Keypoint-Based Detection and Collective Behaviour Analysis of Honeybees Using YOLOv8n-pose

Valentin Daveau[1], Bastien Posez[1], Sebora Memolla[2], Patrice Guyot[2], Stefan Janaqi[2], and Baptiste Magnier[2]

[1]IMT Mines Ales, Ales, France
[2]EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Ales, Ales, France
{valentin.daveau, bastien.posez}@etu.mines-ales.fr
{sebora.memolla, patrice.guyot, stefan.janaqi, baptiste.magnier}@mines-ales.fr

Abstract. Understanding the collective behaviour of honeybees (*Apis mellifera*) inside a hive requires dense, accurate detection of individual bees and their orientations from camera imagery. In this paper, a complete pipeline is presented that fine-tunes YOLOv8n-pose to simultaneously localise bees via bounding boxes, classify them as cleaning or normal, and regress three anatomical keypoints (head, thorax, and tail) for each individual. For inference on high-resolution frames ($\geq 1500 \times 1200$ pixels), we introduce a sliding-window tiling scheme with non-maximum suppression across tile boundaries. To select the best preprocessing strategy, 15 pipelines combining two contrast-enhancement methods and five edge-preserving filters are evaluated against the original images as baseline on an 800-image hive dataset with 7,658 annotated bee instances. Each variant is trained and benchmarked independently under identical YOLOv8n-pose hyperparameters. We analyse the real-time viability of each pipeline: at 30 fps, preprocessing must be completed within approximately 18 ms. Only two pipelines meet the contrast-plus-filtering constraint, with a live-to-offline accuracy gap of 2.5 percentage points in Box mAP50 (77.0% versus 79.5% for the offline optimum). Notably, classical filters proposed decades ago, Symmetric Nearest Neighbour (SNN, 1983) and Kramer–Brückner (1975), remain highly competitive, ranking first and fourth among all 15 pipelines. Building on per-individual heading vectors, eight quantitative behavioural analyses are implemented. These include Rayleigh circular statistics, local alignment scoring, spatial flow field decomposition (divergence and curl), following chain detection, and trophallaxis candidate identification. Together, they transform raw detections into ecologically interpretable collective motion descriptors.

Keywords: hand-crafted preprocessing · edge-preserving filters · YOLOv8n-pose · honeybee detection · keypoint detection · collective behaviour

1 Introduction

Honeybee colonies exhibit rich collective behaviours, such as waggle dances conveying foraging directions, trophallaxis food-sharing chains, coordinated grooming cascades, and thermogenic clustering, that are encoded in the instantaneous positions and orientations of hundreds of individuals. Tracking these orientations in dense, partially-occluded hive imagery is a prerequisite for quantitative ethology at the colony scale. Classical approaches rely on hand-crafted markers [17,5] or highly constrained imaging conditions. Recent work has demonstrated the viability of marker-free deep tracking in dense hive imagery [6]. However, such systems generally estimate orientation from segmentation or shape cues rather than explicit anatomical keypoints, whereas keypoint-based pose estimation can directly encode body landmarks required for per-bee heading estimation.

This work addresses this gap with a unified single-stage system based on YOLOv8n-pose [16]. The detector produces bounding boxes, class labels (cleaning vs. normal), and a three-keypoint skeleton per detection in a single forward pass. The head–tail axis, derived from the head and tail keypoints, provides the heading vector that feeds subsequent collective-behaviour analyses. A key practical requirement for live hive monitoring is *real-time throughput*: the full frame-processing pipeline, preprocessing plus model inference, must complete within 33 ms (milliseconds) for 30 fps (frames per second) operation. Fifteen preprocessing pipelines are systematically benchmarked in terms of detection accuracy (test-set Box and Pose mAP50) and per-frame preprocessing latency, identifying the combinations that remain viable for live deployment.

The remainder of this paper is organised as follows. Section 2 describes the model architecture and the tiling scheme for high-resolution inference. Section 3 presents the 15-pipeline preprocessing study and analyses the live-deployment

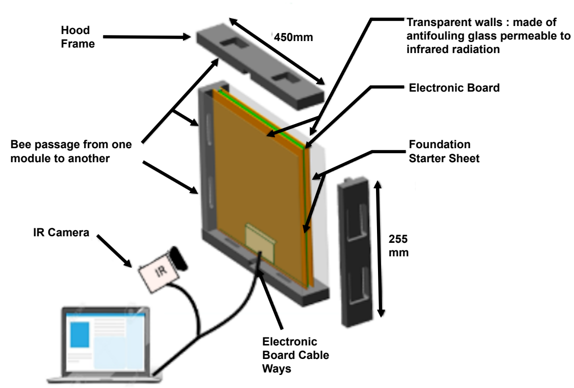


Fig. 1: Modular observation hive and IR camera setup. The transparent antifouling glass wall allows unobstructed infrared imaging, while the foundation starter sheet anchors natural comb construction.

speed-accuracy trade-off. Section 4 details the training protocol and augmentation. Section 5 reports quantitative results on the held-out test split, including a ranked comparison of all 15 pipelines and per-epoch validation curves. Section 6 formalises each of the eight behavioural analyses. Section 7 concludes with limitations and future directions.

