# Who Is Who in Judo? Role-Aware Detection in Match Video

**Lucas Raniére J. Santos**[1], **Everton Kauan S. Oliveira**[1],
**Everton Leandro G. Alves**[1], **João Brunet**[1]

[1]Federal University of Campina Grande, Campina Grande - PB, BR

***Abstract.*** *Computer vision enables scalable solutions for sports performance analysis and competition management. We present publicly available fine-tuned YOLO models for role-aware actor detection in judo match videos, trained on amateur and professional competition data and supported by a replicable frame curation pipeline. The models distinguish referees and athletes according to white and blue judogi roles. Our best model achieves a precision of 0.834, a recall of 0.923, and an mAP50 of 0.854, outperforming a naive baseline across all metrics. These results demonstrate that domain-specific fine-tuning with role-aware annotations improves detection accuracy and robustness, supporting automated sports analytics and referee assistance applications.*

## 1. Introduction

Combat sports such as Judo present a uniquely challenging environment for performance analysis. The combination of rapid movements, frequent occlusions, and intense body-to-body interactions complicates the reliable detection and tracking of athletes in video data. As a result, traditional video analysis pipelines, largely dependent on manual annotation and subjective interpretation, often fail to deliver consistent, scalable, and objective insights [Mendes-Neves et al. 2023].

In recent years, advances in Computer Vision (CV), particularly those driven by deep learning architectures, have enabled a paradigm shift toward automated and high-fidelity data extraction from raw video streams [Fujii 2025]. This transformation is especially relevant in judo, where the intricacy of athlete interactions and the high stakes of officiating decisions demand levels of precision and consistency that exceed human observational capabilities [Shariatmadar and Osman 2025]. Consequently, CV-based systems are increasingly being explored to enhance both analytical and refereeing processes in combat sports.

By leveraging machine learning, CV techniques can support a wide range of tasks, including field registration, multi-object tracking, action recognition, and pose estimation. These capabilities enable large-scale, data-driven analysis of athlete behavior and interaction dynamics, reinforcing the growing importance of automated data collection in modern sports analytics [Fujii 2025].

Within this context, YOLO (You Only Look Once) [Redmon et al. 2016] has emerged as a prominent real-time object detection framework. Unlike traditional methods such as R-CNN, which require multiple passes over an image, YOLO formulates detection as a single regression problem, processing the entire image in one forward pass through a neural network. This unified approach enables simultaneous prediction of object bounding boxes and class labels, resulting in substantial gains in processing speed. Due to its efficiency and strong empirical performance in real-time sports applications, YOLO-based models represent a promising direction for addressing the challenges inherent to

the judo domain [Bhat et al. 2024, Host and Ivaŝić-Kos 2022, Quinn and Corcoran 2022, Quinn and Corcoran 2024].

In this work, we introduce a method for precise actor detection and classification in judo match videos. We employ a YOLO-based model, enhanced through transfer learning, and utilize a carefully curated real-world dataset to identify and classify the referee and both athletes (wearing blue and white judogis). The primary contributions of this paper include:

- YOLO11 and YOLO26 model weights trained to classify referees and judo athletes distinguished by judogi color[1];
- An empirical study on the effectiveness of YOLO-based models for detecting and classifying actors in judo match scenarios;
- A detailed evaluation of detection performance under challenging conditions, providing insights into the limitations and potential of object detection models in combat sports;
- Practical insights and recommendations for applying computer vision techniques to judo analytics and similar interaction-intensive sports.

## 2. Related Work

In sports video analysis using computer vision (CV), research has evolved from manual approaches to highly automated deep learning-based systems [Bhat et al. 2024, Host and Ivaŝić-Kos 2022]. A typical pipeline includes preprocessing, object detection and tracking, pose estimation, and action recognition [Bhat et al. 2024, Miyaguchi et al. 2024].

CV-based performance analysis enables objective feedback for assessing biomechanics, tactics, and techniques. Common applications include movement analysis, detection and tracking, and human action recognition (HAR) [Bhat et al. 2024, Host and Ivaŝić-Kos 2022]. For example, Mendes-Neves *et al.* [Mendes-Neves et al. 2023] predict soccer ball speed from pose data, achieving a 67% correlation with shot speed. Salim *et al.* [Salim et al. 2010] analyze badminton smashes, revealing biomechanical differences between male and female athletes through motion capture.

