Automated Swimmer Detection and Tracking using Vision-based Techniques

El Mehdi Lafkaihi * , Rania Mahdaoui[a]

[a]*Ecole Centrale de Lyon, elmehdi.lafkaihi@centrale-casablanca.ma rania.mahdaoui@centrale-casablanca.ma*

Abstract

In high-level competitive sports, video analysis plays a crucial role in assessing athletes' performance. Manual annotation of performance indicators, such as stroke rate and pacing, is a time-consuming and costly process. To address this challenge, vision-based techniques have emerged as a potential solution for automatically tracking swimmers and extracting relevant performance metrics from large volumes of video data. However, the aquatic environment poses unique challenges, including scene fluctuations due to splashes and reflections, as well as swimmers frequently submerging at various points during a race. In this paper, we propose an approach which consists of a refinement of the Yolo Model by pre-processing (Pool Segmentation) , data augmentation and post-processing steps (Kalman filter and the Hungarian algorithm) . Our approach aims to detect swimmers in all frames despite the constraints of the aquatic pool.

Keywords: Computer Vision, Swimmer detection, YOLO, Kalman Filter, Object detection

1. Introduction

In a sport as competitive as swimming, extracting useful information from race videos has become paramount for both coaches and athletes wishing to gain a competitive edge. This article explores advances in swimmer detection, using adapted Deep Learning methods, and their implications for performance evaluation. By accurately detecting swimmers, coaches can understand swimming mechanics, positioning and stroke dynamics, enabling them to optimize training strategies and improve performance. Consequently, the integration of an effective algorithm can revolutionize the way swimmers approach training, increasing their chances of success.

The paper is structured as follows : Section 2 reviews related work in vision-based sports analysis. In Section 3, we present our proposed methodology and algorithm. Experimental setup and results are discussed in Section 4, followed by a comprehensive analysis. Finally, in Section 5, we conclude the paper by highlighting our contributions and outlining future directions for improvement.

2. Related Work

Object detection aims in general at predicting a bounding box for every object and its associated class from an image. All recent high performing object detection algorithms use deep learning models and can be divided into two categories : two-stage detectors and one-stage detectors.Two-step object detectors, including R-CNN, Fast R-CNN, Faster R-CNN and Mask R-CNN, work sequentially to detect objects. They first generate region proposals, which are then refined to generate precise bounding boxes and corresponding classes. In contrast, single-step object detectors, such as YOLO, YOLO9000, SSD and RetinaNet, predict both object bounding boxes and classes in a single step, eliminating the need for explicit region proposals. This one-step approach both simplifies the detection process and ensures competitive performance in terms of accuracy and efficiency. Deep learning algorithms in object detection saw a considerable emergence in recent research especially in sports like swimming. Several studies have explored the efficiency of using deep learning algorithms to accurately identify and track swimmers in race videos. Sha et al. used a set of complex tracking algorithms to determine a swimmer's current state and therefore the best way to locate his or her current position.While they achieved good results, these were only valid for the pool on which they were focused [4]. Hong et al. proposed a CNN-based detection algorithm using Softmax and Kernel SVM as classifiers which outperformed the existing HOG (Histogram of Oriented Gradients) based method by about 6%.[1] As for Jacquelin, et al. , a tiny Unet-based model was used to detect swimmers in unconstrained swimming videos which outperforms YOLOv3 when trained on a limited dataset.[2]

3. Approach

3.1. Pre-Processing : Extracting crop pool.

First , we start by detecting the swimming pool area by color segmentation. This process consists of defining a specific range for the pool color in HSV coordinates (hue,saturation,value), with the aim of separating the pure color from the intensity (transitioning from the RGB system to the HSV system) . Subsequently, we filter to a mean of 70% centered around the mean. Additionally, morphological operations such as dilation are applied in order to reduce the noise present in the mask.

The aim is to find the exact edge of the pool, which ensures that no occlusion or additional noise is caused by the outside of the pool when locating the swimmer.

