + w_{\text{avg}} \cdot \frac{1}{|R|} \sum_{(i,j) \in R} M_{ij},$$

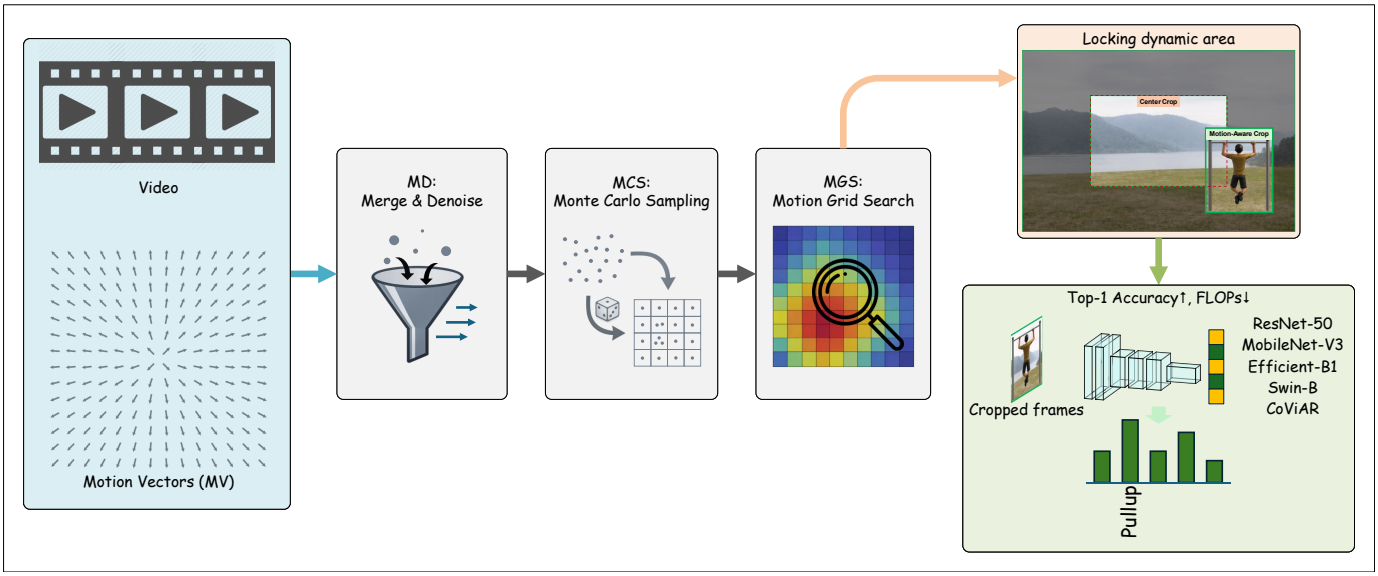

Fig. 1: MoCrop pipeline. MVs identify SSR to guide I-frame cropping; cropped frames are fed to the action recognition model.

where $\mathcal{R}_\rho = \{R : (1-\delta)\rho hw \leq |R| \leq (1+\delta)\rho hw\}$ constrains candidate rectangles to target area ratio ρ with tolerance δ, and $(w_{\text{sum}}, w_{\text{avg}})$ balance total vs. average density (e.g., $(0.6, 0.4)$). The optimal region R^* is converted to normalized box coordinates for cropping.

4) Stage 4: Cropping I-frames: We map the grid region R^* to pixel coordinates and extract the corresponding crop:

$$I_{\text{crop}} = I[y_1 : y_2,\ x_1 : x_2],$$
$$(x_1, y_1, x_2, y_2) = \text{Grid2Pixel}(R^*, W_{\text{video}}, H_{\text{video}}).$$

The same box is applied to all I-frames in the clip, yielding an efficient, training-free, plug-and-play preprocessor.

D. Complexity and Practicality

MD performs top-k selection via `argpartition` in $O(|V_{\text{raw}}|)$ time. MCS computes magnitude weights and samples in $O(|V_{\text{filtered}}|)$. Building the motion-density map costs $O(|V_{\text{sampled}}| + hw)$ for counting on an $h \times w$ grid. MGS enumerates all rectangles within the area constraint: with integral images, scoring each candidate is $O(1)$, and the total search is $O(h^2 w^2)$ but negligible for small grids (e.g., 16×9). Overall preprocessing overhead is negligible, making MoCrop practical for real-time deployment.