In action recognition, Lai *et al.* [Lai et al. 2024] propose FACTS, a transformer-based model achieving 90% accuracy in fencing and 83.3% in boxing without pose estimation. Quinn and Corcoran [Quinn and Corcoran 2022] combine YOLOv5 detection with pose estimation to identify *punch* and *kick* actions in boxing and MMA.

In judo, Kato and Yamagiwa [Kato and Yamagiwa 2022] introduce RT-XSM, a method that predicts successful throws from *kumite* posture using logistic regression over posture features and technique classes.

Our work focuses on object detection. Similar to Quinn and Corcoran [Quinn and Corcoran 2022], we fine-tune YOLO models to detect athletes and referees in judo matches and make them publicly available, addressing the lack of freely available models for detecting referees and athletes differentiated by judogi color in judo match videos.

[1] https://zenodo.org/records/22088412

## 3. Proposed Approach

Our approach for actor detection in judo matches consists of training YOLO (You Only Look Once) [Redmon et al. 2016] models via transfer learning on judo-specific data. Transfer learning with YOLO is a standard approach for training object detection models on custom datasets. Rather than training a neural network from scratch, which typically requires large-scale datasets and substantial computational resources, a model pre-trained on datasets such as COCO [Lin et al. 2015] is fine-tuned for the target task.

The models are trained to detect three classes corresponding to the actors involved in a judo match: the *referee* and the athletes, categorized as *athlete_blue* and *athlete_white* according to the official judogi colors defined by the International Judo Federation[2].

### 3.1. Dataset

Our dataset comprises judo match videos from both amateur and professional competitions, provided by a major Brazilian national team and a private partner company collaborating with athletes. Based on their characteristics and sources, the data are organized into three sub-datasets. Table 1 summarizes the number of videos in each group. The "*pro*" sub-dataset, containing 271 videos of professional judo matches — adult male athletes from a variety of countries; the "*amateur*" sub-dataset, containing 186 distinct 480p videos from local and amateur tournaments, along with 108 duplicate videos available in 1080p — young and adult male athletes; and the "*brazilian_pro*" sub-dataset, containing 31 videos of professional matches featuring adult and female Brazilian judo athletes. In total, the dataset comprises 488 distinct videos across all sub-datasets, or 596 videos when including the 1080p duplicates. Due to privacy matters, the videos cannot be made publicly available.

**Table 1. Number of distinct videos across all subdatasets, totalling 488 judo match videos.**

| Subdataset | Number of Videos |
|---|---|
| ***pro*** | 271 |
| ***amateur*** | 186 (480p) and 108 (1080p) |
| ***brazilian_pro*** | 31 |
| **Total** | **488 distinct videos** |

### 3.2. Frame Temporal Stratification and Selection

To select the frames used for model training and evaluation, an automated and semi-random process was applied. This process was designed to avoid data leakage across the training, validation, and test splits and to ensure data diversity (e.g., variations in poses, camera angles, and distances).

First, the training, validation, and test sets are randomly selected at the video level, ensuring that frames from a given video appear in only one of the dataset splits. The selected video split target proportions were 70% for training, 20% for validation, and 10% for testing, applied to each sub-dataset.

Table 2 summarizes the distribution of videos across the dataset splits: the training set comprises 285 videos (189 from the *pro* sub-dataset, 75 from the *amateur* sub-dataset,

[2] https://fightingfilms.shop/en/rules-for-ijf-judogis-2023-2024/

and 21 from the *brazilian_pro* sub-dataset); the validation set includes 84 videos (55 *pro*, 22 *amateur*, and 7 *brazilian_pro*); and the test set contains 41 videos (27 *pro*, 11 *amateur*, and 3 *brazilian_pro*).