Imaging Setup

The annotated dataset originates from a custom modular observation hive designed for non-invasive, marker-free monitoring [3]. Figure 1 illustrates the hardware assembly. Each module measures 450×255 mm and is enclosed by transparent, anti-fouling glass that is permeable to infrared radiation. An infrared (IR) camera mounted in front of the hive frame records continuous colony activity under controlled imaging conditions. An on-board PCB controls the illumination and integrates environmental sensing, ensuring stable lighting during acquisition. Frames from adjacent modules communicate through standard bee-passage apertures. The resulting IR footage is the primary input to the detection and keypoint pipeline described in the following sections.

2 Model Architecture

2.1 YOLOv8n-pose: nano variant of YOLOv8 (You Only Look Once)

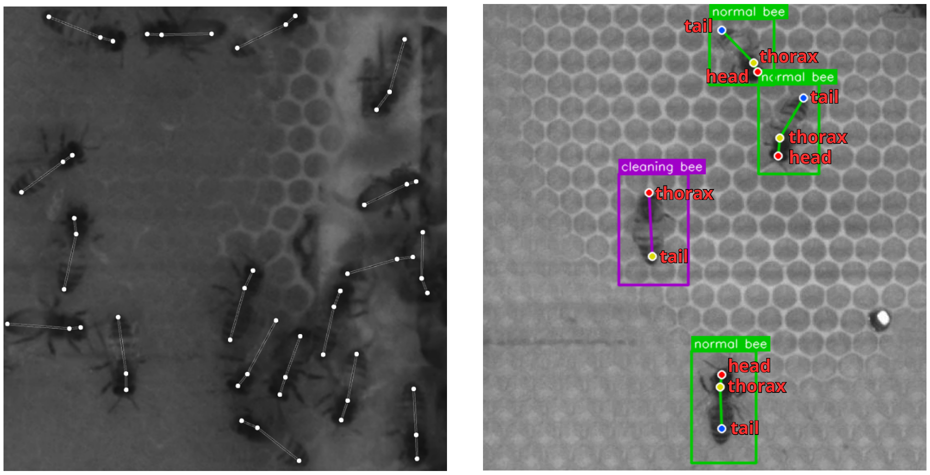
This work uses YOLOv8n-pose [16], the nano variant of Ultralytics YOLOv8 extended with keypoint regression heads. Initialised from COCO-pretrained weights [1], the model predicts bounding boxes, class scores, and K=3 keypoints per detected instance in a single forward pass. These three keypoints are:

1. **kp 0** — head (anterior terminus of the bee),
2. **kp 1** — thorax (dorsal wing root),
3. **kp 2** — tail (posterior terminus / tip of abdomen).

Figure 2 illustrates the three-keypoint annotation format used for YOLOv8n-pose training, shown on two different frames. The first image shows the annotated keypoints for all visible bees in a frame, from the annotation interface[1], providing a representative view of the labelling process. The second image presents a custom rendering on a 640×640 frame, showing several normal bees and one cleaning bee, each with its corresponding bounding box and labelled head–thorax–tail keypoints.

The nano backbone (≈3.3 M parameters) fits entirely in Apple Silicon unified memory, enabling MPS-accelerated (Metal Performance Shaders) training without out-of-memory errors.

[1] `https://roboflow.com/` is a computer vision development platform used for image annotation, model training, and deployment.



(a) Keypoint annotations on bees (b) Normal & cleaning bees with keypoints

Fig. 2: Example of the keypoint annotation format used for YOLOv8n-pose training. Each bee instance is annotated with a bounding box and three anatomical keypoints: head, thorax and tail. The head–tail keypoint distance (kp 0–kp 2) spans ≈74 px, giving ≈0.16–0.20 mm/px for a 12–15 mm worker [15].

2.2 Sliding-Window Tiling for High-Resolution Frames

Field recordings often exceed 1500×1200 px (pixels). Passing such frames directly to a model calibrated for 448×448 crops causes catastrophic down-scaling that erases bee detail. We therefore tile large frames into square windows of 640×640 px (side $T = 640$ px) with stride $S = T - O$, where $O = 320$ px is the overlap, ensuring every point in the image falls inside at least one tile. The stride is applied identically on both axes; the full 2-D tiling grid is the Cartesian product of the same 1-D sliding scheme repeated horizontally and vertically:

$$\mathrm{starts}(D, T, S) = \{0,\ S,\ 2S,\ \ldots\} \cup \{\max(0,\ D - T)\}, \quad S = T - O. \tag{1}$$

For each tile, the keypoints and bounding boxes are shifted to the original coordinate frame by adding the tile offset (x_0, y_0). A centre-bias filter retains each detection, only if its bounding-box centre falls inside a 48 px-inset safe zone of the tile, preventing double-counting at seam boundaries. After all tiles are processed, residual duplicates are suppressed with non-maximum suppression (NMS) at IoU threshold $\tau_{\mathrm{NMS}} = 0.30$:

$$\mathcal{D}^* = \mathrm{NMS}\left(\bigcup_{\text{tiles}} \mathcal{D}_t,\ \tau_{\mathrm{NMS}}\right). \tag{2}$$

A 1500×1200 frame thus produces 15 tiles, as shown in Table 1. The same tiling strategy scales to arbitrarily large images while keeping the per-tile memory cost

Table 1: Inference tile count vs. input resolution (T=640 px, O=320 px).