Hyperparameters. Key parameters include: α (denoising ratio, e.g., 0.01) controls noise filtering strength; γ (sampling ratio, e.g., 0.1) trades variance for speed; β (importance weight, e.g., 4.0) biases sampling toward high-motion regions; (h, w) (grid resolution, e.g., 16×9) balances localization precision and search cost; ρ (target area ratio) sets crop size; $(w_{\text{sum}}, w_{\text{avg}})$ (e.g., 0.6, 0.4) balance region compactness vs. density.

IV. Experiments

A. Dataset and Metrics

All experiments are conducted on UCF101, a widely used benchmark for human action recognition featuring 13,320 videos across 101 action classes [17]. Given that our approach is designed for efficiency, we evaluate performance by jointly considering Top-1 accuracy and computational cost in GFLOPs.

B. Experimental Setup

We evaluate MoCrop on the official UCF101 split. Following standard practice [7], we sample three frames per video for training and use typical data augmentation. At inference, we freeze the model weights and evaluate two main settings: **Baseline:** The model trained on UCF101 is evaluated on the standard test set with a 224px input. **MoCrop:** The same model is evaluated after each video is pre-processed by MoCrop. We test two variants:

- **Efficiency Setting:** MoCrop crops the video, which is then resized to a smaller resolution (192px) to reduce FLOPs while aiming to maintain or improve accuracy.
- **Attention Setting:** MoCrop crops the video, which is resized back to the original resolution (224px). This isolates the accuracy improvement gained by focusing on salient regions, without changing the computational cost.

C. Performance and Efficiency Analysis

Table I demonstrates the powerful dual advantages of MoCrop as a plug-and-play module for various modern backbones. In the Attention Setting, where FLOPs are held constant, MoCrop consistently boosts performance by forcing the model to concentrate on motion-rich areas. The most significant gain is seen with ResNet-50 [18], which improves by a remarkable +3.5% Top-1 accuracy. Other models like EfficientNet-B1 [20] (+2.4%) and MobileNet-V3 [19] (+1.9%) also show substantial improvements.

In the Efficiency Setting, MoCrop proves its ability to reduce computational costs and often enhances accuracy. For instance, ResNet-50 sees a +2.4% accuracy increase while

TABLE I: Demonstration that MoCrop provides dual advantages as a plug-and-play module on modern backbones using the UCF101 dataset. The table highlights the performance gains in both the accuracy-focused **Attention Setting** and the cost-saving **Efficiency Setting**.

Backbone	Dataset	Variant	Top-1 Acc. (%)	GFLOPs	Δ Acc. (%)	Δ FLOPs (%)
ResNet-50 [18]	UCF101	(A) Baseline (224px)	80.1	4.11	-	-
		(B) + MoCrop (Efficiency, 192px)	82.5	**3.02**	+2.4	**-26.5**
		(C) + MoCrop (Attention, 224px)	**83.6**	4.11	**+3.5**	0.0
MobileNet-V3 [19]	UCF101	(A) Baseline (224px)	78.3	0.22	-	-
		(B) + MoCrop (Efficiency, 192px)	79.3	**0.17**	+1.0	**-22.7**
		(C) + MoCrop (Attention, 224px)	**80.2**	0.22	**+1.9**	0.0
EfficientNet-B1 [20]	UCF101	(A) Baseline (224px)	82.1	0.59	-	-
		(B) + MoCrop (Efficiency, 192px)	83.6	**0.43**	+1.5	**-27.1**
		(C) + MoCrop (Attention, 224px)	**84.5**	0.59	**+2.4**	0.0
Swin-B [3]	UCF101	(A) Baseline (224px)	87.3	15.5	-	-
		(B) + MoCrop (Efficiency, 192px)	87.2	**12.6**	-0.1	**-18.7**
		(C) + MoCrop (Attention, 224px)	**87.5**	15.5	**+0.2**	0.0

FLOPs are cut by a massive 26.5%. Even for the heavy Swin-B Transformer [3], where accuracy gains are harder to achieve, our method reduces the computational load by 18.7% with only a negligible 0.1% accuracy dip. These results confirm that MoCrop offers an exceptional and highly flexible trade-off between accuracy and efficiency.

D. Ablation Studies

To validate the contributions of the internal components of MoCrop and its effectiveness as a general module, we present a series of ablation studies in Tables II and III. The studies are organized into three parts: (a) an incremental analysis of our module's components, (b) a comparison against standard cropping methods, and (c) an application of MoCrop to enhance a prior state-of-the-art method.