**Table 2. Videos and frames counts by sub-dataset and training split sets.**

| | Number of Videos | | | | Number of Frames | | | |
|---|---|---|---|---|---|---|---|---|
| **Sub-datasets** | **Train** | **Validation** | **Test** | **Total** | **Train** | **Validation** | **Test** | **Total** |
| ***pro*** | 189 | 55 | 27 | 271 | 700 | 200 | 100 | 1,000 |
| ***amateur (1080p)*** | 75 | 22 | 11 | 108 | 350 | 100 | 50 | 500 |
| ***brazilian_pro*** | 21 | 7 | 3 | 31 | 350 | 100 | 50 | 500 |
| **Total** | 285 | 84 | 41 | 410 | 1,400 | 400 | 200 | 2,000 |

After splitting, the following pipeline is performed for each video:

1. **Temporal sampling:** The first seven seconds of each video are discarded, as the beginning typically captures the match setup, during which the athletes are often outside the camera frame and off the tatami. This number of seconds was defined empirically and is specific to the characteristics of our datasets. The remaining portion of each video is divided into 15-second clips, from which one frame is sampled every 15 frames.
2. **SSIM-based deduplication:** For each clip, near-duplicate frames are removed using structural similarity (SSIM) [Nilsson and Akenine-Möller 2020], computed by the `scikit-image`[3] *Python* library; any frame with an SSIM $> 0.95$ relative to the previously kept frame is discarded.
3. **CLIP embedding and K-means medoid selection:** The remaining frames for each 15-second clip are encoded into semantic vectors using a CLIP `ViT-B/32` model[4]. The embeddings are then clustered using k-means ($k = 3$) with the `scikit-learn` *Python* library. From each cluster, the frame closest to the centroid (the medoid) is selected. This step ensures visual and semantic diversity within each clip window by selecting the three most representative frames.

The output of this pipeline is a set of 25,552 frames derived from all videos in the dataset, with 18,717 frames from the *pro* sub-dataset, 4,146 from the *amateur* sub-dataset, and 2,689 from the *brazilian_pro* sub-dataset. From these frames, 2,000 were randomly selected for annotation. This number was chosen to balance diversity with feasibility, providing a sufficiently varied sample while remaining manageable for manual labeling by a single annotator.

Table 2 also summarizes the number of frames in the training, validation, and test splits for each sub-dataset. The training set includes 1,400 frames, comprising 700 from the *pro*, 350 from the *amateur*, and 350 from the *brazilian_pro* sub-datasets. The validation set contains 400 frames, with 200 from the *pro*, 100 from the *amateur*, and 100 from the *brazilian_pro*. The test set contains 200 frames, with 100 from the *pro*, 50 from the *amateur*, and 50 from the *brazilian_pro*.

We deliberately chose to include more frames from the *pro* sub-dataset due to its higher overall video quality, including improved camera angles and positioning. The other

[3] `https://scikit-image.org/docs/0.25.x/auto_examples/transform/plot_ssim.html`

[4] `https://huggingface.co/laion/CLIP-ViT-B-32-laion2B-s34B-b79K`

two sub-datasets were also incorporated into training to increase variability and enhance the model's generalization to real-world scenarios.

### 3.3. Data Labeling

The 2,000 frames resulting from the previous process were manually annotated by one of the authors and subsequently reviewed by another to ensure labeling accuracy, using the CVAT[5] data annotation platform. As previously stated, three classes were annotated: (i) *referee*, denoting match referees; (ii) *athlete_blue*, denoting athletes wearing the blue judogi; and (iii) *athlete_white*, denoting athletes wearing the white judogi.

During the annotation process, some of the frames were removed from the dataset for two reasons: (i) the frame contained no athletes or referees (e.g., frames showing only the crowd or an empty tatami); and (ii) the frame contained a large number of athletes wearing only white judogi, a situation that can happen in amateur videos. The latter were excluded to maintain a more balanced distribution between the *athlete_blue* and *athlete_white* classes. Table 3 summarizes the number of remaining frames after each exclusion criterion; a total of 140 frames were removed, representing 7.0% of the original dataset, resulting in a final dataset of 1,860 frames.

**Table 3. Number of remaining frames after each exclusion criterion, resulting in a total exclusion of 7.0%.**

| Split | Original Quant. | Exc. Criteria I | Exc. Criteria II | Diff | % Change |
|---|---|---|---|---|---|
| train | 1,400 | 1,394 | 1,312 | -88 | -6.3% |
| val | 400 | 397 | 368 | -32 | -8.0% |
| test | 200 | 200 | 180 | -20 | -10.0% |
| **total** | **2,000** | **1,991** | **1,860** | **-140** | **-7.0%** |

The process resulted in 1,860 annotated frames, with *athlete_white* as the most frequent class (2,544 instances), followed by *athlete_blue* (1,994) and *referee* (1,578), for a total of 6,116 annotations.