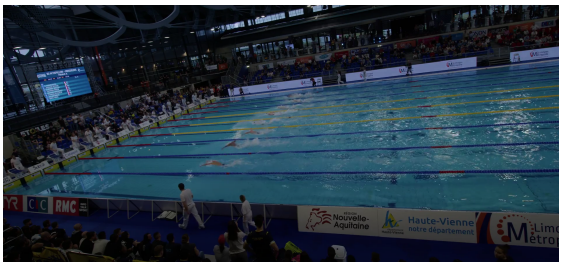

FIGURE 1 – Example of one of the pools being studied

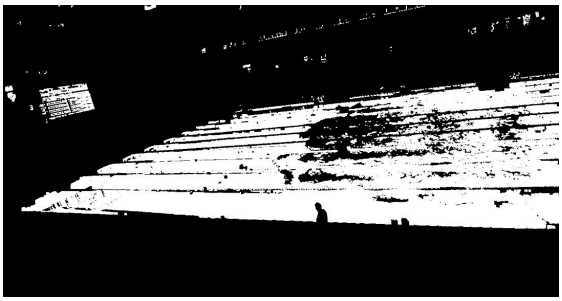

FIGURE 2 – A mask generated by color segmentation is being applied of Fig. 1

Following the pool extraction process described earlier , this step involves dividing the pool into different sections based on the lanes. This entails extracting a certain number of frames from each frame, ensuring that each extracted frame contains, at most, one swimmer.

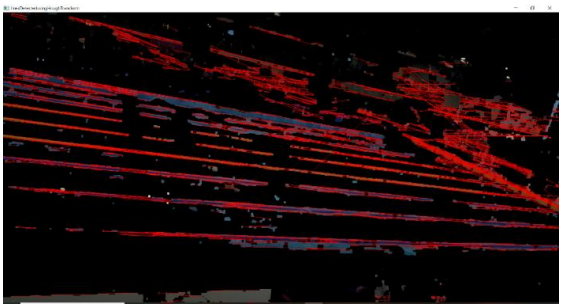

FIGURE 3 – Lanes first generated by Hough transformation

To achieve this, we employ a lane-based approach where we segment the lanes within the pool and allocate a specific region of interest (ROI) for each lane. By dividing the pool into sections corresponding to the lanes, we ensure that each extracted frame focuses on a single swimmer, maximizing the clarity and isolation of the swimmer within the frame. A class of different lane ropes is being introduced in terms of color segmentation as a dictionary , then we apply a hough transformation , Fig. 3 represent to result obtained . Some parameters being adjusted like taking the longest paths made it easier to filter to important lanes .

This approach allows us to obtain individual frames that capture the swimmers' movements without interference or overlapping instances, facilitating subsequent analysis and detection tasks.

In the case where not all lanes are detected : homography

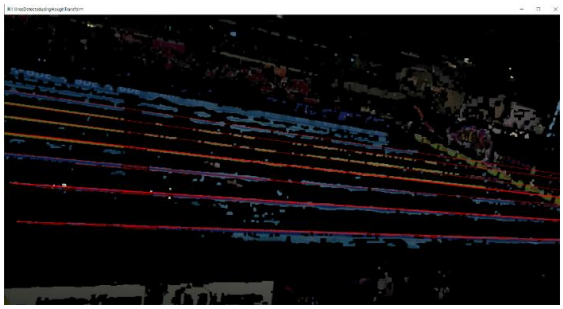

FIGURE 4 – Lanes generated after some parameters adjustement

[2] is used to render all the lane ropes parallel and considering the known number of swimmers. we then take the redundant distance ($\delta_{textmoyen}$) between each two successive swimming lanes detected , and we apply it to the lines that exceed that distance .