Incremental Component Analysis. Using ResNet-18 [18] as a base, we find that each component of MoCrop contributes to the final performance (Table II). MGS is essential as the pipeline cannot run without it. Further incorporating MD or MCS individually yields substantial improvements, and the full pipeline combining all three components (MGS + MD + MCS) achieves the best results, confirming that all parts of our design are integral to its success. Importantly, the pre-processing cost column reveals the remarkable computational efficiency: even MGS alone incurs only 0.121 MOps, and adding MD or MCS reduces this to just 0.022 and 0.031 MOps respectively. The full pipeline achieves the lowest cost at 0.021 MOps—over 70,000× smaller than the model inference cost of 1.54 GFLOPs (1540 MOps)—demonstrating truly negligible overhead.

Comparison with Cropping Strategies. MoCrop significantly outperforms standard cropping techniques across multiple crop ratios (Table III). At 90% crop ratio, both Random and Center crops offer slight improvements over the baseline (80.4% for both). However, as the crop ratio decreases to 50%, these fixed strategies show severe degradation (73.5% and 71.4%). In contrast, our adaptive method achieves 81.4% accuracy with an adaptive crop ratio, demonstrating superior motion-guided region selection over static approaches across all settings.

TABLE II: Component ablation study on ResNet-18. The Overhead column shows the negligible computational cost of MGS preprocessing (place_blocks + search_box) in Million Operations (MOps), compared to model inference at 1.54 GFLOPs (1540 MOps).

Setup	MGS	MD	MCS	GFLOPs	Overhead (MOps)
Baseline (No MoCrop)	–	–	–	1.82	0
w/o MGS (Pipeline cannot run)	–	–	–	–	–
+ MGS	✓	–	–	1.54	0.121
+ MGS + MD	✓	✓	–	1.54	0.022
+ MGS + MCS	✓	–	✓	1.54	0.031
+ MGS + MD + MCS (Full)	✓	✓	✓	**1.54**	**0.021**

TABLE III: Performance comparisons on ResNet-18 and CoViAR. (a) Comparison with standard cropping strategies. (b) Enhancing CoViAR with MoCrop on UCF101 (I-frames only).

Method	Acc. (%)	GFLOPs
(a) Cropping Strategies on ResNet-18		
Baseline	79.3	1.82
Random Crop (90%)	80.4	1.73
Center Crop (90%)	80.2	1.73
MoCrop (Adaptive)	**81.4**	**1.54**
(b) Enhancing CoViAR (UCF101, I-frames only)		
CoViAR Baseline (Reported in [7])	87.7	11.6
+ MoCrop (Efficiency, 192px)	88.5	**8.5**
+ MoCrop (Attention, 224px)	**89.2**	11.6

Enhancing Prior Works. To show its versatility, we applied MoCrop to CoViAR, a well-known compressed-domain method (Table III). The results are compelling. In the efficiency setting, MoCrop improves CoViAR's accuracy to 88.5% while reducing its FLOPs from 11.6 GFLOPs to just 8.5 GFLOPs. In the attention setting, it boosts the accuracy to