### 3.4. Train Process

Training was conducted using the official `ultralytics`[6] *Python* library and executed on an IBM Virtual Private Cloud (VPC). The machine specifications were as follows: 32 vCPUs, 160 GB of RAM, 100 GB of storage, and two NVIDIA L4 GPUs (24 GB each), at a cost of US$3.11 per hour. The code is available online[7].

**Training Scenarios.** Two versions of YOLO models were used for training: YOLO11 (the most stable version, known for achieving strong performance in object detection) and YOLO26 (the most recent version, which promises faster CPU inference), each evaluated in nano, small, and medium configurations. Specifically, the models used are: `yolo11n.pt`, `yolo11s.pt`, `yolo11m.pt`, `yolo26n.pt`, `yolo26s.pt`, and `yolo26m.pt`.

[5] `https://www.cvat.ai/`
[6] `https://docs.ultralytics.com/usage/python/`
[7] `https://github.com/lucasraniere/judo-ai-yolo-training-actors-detection/`

**Hyperparameters.** Although the training data are diverse and cover a wide range of judo scenarios, the overall dataset size may still be considered small for a deep learning task. To mitigate this limitation, a set of data augmentation techniques was incorporated into the hyperparameter tuning process, a native feature of the `ultralytics` library, and applied during training.

## 4. Evaluation Method

We evaluate our approach in two ways. First, we assess training quality using the default outputs of the `ultralytics` library on the validation set. Second, we evaluate model performance on the test set by comparing it against a naïve color-based baseline.

### 4.1. Baseline Approach

The baseline is a simple hybrid pipeline that combines detection and tracking, using YOLO11 for person detection and a color- and geometry-based identity assignment module. It leverages prior knowledge of standardized judogi colors and requires no additional training. We chose this approach as a baseline , to the best of our knowledge, no publicly available trained models for detecting referees and athletes in judo videos were identified.

Color-based classification extracts the predominant color from representative regions of each detection using k-means clustering (`scikit-learn`) to identify the dominant pixel group. Classification follows simple rules: black indicates the referee (either as a separate class or discarded), white indicates a white-judogi athlete, and blue indicates a blue-judogi athlete. In ambiguous cases, additional cues such as belt color are used for refinement.

The system processes videos in which athletes and the referee may or may not be present. For each frame, it outputs detected individuals along with their assigned labels (white-judogi athlete, blue-judogi athlete, or referee), as well as annotated frames or videos[8].

### 4.2. Metrics

In object detection (including YOLO models), *precision* and *recall* are not computed directly on class labels alone — they depend on whether predicted bounding boxes sufficiently match ground-truth boxes. This introduces the key concept of *Intersection over Union (IoU)* [Rezatofighi et al. 2019].

Also known as the *Jaccard Index*, the *IoU* measures the ratio between the area of overlap of two bounding boxes and the total area covered by their union. The score ranges from 0 to 1, where 0 indicates no overlap and 1 a perfect match between predicted and ground-truth bounding boxes.

#### 4.2.1. From Precision/Recall to mAP.

In object detection evaluation, *mAP50* and *mAP50-95* are standard metrics that differ in the strictness of bounding box alignment [Tsirtsakis et al. 2025]. **Average Precision**

[8] `https://github.com/EvertonKauan/color-based/blob/main/README.md`

**(AP)** measures the precision–recall trade-off for a single class as the area under the precision–recall curve, while **Mean Average Precision (mAP)** extends this by averaging AP across all classes, providing a unified metric assigning equal weight to each class.

The two variants mainly differ in how bounding box overlap is assessed. **mAP50** uses a fixed *IoU* threshold of 0.50, making it a more lenient metric that rewards correct detections even with limited localization precision. In contrast, **mAP50-95** averages mAP over *IoU* thresholds from 0.50 to 0.95 (step 0.05), progressively penalizing localization errors and offering a more stringent and informative evaluation, particularly for tasks requiring precise bounding box alignment.

## 5. Results

This section presents two evaluations based on the metrics from Section 4.2: training results and test set performance, including a comparison with the color-based baseline.

### 5.1. Training Results

At the end of training, YOLO provides default metrics and visualizations on the validation set to support evaluation. Table 4 summarizes these results, showing strong performance across all models. Models were trained for up to 250 epochs: YOLO26-Medium took the longest to converge (epoch 235, 173.5 minutes), while YOLO11-Nano was the fastest to complete (epoch 221, 43.6 minutes).