Resolution	Number of tiles	Regime
512×512	1	single pass
1280×720	6	tiled
1500×1200	15	tiled
2000×2000	16	tiled

constant, though frames are processed independently with no temporal linking across detections (see Section 7).

3 Image Preprocessing Pipelines

Raw IR (infrared) hive footage is characterised by low global contrast, heterogeneous illumination across the comb surface, and sensor noise that can obscure bee boundaries. To assess the impact of preprocessing on downstream keypoint accuracy, we design a two-stage pipeline: a *contrast enhancement* step followed by an *edge-preserving denoising or sharpening* step. We evaluate three contrast methods and five denoising filters in all pairwise combinations, yielding **15 distinct pipelines**. Each pipeline is trained and evaluated independently under identical YOLOv8n-pose hyperparameters.

3.1 Contrast Enhancement Methods

Original. The raw 8-bit greyscale frame is passed through without alteration. This baseline establishes the lower bound for contrast-related improvements and isolates the contribution of the denoising stage.

CLAHE. Contrast Limited Adaptive Histogram Equalisation (CLAHE) [8] operates directly on the single greyscale intensity channel. The histogram of each 8×8 tile is equalised and clipped at limit $\beta = 2.0$, redistributing contrast locally to enhance bee-body edges without amplifying noise in uniform comb regions.

Multi-Scale Retinex (MSR). Multi-Scale Retinex [9] models the human visual system's luminance adaptation. Applied to the greyscale intensity image I, the log-domain reflectance is estimated by subtracting a weighted sum of Gaussian surround functions at three scales $\sigma \in \{25, 50, 100\}$ px:

$$R(x,y) = \frac{1}{3} \sum_{\sigma} \Big[\log I(x,y) - \log\big(\max(I * G_\sigma,\, 10^{-3})\big) \Big], \tag{3}$$

where G_σ is an isotropic Gaussian with standard deviation σ. Log-ratio values are clipped to $[-5, 5]$ before min-max normalisation to prevent numerical instability on near-uniform regions.

3.2 Edge-Preserving Denoising Filters

Bilateral Filter. The bilateral filter [10] extends Gaussian smoothing by weighting each neighbouring pixel by both spatial proximity and radiometric similarity:

$$\hat{I}(p) = \frac{1}{W_p} \sum_{q \in \Omega} G_{\sigma_s}(\|p - q\|) \cdot G_{\sigma_r}(|I(p) - I(q)|) \cdot I(q). \tag{4}$$

In our experiments, we use a spatial neighborhood of diameter $d = 50$, with spatial standard deviation $\sigma_s = 100$ and range standard deviation $\sigma_r = 20$.

Non-Local Means (NLM). Non-Local Means [12] averages patches across the *entire* image, weighted by patch-similarity rather than spatial distance:

$$\hat{I}(p) = \frac{1}{C(p)} \sum_{q} w(p,q)\, I(q), \quad w(p,q) = \exp\left(-\frac{\|v(p) - v(q)\|^2}{h^2}\right), \tag{5}$$

where $v(\cdot)$ denotes a patch around the pixel and $h = 10$ controls the filter strength. The size of the template and the search window is 7×7 and 21×21 px, respectively. NLM is highly effective at removing grain while retaining fine texture, at the cost of high computation (≈157 ms per 448×448 greyscale frame on Apple Silicon, see thereafter on Table 2).

Symmetric Nearest Neighbour (SNN). The SNN filter [13] considers symmetric pairs (a, b) of neighbours about the centre pixel c and selects whichever is radiometrically closer to c:

$$\hat{I}(c) = \frac{1}{|\mathcal{P}|} \sum_{(a,b) \in \mathcal{P}} \begin{cases} a & \text{if } \|a - c\| \leq \|b - c\|, \\ b & \text{otherwise,} \end{cases} \tag{6}$$

with radius $r = 2$ px. This strategy smooths homogeneous regions while efficiently keeping edges and isolated bright features.

Table 2: Preprocessing latency per step at 448×448 px on Apple Silicon. ✓ marks steps that satisfy the live budget (≤ 20 ms).

Filter / Step	ms / frame	Live?
EPOAF	7 ms	✓
CLAHE	10 ms	✓
Bilateral filter	86 ms	—
Retinex (MSR)	117 ms	—
NLM	157 ms	—
SNN	166 ms	—
Kramer-Brückner	179 ms	—

Kramer-Brückner Filter. The Kramer-Brückner filter [14] extends SNN by initialising the accumulator with the centre pixel (giving it unit weight) before summing selected neighbours. This double-weighting of the centre better preserves point features and fine structural details compared to SNN.

EPOAF. The Edge-Preserving Oriented Adaptive Filter [11] computes local gradient orientation and averages each pixel along the edge *tangent* direction only, leaving the normal direction (across the edge) unsmoothed:

$$\hat{I}(p) = \frac{1}{1+|S|}\Big[I(p) + \sum_{s\in S} I(p + s\,\hat{\mathbf{t}}_p)\Big], \quad S = \{-2, -1, +1, +2\}, \tag{7}$$

where $\hat{\mathbf{t}}_p$ is the unit tangent to the edge at p. EPOAF removes noise parallel to edges without blurring them. Critically, it runs in only ≈7 ms per frame — the fastest denoising filter evaluated and the only one unconditionally live-viable regardless of the contrast step.