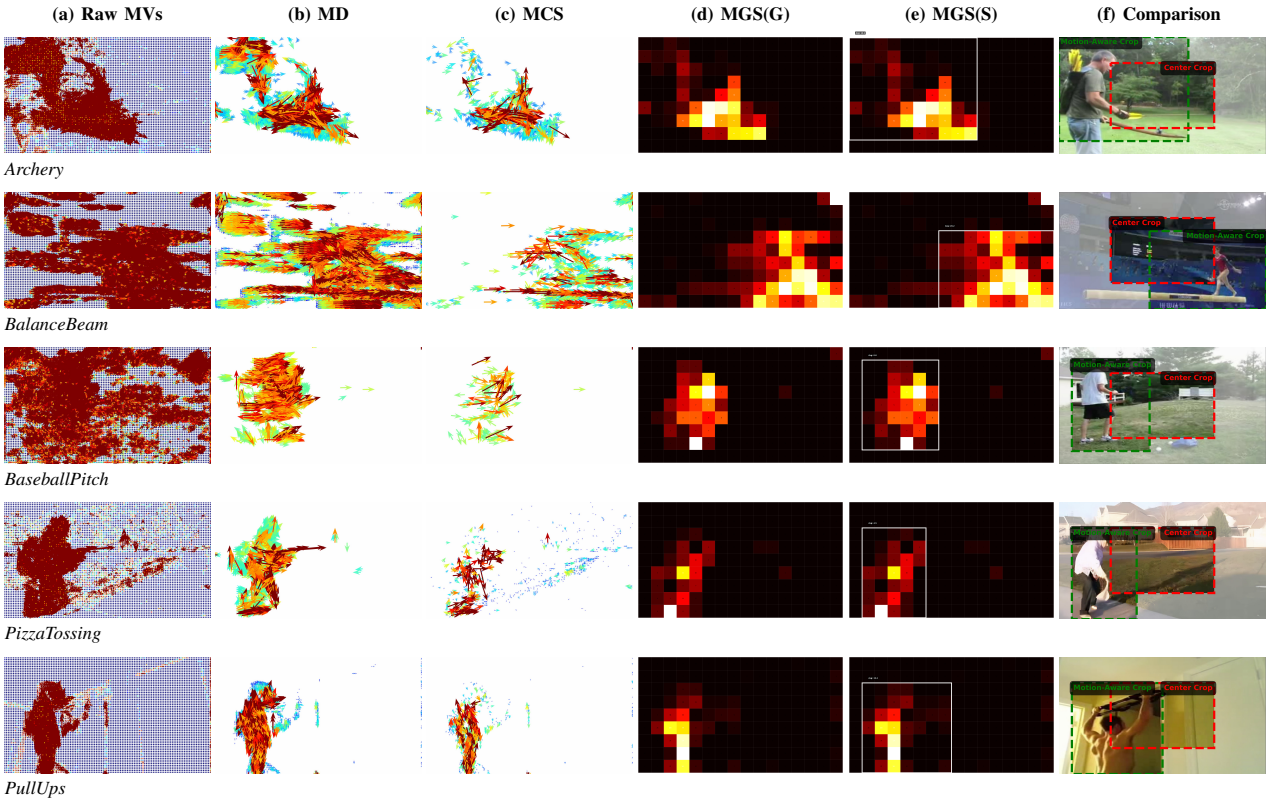

Fig. 2: Qualitative visualization of the complete MoCrop pipeline across six representative UCF101 action classes. Each row shows one video processed through six stages: **(a) Raw MVs**: MVs extracted from H.264 compressed video, visualized as arrows showing magnitude and direction. **(b) MD (Denoised)**: Top-1% MVs retained after percentile-based filtering. **(c) MCS (Sampled)**: MVs after weighted importance sampling (10% of denoised MVs, biased toward high-magnitude motion). **(d) MGS - Grid**: Motion-density heatmap aggregated on 16×9 spatial grid. **(e) MGS - Search**: Optimal region identified by dual-objective scoring (white box overlaid on heatmap). **(f) Comparison**: Overlay comparison of Motion-Aware Crop (green box) vs. Center Crop (red box) on the same RGB frame, demonstrating MoCrop's ability to focus on action-relevant regions.

an impressive 89.2%. This shows that MoCrop is not just a standalone module but can also serve as a powerful enhancer for other methods in the field.

E. Qualitative Analysis

Fig. 2 visualizes the complete MoCrop pipeline across four representative UCF101 action classes. Each row demonstrates how raw MVs are progressively refined through MD (percentile filtering), MCS (importance sampling), and MGS (grid aggregation and region search) to produce focused crops. The final comparison (column f) shows that MoCrop consistently identifies tighter, action-centric regions compared to fixed center crops, removing irrelevant background while preserving the actor.

V. Discussion and Conclusion

We present MoCrop, a lightweight, training-free module using compressed-domain MVs to identify motion-dense regions for adaptive cropping. By focusing inference, MoCrop reduces resolution and FLOPs, often boosting accuracy, and integrates with existing backbones without retraining.

When it helps. MoCrop is most effective when actions coincide with localized motion and when background clutter dilutes fixed crops. The map-based formulation is robust to moderate noise after filtering and sampling, and it generalizes across backbones since it operates as a preprocessing step.

Limitations. Fig. 3 illustrates five failure modes where MoCrop's motion-density approach encounters challenges. MoCrop relies on codec-provided MVs (e.g., H.264) and is less informative for raw videos unless motion estimation is run in parallel. This dependency is acceptable since most datasets and streaming pipelines use compressed video.

Specifically, the following scenarios can bias the density map: **(1) Camera shake** (*Archery*): When the cameraman shakes the lens, spurious high-magnitude MVs appear (e.g., bright regions in corners), dominating the density map and drawing the crop away from the actual actor. **(2) Perspective effects** (*BandMarching*): When subjects move uniformly across the frame, perspective makes closer objects move faster. Since our pipeline rescales MVs by clip-wise median magnitude for robustness, this causes the near-camera region to dominate, yielding incorrect crops. **(3) Stationary actors**