The models achieved similar overall performance across the evaluated metrics. YOLO11-Medium is highlighted for attaining the highest recall (0.885) and the highest mAP50 (0.943), while YOLO11-Nano achieved the highest precision score (0.959). Among the YOLO26 variants, YOLO26-Medium also achieved the highest mAP50–95 (0.828).

**Table 4. Performance comparison of YOLO11 and YOLO26 model variants, including training time, optimal epoch, and detection metrics (Precision, Recall, mAP50, mAP50–95). The best results are highlighted.**

| Model | Best Epoch | Train Time | Precision | Recall | mAP50 | mAP50-95 |
|---|---|---|---|---|---|---|
| YOLO11-Nano | 221 | 43.6 min | **0.959** | 0.848 | 0.934 | 0.800 |
| YOLO11-Small | 237 | 78.0 min | 0.945 | 0.871 | 0.940 | 0.821 |
| YOLO11-Medium | 206 | 144.9 min | 0.936 | **0.885** | **0.943** | 0.821 |
| YOLO26-Nano | 239 | 51.0 min | 0.935 | 0.850 | 0.919 | 0.797 |
| YOLO26-Small | 132 | 61.3 min | 0.939 | 0.857 | 0.928 | 0.779 |
| YOLO26-Medium | 235 | 173.5 min | 0.949 | 0.863 | 0.935 | **0.828** |

Figure 1 shows the confusion matrices for YOLO11-Small, YOLO11-Medium, and YOLO26-Medium. All models achieve strong performance on the three-class task, with minimal inter-class confusion. The main remaining errors involve the background category, reflecting false-positive and/or false-negative detections rather than confusion among the three actor classes. The *athlete_white* class is consistently the most challenging, with higher false negative rates, likely due to its visual similarity to background elements; this could be mitigated with targeted data augmentation or additional samples.

Regarding training quality, Figure 2 summarizes the main evaluation metrics for the YOLO11-Medium model (presented for illustration purposes, as the other models

**Figure 1. Confusion matrices of the best-performing models during training: YOLO11-Small, YOLO11-Medium, and YOLO26-Medium.**

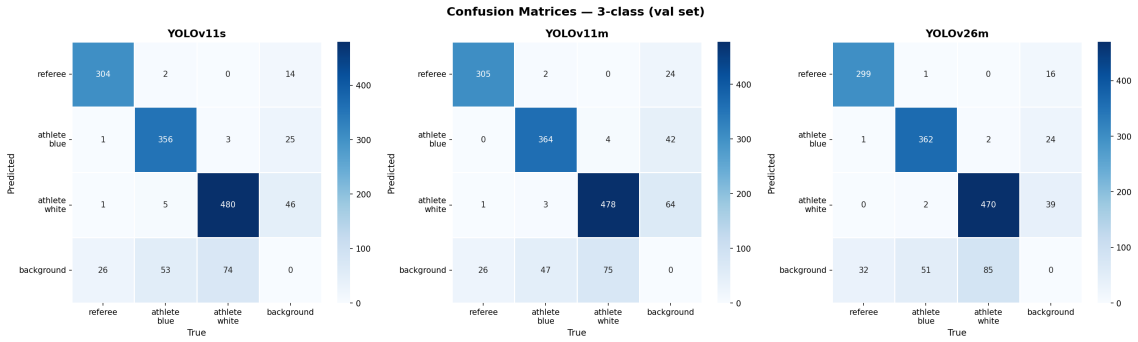
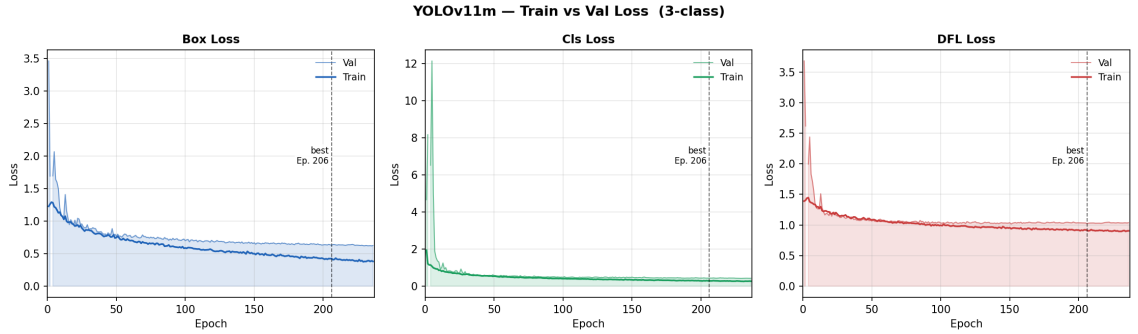