3.3 Pipeline Combinations

The pairwise combinations of three contrast methods and five filters yield **15 two-step pipelines**. Figure 3 presents the visual output of each combination on a representative hive frame, and Figure 4 gives the full pipeline-structure overview.

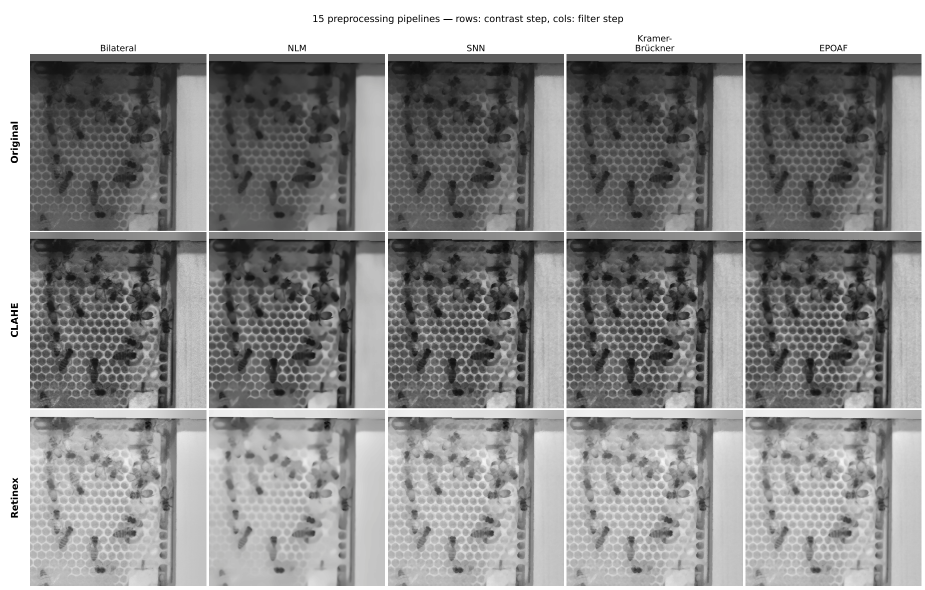


Fig. 3: Preprocessing pipeline outputs on a single hive frame. Contrast step, Line 1: Original, Line 2: CLAHE [8], Line 3: Retinex [9]. Denoising filter, from left to right: Bilateral [10], NLM [12], SNN [13], Kramer-Brückner [14], EPOAF [11].

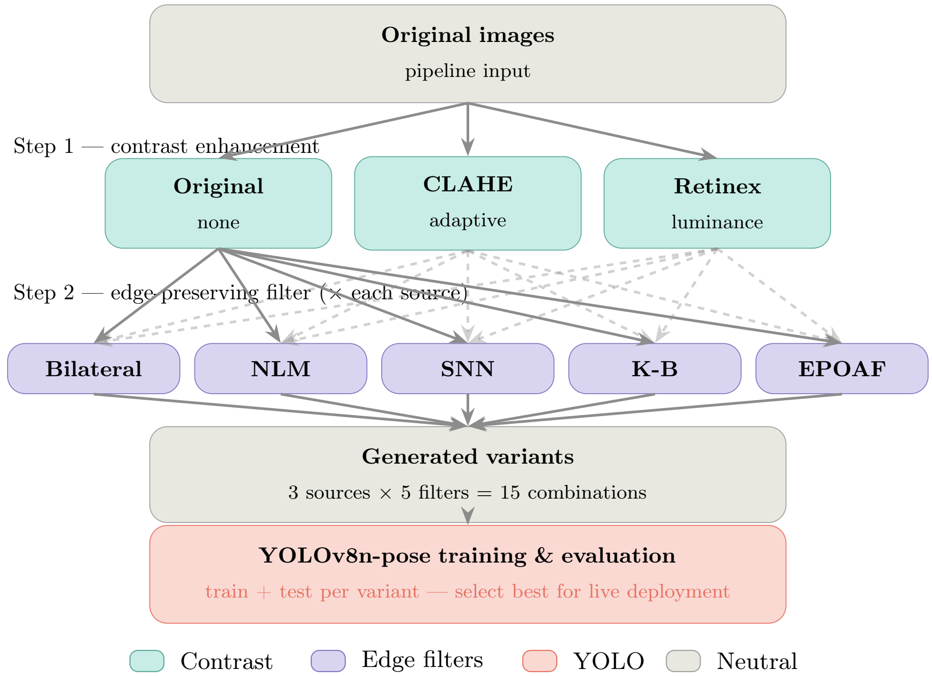


Fig. 4: Overview of the preprocessing pipeline. Three contrast methods feed five edge-preserving filters, yielding 15 pipeline variants. These variants are then evaluated under identical YOLOv8n-pose training conditions to identify the best preprocessing configuration.

3.4 Live Deployment Constraints

A principal objective of this system is continuous, real-time hive monitoring. At 30 fps, the total frame budget is $\lfloor 1000/30 \rfloor = 33$ ms. YOLOv8n-pose inference on Apple Silicon MPS requires approximately 12 to 15 ms per frame, leaving a preprocessing budget of roughly **18 to 21 ms**. Note that these figures are hardware-specific; on other platforms the inference time and available preprocessing budget will differ.