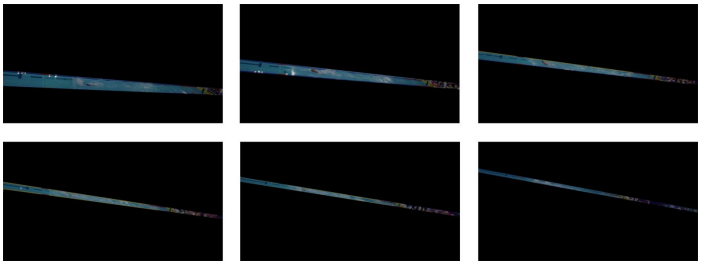

FIGURE 5 – Example of our Pre-processing approach being applied on Fig 1

3.2. *Processing : Data augmentation + Yolo*

As T. Woinoski [5] concluded, only 10-20% of the number of the training data he used was sufficient to reach a reasonable detection accuracy. Therefore, we used a dataset comprising 1000 frames and dividing it as described in the section above according to the number of swimming sections existing, ultimately obtaining a dataset of over distinct 7000 frame captured. To ensure comprehensive training, we carefully selected two classes (Swimming 0.8% and Underwater 0.2%) with diverse environmental conditions, encompassing variations in time of day and pool types (indoor/outdoor). The intention was to cover a wide range of scenarios for better model generalization. Furthermore, we took precautions to ensure that the validation and test sets exclusively contained footage from venues that were not included in the training set. This deliberate choice allows for a robust evaluation of the model's ability to handle new camera angles and color variations encountered in previously unseen venues.

In order to make the training of our model on high-quality data, we have implemented data augmentation techniques to enhance the diversity and robustness of our training dataset. Specifically, we have applied a data augmentation factor of 2, considering various edge cases that present challenges to our model's performance. These edge cases include parameters such as contrast and brightness changes, cropping, zooming out, hue variations, and horizontal flipping [2] .

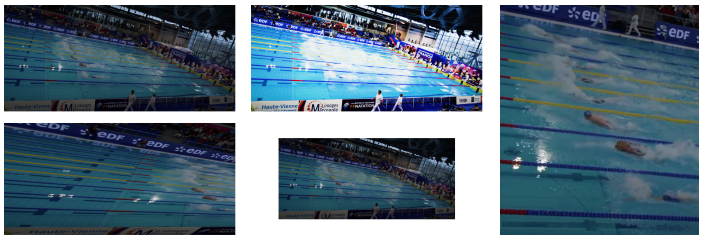

FIGURE 6 – The data augmentations used . From Left to right , top to bottom : original image , contrast and brightness change , crop , side switch , zoom out

By incorporating these data augmentation techniques, we aim to expose the model to a wider range of scenarios and variations that it may encounter during real-world deployment. This approach helps to improve the model's ability to generalize and handle diverse conditions, ultimately enhancing its accuracy and reliability in detecting and analyzing swimmer-related features. The model being used here is Yolo v5 for his ratio of accuracy by speed of image processing

3.3. *Post-Processing : Kalman Filter + Hungarian Algorithm (refinement)*

While our detector has demonstrated a promising level of accuracy, the exact positioning of the bounding box poses a challenge. Since our objective is to accurately localize the swimmer in each frame, it is crucial to enhance our system by incorporating tracking techniques capable of effectively smoothing the results and predicting the swimmer's location in cases where the detector may fail. To this end, we can choose between the L1 tracker and the Kalman filter.

When comparing both of them for swimmer detection in our research, both algorithms offer unique strengths. The L1 tracker demonstrates robustness to occlusions and adaptability to gradual appearance changes, making it a viable option for tracking swimmers within their respective sections of the pool. However, the Kalman filter surpasses the L1 tracker in its predictive capabilities and smooth trajectory estimation. With its ability to predict future states based on motion dynamics and handle missing detections, the Kalman filter proves to be an excellent choice for accurately estimating swimmer positions and velocities. Furthermore, the Kalman filter's computational efficiency is well-suited for real-time applications. Considering these factors, the Kalman filter emerges as the most suitable algorithm for our research, offering enhanced tracking performance and the ability to handle various challenges in swimmer detection during competitions.

The Kalman filter, functioning as a state-space system [4] (referring to equations 1 and 2), is employed to make predictions based on a sequence of measurements observed over time, which may contain noise and inaccuracies. In our case, the detection results [x1, y1, x2, y2], which correspond to the pixel coordinates of the top left and bottom right points of the bounding boxes, serve as the measurement (z) in the equations. The state variable (x) represents the measurement itself and its first derivative, which can be interpreted as velocity. As there is no input in the tracking problem, the variable (u) is omitted. The transition and measurement models (F and H, respectively) and the noise estimations (w and v) can be approximated relatively easily, given that the swimmer experiences relatively little movement between frames. For each frame, the Kalman filter initially predicts the swimmer's location and subsequently updates itself with the measurement if the detector successfully detects the swimmer.

$$x_k = F_k x_{k-1} + B_k u_k + w_k \quad (1)$$

$$z_k = H_k x_k + v_k \quad (2)$$

The application of the Kalman filter (Fig 7) enables the reduction of noise, leading to smoother and more stable results. The integration of the Kalman filter enhances the performance of our system ; however, it comes at the cost of increased processing time.