**Figure 2. Training results for the YOLO11-Medium model, indicating stable convergence, good generalization, and strong detection performance.**




achieve similar results). Overall, this represents a stable and successful YOLO training process, with only minor issues worth noting.

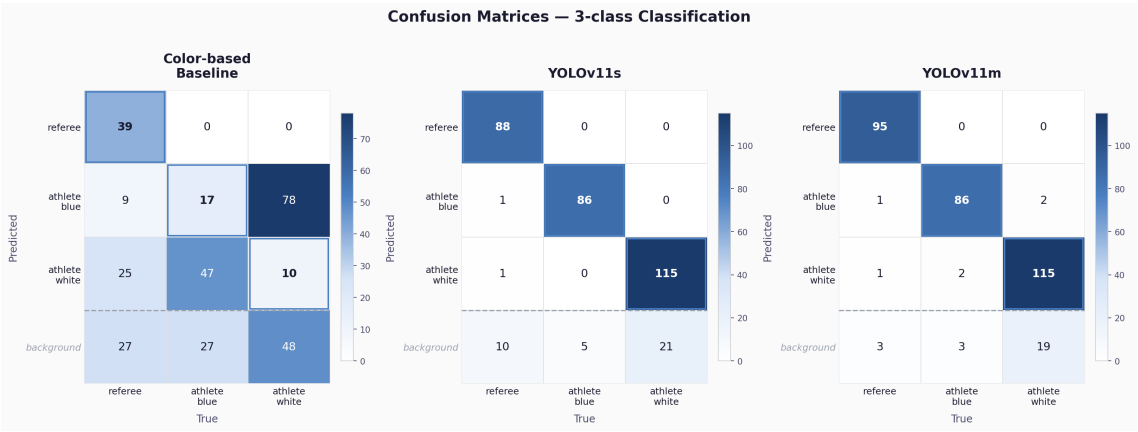
The training curves show a stable optimization process across all loss components. Classification loss converges within the first 20 epochs and remains closely aligned between train and validation, indicating good generalization. Box and DFL losses decrease smoothly, with no significant divergence up to the best checkpoint at epoch 206.

The box loss shows a small gap between training and validation losses, with the latter plateauing while the former continues to decrease slightly. Although this may indicate mild overfitting, the best-performing model is obtained at epoch 206, before substantial divergence occurs. Moreover, this curve reflects box localization rather than classification performance, whose training and validation losses remain closely aligned. The relatively small gap between training and validation losses suggests that substantial overfitting is not evident during training.

## 5.2. Baseline Comparative Results

For this evaluation, the YOLO11 small and medium results are compared against the baseline on the test set. Table 5 displays the metrics achieved by each approach, with best results highlighted. The baseline results are substantially inferior to those achieved by the trained models, with 0.472 of precision, 0.370 of recall, 0.224 of mAP50, and an

**Figure 3. Comparison of confusion matrices between the baseline approach and the YOLO11 small and medium models.**




mAP50-95 of 0.173. The trained models achieved consistently higher performance, with YOLO11-Small attaining the highest precision (0.857), and YOLO11-Medium achieving the highest recall (0.923), mAP50 (0.854), and mAP50-95 (0.660).

**Table 5. Comparison of detection performance across model variants, reporting Precision, Recall, mAP50, and mAP50–95, best results are highlighted.**

| Approach | Precision | Recall | mAP50 | mAP50–95 |
|---|---|---|---|---|
| Baseline | 0.472 | 0.370 | 0.224 | 0.173 |
| YOLO11-SMALL | **0.857** | 0.896 | 0.822 | 0.650 |
| YOLO11-MEDIUM | 0.834 | **0.923** | **0.854** | **0.660** |

The confusion matrices displayed in Figure 3 further illustrate the previously discussed results. The baseline model exhibits a high degree of inter-class misclassification, consistently confusing blue and white athletes. The baseline also exhibits a high number of background misclassifications, indicating a limited ability to distinguish between athletes and non-participants.