Per-filter benchmark. Table 2 reports measured preprocessing latency at 448×448 px on Apple M-series hardware (single core, averaged over 8 runs).

NLM, SNN, and Kramer-Brückner filters are neighbourhood-scan algorithms implemented in Python/NumPy and are fundamentally $\mathcal{O}(n \cdot r^2)$; already using OpenCV's optimised `fastNlMeansDenoising` for NLM, latencies cannot be reduced substantially without CUDA. Retinex is dominated by three large-σ Gaussian blurs ($\sigma \leq 100$ px); a single-scale variant could reduce this to ≈35 ms at the cost of reduced shadow recovery, still outside the live budget.

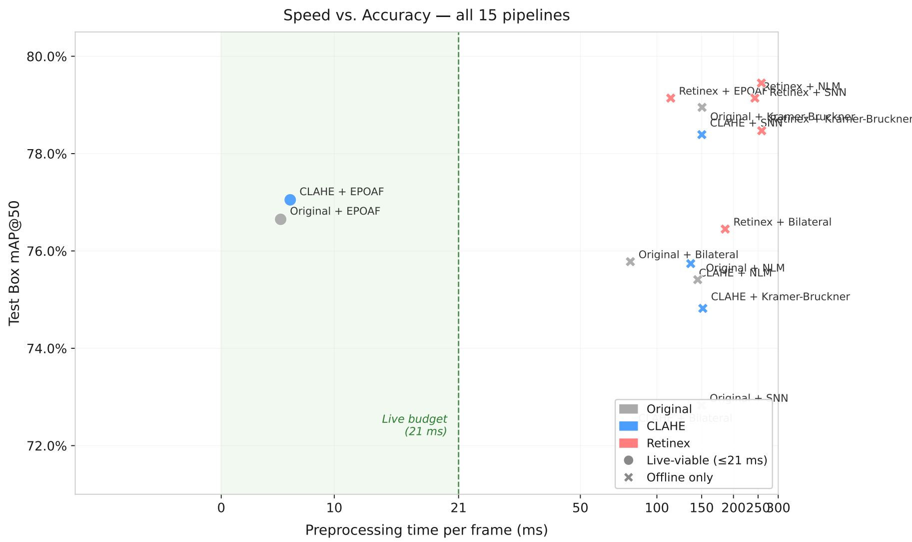


Fig. 5: Speed vs. accuracy for all 15 preprocessing pipelines. Circles (●) satisfy the 30 fps live constraint; crosses (×) do not. The shaded green zone marks preprocessing latency ≤ 21 ms. Colours encode the contrast family (grey: Original; blue: CLAHE; red: Retinex). The best live pipeline (CLAHE + EPOAF, 77.0%) is 2.5 pp below the offline optimum (Retinex + SNN, 79.5%).

Live-viable pipelines. Accounting for the ≈12 ms YOLO inference time, only combinations with total preprocessing under 21 ms satisfy the 33 ms real-time constraint. Among the 15 evaluated pipelines, only those using EPOAF as the filter qualify:

1. **Original** + **EPOAF** — 7 ms preprocessing, **76.6%** Box mAP50,
2. **CLAHE** + **EPOAF** — 17 ms, **77.0%** Box mAP50.

The best-performing offline pipeline (Retinex + SNN, 79.5%) requires 283 ms of preprocessing — eight times the entire 33 ms live budget. **CLAHE** + **EPOAF** is therefore the recommended live pipeline: the CLAHE contrast step costs 10 ms but provides a consistent accuracy lift, while EPOAF's 7 ms tangent-direction smoothing makes the combination live-viable. Critically, the accuracy penalty for live deployment is only **2.5 pp** (percentage-point difference) relative to the offline optimum — a substantially smaller gap than typically observed with heavier denoising filters.

Figure 5 visualises the full speed-accuracy landscape for all 15 pipelines. The two live-viable pipelines are the only ones inside the shaded green zone; the horizontal spread of the remaining 13 reflects the wide range of filter latencies (86–296 ms), while their vertical spread (72–80%) reflects the genuine preprocessing contribution to accuracy.

4 Training Protocol

4.1 Dataset and Splits

Annotations were created in COCO keypoint format using an 800-image hive dataset captured with the modular IR observation setup described in Section 1. Each image is annotated with instance bounding boxes, a binary class label (cleaning / normal bee), and up to three keypoints per individual, each carrying a COCO visibility flag ($v \in \{0, 1, 2\}$). Annotations with no keypoints labelled are excluded from training. The dataset is split at the image level with a 70/15/15% train/validation/test ratio using a fixed random seed of 42, as reported in Table 3.

4.2 YOLOv8n-pose Hyperparameters

All 15 preprocessing pipelines described in Section 3, are evaluated under identical YOLOv8n-pose [16] training conditions, as summarized in Table 4.

Mosaic augmentation is kept active throughout all epochs (`close_mosaic=0`) to prevent artificial inflation of validation mAP50 in the final epochs. A small batch size (4) is used to avoid MPS out-of-memory errors and to mitigate the Task-Aligned Learning (TAL) shape mismatch instability observed in ultralytics 8.4.x on Apple Silicon. Stale checkpoints are cleared before each pipeline run to prevent auto-resume from a previous training affecting results.