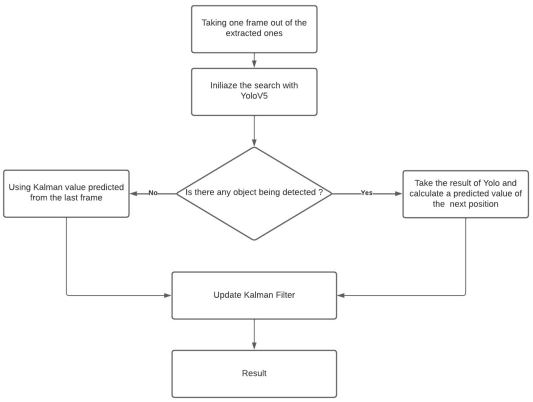

FIGURE 7 – the flowchart diagram of the use algorithm with the Kalman filter in our approach

The third algorithm in our Approach utilizes a constant velocity assumption to estimate the movement of objects across frames. It calculates the distance between the estimated bounding box of the object and all detected objects in the subsequent frame. This distance information is then input into the Hungarian algorithm, which determines the optimal associations between pairs of detections in adjacent frames. By finding the best overall associations, this algorithm effectively tracks the objects through their motion.

a) While many of our detections show a close proximity to the ground truth, joining them into complete tracks is still a non-trivial task. The smaller blue dots represent the detections, while the faint green lines depict the ground truth. Despite the overall accuracy of the detections, the challenge lies in connecting them to form coherent and continuous tracks.

b) We initiate the process by executing Algorithm to generate potential tracks, represented by the blue lines. It should be noted that in some cases, small tracks may emerge that do not correspond to any swimmers in the scene.

c) After merging the first-order tracks with the second-order tracks obtained from the previous iteration, we eliminate any short tracks. This process can be repeated iteratively to address

Algorithm 1 Track Association Algorithm

Input: p - predicted points grouped by frame,
m_d - maximum distance threshold,
m_f - maximum frames without detection
Output: Set of tracks
$O \leftarrow$ // open tracks
$C \leftarrow$ // closed tracks
for p_i in p **do**
  $p_i \leftarrow$ predicted points on frame i
  $e \leftarrow$ estimated next points from O
  Find a distance matrix D which holds the distance between every $p_{i,j}$ and e_k
  Associate points to tracks using the Hungarian algorithm on D
  Add associated points to tracks in O
  Move old tracks from O to C
  Create new tracks from points in p_i that weren't already accounted for and add to O
end for

any remaining gaps in the tracks.

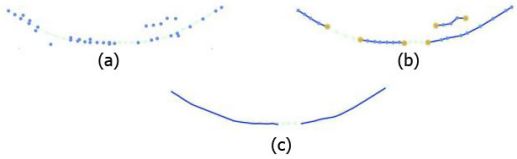

FIGURE 8 – Illustrative of hungarian algorithm in process : a) Although the majority of our detections closely align with the ground truth, merging them into complete tracks remains a non-trivial task. Despite our progress, there are still challenges in effectively eliminating noise and ensuring seamless track formation. b) the Hungarian algorithm produces a set of potential tracks represented by blue lines. However, it is important to note that occasionally, smaller tracks may emerge that do not correspond to any actual swimmers. c) In order to address the issue of short tracks that do not represent actual swimmers, we implement a filtering mechanism to remove them from our track set , we can also iterate the algorithm to cover any remaining gaps in the track formation process, leading to a more comprehensive and robust tracking solution.

4. Results

The precision will be measured using the metrics of Average Precision (AP) at 25 :
$AP25 = \frac{1}{N} \sum_i \frac{GoodDetections_i}{Positives_i}$

We evaluate the detection performance using the AP 25 metric, which allows imperfectly fitted bounding boxes around the swimmers.

As long as the intersection over union (IOU) between the detected boxes and the annotated ground truth is above 0.25. While both metrics (AP and Average Recall) are important in evaluation in any evaluation, we consider AP to be the primary metric for evaluating and comparing the performance of our detection system [2] . Due to the inherent imprecision of bounding box dimensions, even during annotation, we opted for a smaller threshold of 25 instead of 50.