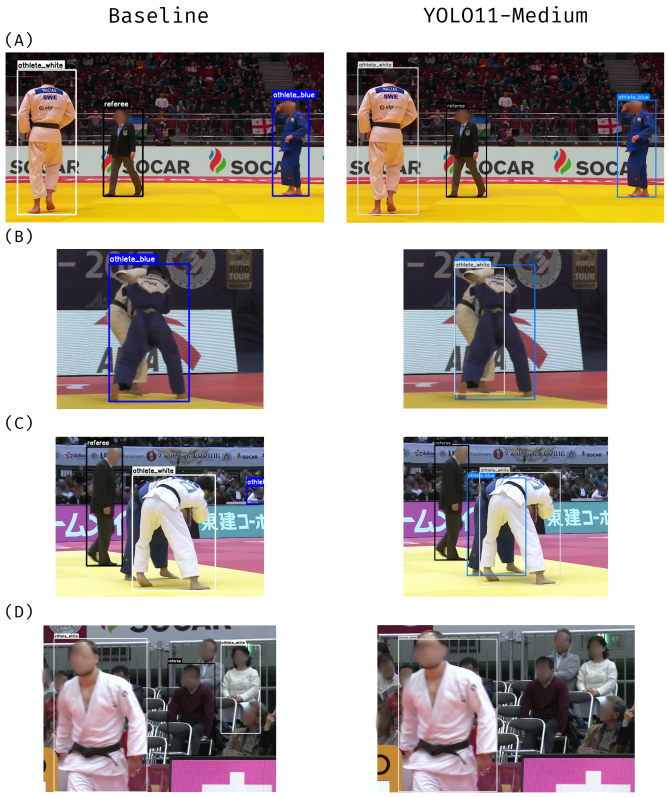
The trained models, in contrast, exhibit strong diagonal alignment, with very low inter-class misclassification, consistent with the results observed during the training phase. The primary remaining issue is background misclassification; however, its occurrence is significantly lower than in the baseline results.

Figure 4 presents a comparison of judo actor detection between the baseline approach (left frames) and the YOLO11-Medium (right frames) model across different scenarios. Frames (A) illustrate an optimal scenario in which all actors are clearly distinguishable, and no occlusion is present, resulting in perfect detection by both approaches.

In frames (B) and (C), occlusion is present, creating more challenging scenarios. The baseline approach struggles to detect athletes under these conditions, as the standard YOLO model is not optimized for detecting individuals engaged in close-contact interactions such as combat. In contrast, the proposed model, trained on domain-specific data from judo matches, demonstrates improved robustness in these situations.

Frame (D) highlights another limitation of the baseline approach: the lack of ro-

**Figure 4. Comparison between the baseline and YOLO11-Medium results, with side-by-side frames. Faces were manually blurred after object detection.**




bustness to background variability. In this example, the standard YOLO model incorrectly detects two individuals seated in the bleachers, as their clothing colors resemble those of the referee and the *athlete_white* class.

Background elements can also mislead the color-based clustering process used in the baseline approach. For instance, athletes wearing blue judogi were sometimes misclassified as *athlete_white* due to the presence of white elements in the background, while athletes wearing white judogi were misclassified as *athlete_blue* due to bluish lighting conditions in the stadium. These issues are less frequent in the trained models, as they leverage not only color information but also contextual features specific to judo, such as the structure and appearance of the judogi, independent of background variations.

These findings suggest that applying standard YOLO models to judo match scenarios is considerably less effective than using models fine-tuned for the specific task of detecting referees and athletes.

## 6. Conclusion

In this work, we fine-tuned YOLO11 and YOLO26 models on 1,860 manually annotated frames extracted from judo match videos to detect and classify referees and athletes, distinguishing them by judogi color (blue or white). The training results indicate that the models are well-trained and achieve strong performance, with a stable training process that converges appropriately and yields high detection accuracy.

We evaluate the models and compare them to a naïve color-based baseline using

*precision*, *recall*, *mAP50*, and *mAP50–95*. The proposed models substantially outperform the baseline, demonstrating that training on judo-specific data is an effective approach for object detection tasks in this domain.