5 Results

5.1 Full Pipeline Comparison

Table 5 reports test-set Box and Pose mAP50 for all 15 preprocessing pipelines, sorted by Box mAP50. Live-viable pipelines (preprocessing $\leq$ 21 ms) are marked ✓. Figure 6 plots the same results as a ranked horizontal bar chart with Box and Pose mAP50 side by side, colour-coded by contrast family.

Key observations.

1. **Retinex dominates the top positions.** The top pipeline is Retinex + SNN (79.5%), and three of the top-five pipelines use Retinex as the contrast step. Retinex's illumination-removal reveals true structural gradients and enhances patch distinctiveness across the comb, benefiting all five denoising filters.

Table 3: Dataset split statistics.

Split	Images	Bee instances	Cleaning / Normal
Train	560	≈5 360	≈131 / ≈5 229
Validation	120	≈1 149	≈28 / ≈1 121
Test	120	≈1 149	≈28 / ≈1 121
Total	800	7 658	187 / 7 471

Table 4: YOLOv8n-pose training hyperparameters (identical for all 15 pipelines).

Hyperparameter	Value
Epochs (max)	50
Early-stopping patience	15
Batch size	4
Image size	448 px
Optimiser	AdamW [2] (auto)
Mosaic augmentation	0.5 (all epochs, `close_mosaic=0`)
Horizontal flip (p)	0.5
Rotation (±deg)	10
HSV hue / saturation / value	0.015 / 0.4 / 0.3
Scale	0.3
Device	Apple MPS

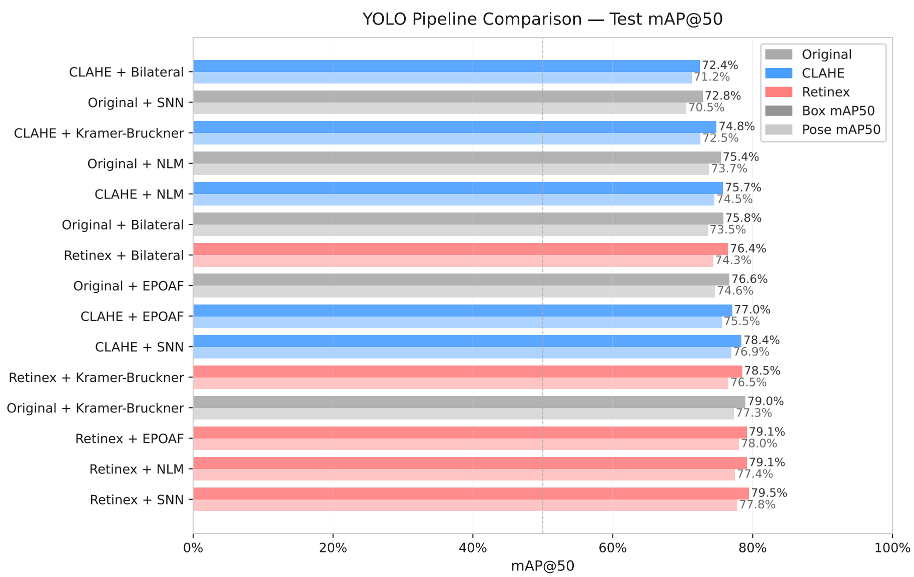


Fig. 6: Test-set Box mAP50 (solid) and Pose mAP50 (faded) for all 15 preprocessing pipelines, ranked in descending order of Box mAP50. Bar colours encode the contrast family: grey = Original, blue = CLAHE, red = Retinex.

2. **Narrow accuracy spread.** The top-15 pipelines span only 7.1 pp (72.4–79.5%), and the top-7 span just 2.5 pp (77.0–79.5%). With 120 test images, differences below ≈3 pp correspond to fewer than two instances and lie within sampling noise.
3. **EPOAF is the live filter of choice.** EPOAF at 7 ms is the only filter compatible with real-time constraints. Paired with CLAHE, it achieves 77.0%

Table 5: Test-set YOLOv8n-pose results for all 15 preprocessing pipelines, sorted by Box mAP50. ✓ marks pipelines satisfying the 30 fps live constraint (preprocessing $\leq$ 21 ms).

Pipeline	Box mAP50	Pose mAP50	Preproc. (ms)	Live
Retinex + SNN	79.5%	77.8%	283	
Retinex + NLM	79.1%	77.4%	274	
Retinex + EPOAF	79.1%	78.0%	124	
Original + Kramer-Brückner	79.0%	77.3%	179	
Retinex + Kramer-Brückner	78.5%	76.5%	296	
CLAHE + SNN	78.4%	76.9%	176	
CLAHE + EPOAF	77.0%	75.5%	17	✓
Original + EPOAF	76.6%	74.6%	7	✓
Retinex + Bilateral	76.4%	74.3%	203	
Original + Bilateral	75.8%	73.5%	86	
CLAHE + NLM	75.7%	74.5%	167	
Original + NLM	75.4%	73.7%	157	
CLAHE + Kramer-Brückner	74.8%	72.5%	189	
Original + SNN	72.8%	70.5%	166	
CLAHE + Bilateral	72.4%	71.2%	96	

— only 2.5 pp below the offline optimum — making live deployment highly competitive with offline analysis in this dataset.