	AP 25 (%)
Faster-CNN	49
Faster-CNN + tracking	72
Yolo	24
Yolo + approach	70
U-Net	39

TABLE 1 – Mesure of average Precision at 25 for Faster-CNN , Yolo , Yolo + approach , U-Net

The detection of small objects s a common weakness among general object detection algorithms, including Faster R-CNN and YOLO. Consequently , the diminished size of swimmers particularly in far lanes, often contributes to poor performance . our focus on YOLO has led to an observed increase in precision, thanks to our approach that addresses YOLO's limitations. This approach includes :

a) Enhancing accuracy for small objects through a carefully curated training dataset that incorporates data augmentation techniques.

b) Overcoming the challenges posed by overlapping objects by dividing the swimming pool into multiple sections.

c) Overcoming the limitations of limited contextual understanding by implementing the Kalman filter. This involves treating each frame as a state-space system and incorporating memory to facilitate more comprehensive tracking and contextual analysis.

d) employing the Hungarian Algorithm to eliminate noise and address the issue of confusion between reflections of light and swimmers underwater.

By implementing these strategies, we have been able to mitigate the weaknesses of YOLO and achieve improved performance, especially in terms of precision for swimmer detection.

FIGURE 9 – Two tracking experiment results. The top row is the raw detection, the bottom row is the result with our approach

FIGURE 10 – Two tracking experiment results. The top row is the raw detection, the bottom row is the result with our approach

5. Summary and further work

Our approach offers several advantages in the context of swimmer detection. Firstly, our model demonstrates robustness in predicting the positions of swimmers in video clips captured from vertical or large angles. As long as the angle between the axe of the camera and the horizontal plan of the swimming pool is greater the better results would be,they are at best for $\alpha = \pi/2$. However, there is a notable disadvantage associated with our approach . We have observed some problems in divided sections for clips taken with a small α , adding to this the high computational cost that the approach needs.

Concerning our future work, and to get over that disadvantage we have drawn inspiration from the Kalman filter's memory and prediction capabilities. We propose the following suggestion to improve our system : Instead of treating a single frame in isolation, we recognize that it may not provide sufficient visual information to accurately predict a swimmer's location and stroke probability. This limitation arises due to environmental factors such as submersion and splashing. To address this, we propose using multiple input frames centered around the frame of interest.

To incorporate the information from multiple frames, we employ early fusion by treating each frame as a separate input channel to a 2D CNN. This approach allows us to combine the information from the n input frames effectively. Each input example is then represented as a tensor with shape [n*d, h, w], where n is the number of fused frames, d = 3 represents the number of channels for each video frame (e.g., RGB channels), and h and w denote the height and width of the frames, respectively.

Acknowledgements

We would like to express our sincere gratitude to our Professor Romain Vuillemot for his invaluable guidance, support, and expertise throughout the development of this article.We are also deeply grateful for the time and effort that Professor Philippe Michel invested in providing us his advices in the writing process.

Références

[1] Hong, D., Kim, Y. : Efficient swimmer detection algorithm using CNN-based SVM. J. Korean Inf. Sci. Soc. 22(12), 79â85, 2017.

[2] Nicolas Jacquelin, Romain Vuillemot, Stefan Duffner. Detecting Swimmers in Unconstrained Videos with Few Training Data. Machine Learning and Data Mining for Sports Analytics, Sep 2021.

[3] Muhammad Rizwan Munawar,Comparing YOLOv5 and YOLOv8 : Which one should you use ?, Medium,2023.

[4] L. Sha, P. Lucey, S. Morgan, D. Pease, and S. Sridharan. Swimmer Localization from a Moving Camera. In 2013 International Conference on Digital Image Computing : Techniques and Applications (DICTA), pages 1â8. IEEE, nov 2013.

[5] T. Woinoski and I. V. BajiÄ, Swimmer Stroke Rate Estimation from Overhead Race Video,2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Shenzhen, China, 2021.

[Yang et al] Yang, Goutian Iqbal, Atif Saleem, Adeel Bozdar, Muhammad Mateen. Autonomous Swimmers Tracking Algorithm Based on Kalman Filter and CamShift, 2019.

[7] Exactly how the Hungarian Algorithm works, Think Autonomus, 2023.