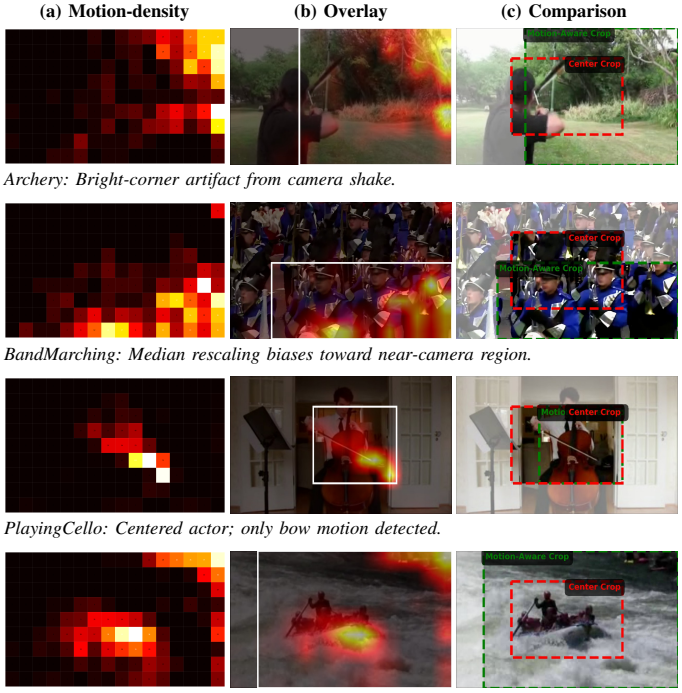

Fig. 3: Failure mode analysis across five representative cases. (a) Motion-density map. (b) Heatmap overlay with search result (white box). (c) Motion-Aware Crop (green) vs. Center Crop (red). See text for detailed analysis.

(*PlayingCello*): When actors are already centered with minimal body translation, only small regions (e.g., the bow) exhibit motion. The density map localizes these regions but fails to cover the full actor, offering little advantage over center cropping. **(4) Scene-wide motion** (*Rafting*): Turbulent water or dynamic backgrounds generate MVs across the entire frame, producing nearly uniform density maps. MGS then selects regions close to the full frame, providing no effective spatial focus.

Mitigations include global-motion suppression, temporal smoothing, and combining MV magnitude with residual cues. These failure modes occur in a minority of UCF101 action classes and primarily arise when motion does not reliably indicate actor location.

Future work. We plan to extend MoCrop to diagonally elongated or multi-region actions via multi-box proposals, explore codec-agnostic motion signals, and validate on larger datasets and backbones. We will also study learned selection over MoCrop proposals while keeping inference cost low.

In summary, MoCrop offers a practical plug-and-play path to efficiency and accuracy in compressed-domain video understanding, requiring no additional parameters and minimal compute overhead.

REFERENCES

[1] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[2] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[3] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu, "Video swin transformer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3202–3211.

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani, "Is space-time attention all you need for video understanding?," in *ICML*, 2021, vol. 2, p. 4.

[5] Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Song-Hai Zhang, Ralph R Martin, Ming-Ming Cheng, and Shi-Min Hu, "Attention mechanisms in computer vision: A survey," *Computational visual media*, vol. 8, no. 3, pp. 331–368, 2022.

[6] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[7] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl, "Compressed video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6026–6035.

[8] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra, "Overview of the h. 264/avc video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[9] Hezhen Hu, Wengang Zhou, Xingze Li, Ning Yan, and Houqiang Li, "Mv2flow: Learning motion representation for fast compressed video action recognition," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 3s, pp. 1–19, 2020.

[10] Ziwei Zheng, Le Yang, Yulin Wang, Miao Zhang, Lijun He, Gao Huang, and Fan Li, "Dynamic spatial focus for efficient compressed video action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 2, pp. 695–708, 2023.

[11] Mostafa Shahabinejad, Irina Kezele, Seyed Shahabeddin Nabavi, Wentao Liu, Seel Patel, Yuanhao Yu, Yang Wang, and Jin Tang, "Video action recognition with adaptive zooming using motion residuals," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1214–1223.

[12] Chaolong Zhang, Yuanping Xu, Zhijie Xu, Jian Huang, and Jun Lu, "Hybrid handcrafted and learned feature framework for human action recognition," *Applied Intelligence*, vol. 52, no. 11, pp. 12771–12787, 2022.

[13] Karen Simonyan and Andrew Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in neural information processing systems*, vol. 27, 2014.

[14] Yue Ming, Jiangwan Zhou, Nannan Hu, Fan Feng, Panzi Zhao, Boyang Lyu, and Hui Yu, "Action recognition in compressed domains: A survey," *Neurocomputing*, p. 127389, 2024.

[15] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[16] Alexey Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[17] K Soomro, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al., "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[20] Mingxing Tan and Quoc Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.