4. **SNN and Kramer–Brückner remain competitive despite their age.** SNN (79.5% with Retinex) and KB (79.0% with Original) rank first and fourth despite their high latency in our Python/NumPy implementation. These classical algorithms (1983 and 1975) preserve bee-body boundaries well through their pixel-selection strategy. Their $\mathcal{O}(n \cdot r^2)$ complexity at small radius r suggests that optimised C++ or GPU implementations could bring them within the live budget.

Figure 7 plots the validation Box mAP50 per epoch for all 15 pipelines throughout training. The top-5 pipelines (by final test mAP50) are drawn at full opacity; the remaining ten are faded for clarity. Curve colours encode the contrast family: red = Retinex, blue = CLAHE, grey = Original.

All pipelines converge within 30–40 epochs under the early-stopping patience of 15. Retinex-based pipelines (red) cluster consistently above CLAHE (blue) and Original (grey) variants throughout training, confirming that the illumination-removal step provides a stable, epoch-independent advantage rather than a transient initialisation effect. The two live-viable pipelines (Original + EPOAF and CLAHE + EPOAF, solid lines) track the upper cluster closely in the first 20 epochs before their final values settle 2–3 pp below the offline optimum, reflecting the modest accuracy cost of the faster EPOAF denoising step.

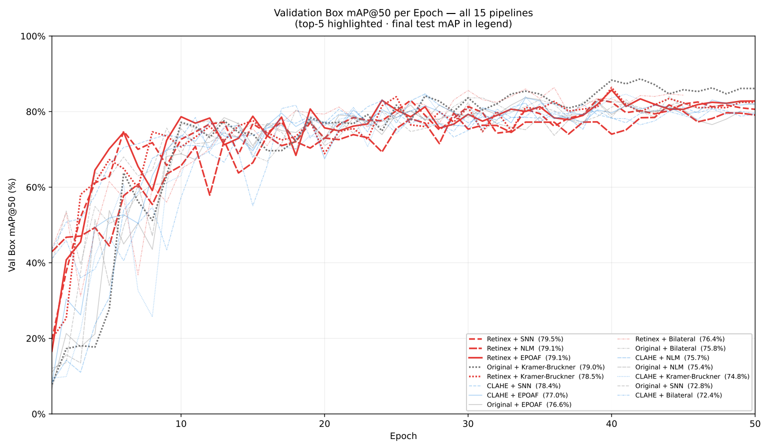


Fig. 7: Validation Box mAP@50 per epoch for all 15 preprocessing pipelines. Top-5 pipelines (by final test mAP50) are highlighted; the remaining ten are faded. Colours encode the contrast family (red: Retinex, blue: CLAHE, grey: Original); line styles encode the filter (solid: EPOAF, dashed: SNN, dash-dot: Bilateral, long-dash: NLM, dotted: Kramer-Brückner).

6 Behavioural Analysis Framework

Detections with head/tail visibility score > 0.5 form the filtered set $\mathcal{B} = \{b_i\}$. Per bee we derive centroid $\mathbf{c}_i = (\mathbf{h}_i + \mathbf{t}_i)/2$, unit heading $\hat{\mathbf{u}}_i = (\mathbf{h}_i - \mathbf{t}_i)/\left\|\mathbf{h}_i - \mathbf{t}_i\right\|$, and angle $\theta_i = \mathrm{atan2}(u_i^y, u_i^x)$. Eight analyses are derived from these quantities; Table 6 summarises them.

Table 6: The eight behavioural analyses computed per frame.

Analysis	Key output	Ecological signal
Heading map	Arrow overlay (orange/green)	Class distribution
Rose / Rayleigh r	$r \in [0, 1]$, mean $\bar{\theta}$	Global alignment
Local alignment	Score $a_i \in [-1, 1]$ per bee	Sub-group coherence
Flow field	Divergence, curl maps	Spread / vortex zones
Following chains	Chain length ℓ	Tandem running
Density heatmap	KDE peak $\hat{p}_{\max}$	Congestion hotspots
Facing pairs	Count $\lvert\mathcal{F}\rvert$	Trophallaxis candidates
Crowding index	ρ_i per bee	Local density

Rayleigh statistic. Global directional concentration is measured by the mean resultant length [4]:

$$r = \left\| \tfrac{1}{n} \textstyle\sum_i \hat{\mathbf{u}}_i \right\| \in [0, 1], \tag{8}$$

with $r>0.7$ strongly aligned, $r\leq 0.2$ scattered.

Local alignment. The $K=5$ nearest-neighbour dot-product score $a_i = \frac{1}{K}\sum_{j\in\mathcal{N}_K(i)} \hat{\mathbf{u}}_i \cdot \hat{\mathbf{u}}_j$ resolves spatially distinct sub-groups that global r conflates.

Flow field. Unit headings are interpolated onto a grid (cubic spline interpolation of scattered heading vectors onto a regular grid via `scipy.griddata`) and differentiated to obtain divergence $\nabla \cdot \mathbf{F} = \partial_x F_x + \partial_y F_y$ (spread/convergence zones) and curl $(\nabla\times\mathbf{F})_z = \partial_x F_y - \partial_y F_x$ (vortex patterns).

Following chains. Bee i follows j when $\|\mathbf{h}_i - \mathbf{t}_j\| < 0.15\min(H,W)$ and $\hat{\mathbf{u}}_i \cdot \hat{\mathbf{u}}_j > \cos 60°$. Maximal directed paths in the resulting graph indicate tandem-running.