Among the evaluated models, YOLO11-Small achieved the highest precision (0.857), offering reliable object detection with lower computational requirements, making it suitable for real-time applications. YOLO11-Medium achieved the highest recall (0.923), mAP50 (0.854), and mAP50-95 (0.660), representing the best overall performance and being more appropriate for offline video processing. Overall, the results indicate that the models are suitable for practical deployment, with improvements in *athlete_white* detection and background suppression representing the most promising directions for further optimization.

The trained and publicly available models represent foundational building blocks and an initial step toward the automated annotation of judo match videos and can be applied to a variety of real-world tasks and more complex pipelines. The primary application is real-time actor detection in live or recorded matches. Based on the bounding boxes generated for each actor, regions of interest can be defined according to the target task. For example, to isolate the combat area and exclude background noise, a combat zone can be defined as the union of the athletes' and referee's bounding boxes, with all remaining regions disregarded. Bounding boxes can also be used to isolate individual actors. For tasks focused on a specific role, such as the referee, only the corresponding bounding boxes classified as *referee* can be selected; the same applies to each athlete class.

### 6.1. Limitations

The primary limitations of this work are related to the dataset. The annotation process relies on manual labeling, which is inherently subject to human error. Although annotation was performed by one author and subsequently reviewed by a second author, it is not possible to ensure the complete absence of errors or perfectly precise bounding box delimitation.

The limited number of annotators also constrained the volume of data that could be labeled within a reasonable timeframe. While the training results are satisfactory and the models show potential for real-world applications, the sensitivity of deep learning performance to dataset size and composition prevents us from asserting how performance would scale with a larger annotated dataset.

Finally, despite the dataset variability, including athletes from different countries, genders, weight classes, and age groups, it does not cover all real-world scenarios, and generalization cannot be assured for matches recorded under substantially different conditions.

## Acknowledgement

This work was conducted as part of the Judo-AI research project developed in partnership between Federal University of Campina Grande (UFCG) and IBM, with funding from IBM.

## References


Bhat, H., Gupta, V., Mundada, P., and Mali, H. (2024). Analyzing sports videos using computer vision methods. In *2024 Intelligent Systems and Machine Learning Conference (ISML)*, pages 63–72. IEEE.

Fujii, K. (2025). *Computer Vision for Sports Analytics*, pages 21–57. Springer Nature Singapore, Singapore.

Host, K. and Ivaŝić-Kos, M. (2022). An overview of human action recognition in sports based on computer vision. *Heliyon*, 8(6).

Kato, S. and Yamagiwa, S. (2022). Predicting successful throwing technique in judo from factors of kumite posture based on a machine-learning approach. *Computation*, 10(10).

Lai, C., Mo, J., Xia, H., and fang Wang, Y. (2024). FACTS: Fine-grained action classification for tactical sports.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2015). Microsoft COCO: Common objects in context.

Mendes-Neves, T., Meireles, L., and Mendes-Moreira, J. (2023). A survey of advanced computer vision techniques for sports.

Miyaguchi, A., Moutahir, J., and Sutar, T. (2024). Annotation techniques for judo combat phase classification from tournament footage.

Nilsson, J. and Akenine-Möller, T. (2020). Understanding SSIM.

Quinn, E. and Corcoran, N. (2022). Automation of computer vision applications for real-time combat sports video analysis. In *European conference on the impact of artificial intelligence and robotics*, volume 4, pages 162–171.

Quinn, E. and Corcoran, N. (2024). The comparative analysis of YOLOv5/v8/v9 for object detection, tracking, and human action recognition in combat sports. In *Proc. Int. Conf. on AI Research (ICAIR)*.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., and Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 658–666.

Salim, M., Lim, H., Salim, M., and Baharuddin, M. (2010). Motion analysis of arm movement during badminton smash. In *2010 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, pages 111–114.

Shariatmadar, K. and Osman, A. (2025). Ai-enhanced precision in sport taekwondo: Increasing fairness, speed, and trust in competition (fst.ai).

Tsirtsakis, P., Zacharis, G., Maraslidis, G. S., and Fragulis, G. F. (2025). Deep learning for object recognition: A comprehensive review of models and algorithms. *International Journal of Cognitive Computing in Engineering*, 6:298–312.