Facing pairs / Trophallaxis detection. $(i,j) \in \mathcal{F}$ iff bounding boxes overlap, head-to-head distance $\|\mathbf{h}_i - \mathbf{h}_j\| \leq 40$ px, mutual facing dot product $\hat{\mathbf{u}}_i \cdot \hat{\mathbf{u}}_j \leq -0.80$, and each bee points toward the other's head (mutual-facing gate ≥ 0.25). These four gates minimise false-positive trophallaxis or antennation candidates [7].

For video input the statistics are computed per frame and assembled into time-series (e.g. Rayleigh spikes during waggle dances, rising $|\mathcal{F}|$ during feeding cascades).

7 Conclusion

A complete pipeline for honeybee detection, two-class classification, and three-keypoint pose estimation using YOLOv8n-pose is presented, together with a suite of eight ecologically motivated collective-behaviour analyses derived solely from predicted head and tail positions. The sliding-window tiling scheme extends the system to full-resolution hive cameras ($\geq 1500 \times 1200$ px) without retraining.

Our systematic evaluation of 15 preprocessing pipelines on an 800-image hive dataset yields two practical conclusions. For *live deployment at 30 fps* under our Python/NumPy implementation, only two pipelines satisfy the 21 ms preprocessing budget: Original + EPOAF (76.6%, 7 ms) and **CLAHE + EPOAF** (77.0%, 17 ms, recommended). Crucially, the live-to-offline accuracy gap is only **2.5 pp** — a substantially smaller penalty than typically observed with heavier denoising approaches, and within the sampling-noise floor for this dataset size. It is worth noting that SNN and Kramer–Brückner, while slow in Python, have a favourable algorithmic complexity ($\mathcal{O}(n \cdot r^2)$ at small radius); optimised C++ or GPU implementations could make them live-viable without sacrificing their accuracy advantage.

Several limitations remain open. First, the training set size is moderate; a larger annotation campaign would further reduce variance between pipeline rankings and push the absolute performance ceiling higher. Second, the current system processes each frame independently and does not maintain inter-frame identity: detections are not linked across time, so trajectory-based statistics (e.g. individual speed, path curvature) are not yet available. Integrating a multi-object tracker would unlock these capabilities. Third, the cleaning-bee class imbalance (187 vs. 7,471 normal bees) limits the classifier sensitivity; targeted data collection or synthetic augmentation of cleaning-bee frames would improve this.

Future work will focus on semi-supervised extension of the annotation set, multi-object tracking across video frames, benchmarking the descriptors against expert-annotated reference sequences, and integration of waggle dance decode logic [17] into the behavioural analysis pipeline.

Acknowledgements: The authors gratefully acknowledge funding support from the ANR BeeBehaviours project (an. code: RRCPUBLNAT24056H).

References

1. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: ECCV, LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014)
2. Loshchilov, I., Hutter, F.: Decoupled weight decay regularisation. In: ICLR (2019)
3. Druon, S., Magnier, B., Bares, J., Triboulet, J., Oms, J.L., Rousset, M.: Sensor-based behavior analysis and modeling for the honeybee superorganism: a platform. In: MMSP, pp. 1–6. IEEE (2023)
4. Mardia, K.V., Jupp, P.E.: Directional Statistics. Wiley, Chichester (2000)
5. Boenisch, F., et al.: Tracking all members of a honey bee colony over their lifetime using learned models and propositional logic. In: Frontiers in Robotics and AI, **5**(35), (2018)
6. Bozek, K., Hebert, L., Portugal, Y., Mikheyev, A.S., Stephens, G.J.: Markerless tracking of an entire honey bee colony. *Nature Communications* **12**, 1733 (2021)
7. Farina, W.M., Grüter, C., Díaz, P.C.: Social learning of floral odours inside the honeybee hive. Proc. R. Soc. B **272**(1575), 1923–1928 (2005)
8. Zuiderveld, K.: Contrast limited adaptive histogram equalisation. In: Graphics Gems IV, pp. 474–485. Academic Press (1994)
9. Rahman, Z., Jobson, D.J., Woodell, G.A.: Multi-scale retinex for color image enhancement. In: ICIP, vol. 3, pp. 1003–1006. IEEE (1996)
10. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: ICCV, pp. 839–846. IEEE (1998)
11. Bakker, P., van Vliet, L.J., Verbeek, P.W.: Edge preserving orientation adaptive filtering. In: CVPR, vol. 99, pp. 535–540 (1999)
12. Buades, A., Coll, B., Morel, J.-M.: A non-local algorithm for image denoising. In: CVPR, vol. 2, pp. 60–65. IEEE (2005)
13. Lee, J.S.: Digital image smoothing and the sigma filter. Comput. Vision Graph. Image Process. **24**(2), 255–269 (1983)
14. Kramer, H.P., Bruckner, J.B.: Iterations of a non-linear transformation for enhancement of digital images. Pattern Recognit. **7**(1–2), 53–58 (1975)
15. Ruttner, F.: Biogeography and Taxonomy of Honeybees. Springer, Berlin, Heidelberg (1988)
16. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO. `https://github.com/ultralytics/ultralytics` (2023)
17. Wario, F., Wild, B., Rojas, R., Landgraf, T.: Automatic detection and decoding of honey bee waggle dances. PLOS ONE 12(12), e0188626 (2017)