

# Data-driven Summarization of Broadcasted Cycling Races by Automatic Team and Rider Recognition

Steven Verstockt<sup>a</sup>, Alec Van den broeck<sup>b</sup>, Brecht Van Vooren,  
Simon De Smul and Jelle De Bock<sup>c</sup>  
*IDLab, Ghent University-imec, Belgium*

Keywords: Data-driven Summarization, Rider Recognition, Storytelling.

Abstract: The number of spectators for cycling races broadcasted on television is decreasing each year. More dynamic and personalized reporting formats are needed to keep the viewer interested. In this paper, we propose a methodology for data-driven summarization, which allows end-users to query for personalized stories of a race, tailored to their needs (such as the length of the clip and the riders and/or teams that they are interested in). The automatic summarization uses a combination of skeleton-based rider pose detection and pose-based recognition algorithms of the team jerseys and rider faces/numbers. Evaluation on both cyclocross and road cycling races show that there is certainly potential in this novel methodology.


## 1 INTRODUCTION


Over the last few years, a weakening public interest is noticed to almost all cycling races. The ‘traditional’ cycling broadcast concept in which races are shown from start to finish is losing popularity, especially among the younger generations. Other sports face similar problems as young fans drop out (Lombardo, 2017). More personalized and interactive experiences are needed to keep the end user happy and get back the youngsters. The data-driven summarization mechanism proposed in this paper is a first step in this direction. It allows to automatically create summaries of a particular length, focusing on the events, riders and/or teams of choice. In order to achieve this, fine-grained metadata is needed for each video shot in the broadcasted stream. Available sensor data (e.g. provided by Velon or Strava) and video-based recognition techniques are used to generate the required metadata. This more detailed metadata is also an added value for archiving and querying purposes.


By improving the metadata collection and indexing strategies broadcasters will, over time, also be able to offer new services to teams, such as auto-generated team stories (e.g., a video with the

highlights of a team in a particular race which can be easily integrated in the team’s fan pages or social media account) or providing stats about how much the team or a particular rider was present/shown in the live stream. The latter could, for example, be very interesting for teams to negotiate new sponsor deals or to analyze how active a rider was in a particular race. Figure 1 shows an example of such stats that were generated with the proposed methodology. As can be seen, two teams dominated the broadcasted final of the last 40km of the Classica San Sebastian race in 2019. Finally, the fine-grained metadata can also be used for training/coaching purposes to optimize rider performance, adapt race strategies or scout opponents.

The remainder of this paper is organized as follows. Section 2 presents the related work on data-driven analysis and summarization of sport videos. Section 3 explains our data-driven summarization methodology. Results of a Tour de France test and a cyclocross race in Leuven are shown in Section 4. Finally, Section 5 concludes the paper and points out directions for future work.

<sup>a</sup>  <https://orcid.org/0000-0003-1094-2184>

<sup>b</sup>  <https://orcid.org/0000-0002-3593-7851>

<sup>c</sup>  <https://orcid.org/0000-0002-1676-9813>

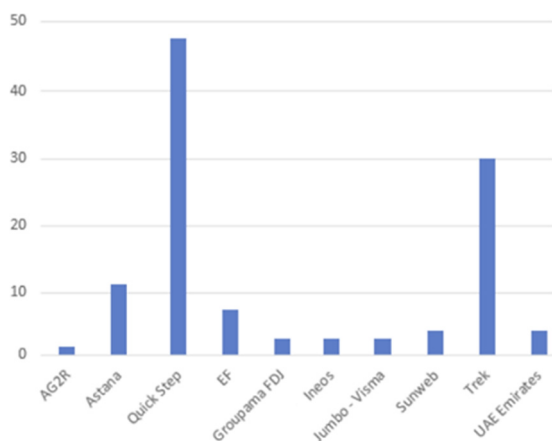


Figure 1: Clásica San Sebastián 2019 - detected teams in footage of last 40km.

## 2 RELATED WORK

The biggest problem that contemporary cycling races are struggling with is that watching a race for multiple hours can be boring and too time consuming. A solution for this problem can possibly be found in the area of highlight summarization, which has been an active research topic over the last decade. Traditional summarization methods (e.g. HMM based audio-visual highlight detection as proposed by (Qian et al., 2012) have been improved by recent deep learning-based methods that, for example, focus on player action related features to distinguish between interesting/boring parts (Tejero-de-Pablos et al., 2018). The accuracy of these techniques is still too low (+/- 75% f-score) and the metadata description is at too high level to create summarizations targeted to specific end-users/applications. Furthermore, these solutions don't tell us which rider is performing the particular action, making it difficult to be used in the proposed concept of personalized summaries. Combining the player action data with rider recognition and linking it to the available sensor data will make it possible to generate the level of metadata detail needed for this purpose.

### 2.1 Team and Rider Recognition

Related work on rider and team recognition in cycling is rather scarce or not existing, i.e., we did not find any hit in literature. However, in other sports like basketball, soccer and football, several approaches have been proposed over the last decade on how to identify players.

The jersey number recognition proposed by (Liu and Bhanu, 2019), for example, makes use of a Region-based Convolutional Neural Network (R-CNN) trained on persons/players and digits. Region proposals are geometrically related and fed to an additional classifier, ultimately generating jersey number proposals. Some results of this approach are shown in Figure 2. When a player is turned backwards the recognition goes well, but in any other orientation problems can occur, as shown in the last two examples. Furthermore, for any frontal view this approach will depend on the tracking of previous detections. Tracking in sports videos, especially when people are wearing similar sportswear and have a lot of occlusions, is also error-prone.



Figure 2: Jersey number recognition results proposed in (Liu and Bhanu, 2019).

In (Mahmood et al., 2015) an Adaboost based face detection and recognition methodology, recognizing baseball players in action, is presented. A shortcoming of this algorithm is that it requires that the detected players' faces are frontal or near frontal. Again, when the orientation of a player changes, tracking issues can obfuscate the identification.

Broadcasted cycling videos continuously switch between cameras and viewpoints, which makes tracking even more difficult, i.e., none of the previously mentioned solutions would give us satisfying results. This is the main reason why we decided to develop a pose-based methodology that works on frontal, lateral and dorsal views, and tracks riders within the same shot when no occlusions occur. In case of occlusions, the tracking stops and the

detection algorithm tries to resolve them. Based on the pose and the type of shot, decisions between using face recognition, jersey recognition and/or number recognition are made, and available sensor data is used to further filter or verify the set of possible candidates, as explained in Section 3.

## 2.2 Cycling Sensor Data

Several third parties (such as Velon and Gracenote) provide structured and very detailed live data of sports events at a high frequency. Velon, for example, provided location, heart rate, power, and speed of each cyclist during several stages of Tirreno Adriatico 2019 and Gracenote provided exact location of each group of riders during the Tour of Flanders 2019. If such sensor data of the race is available, it is definitely the most accurate and computationally most interesting solution for geo-localization and event detection. When there are multiple groups of riders, however, an additional method (such as team or cyclist recognition) is needed to know which particular group is shown in the live video stream (as is further discussed in Section 3).

If detailed sensor data were available, several events can be detected in it, such as breakaways, crashes, or difficult sectors (e.g. barriers and sandpits in cyclocross or gravel segments in road cycling). For the latter type of events, the approach of (Langer et al, 2020) for difficulty classification of mountainbike downhill trails can, for example, be tailored to cyclocross and road cycling segment classification. The work of (Verstockt, 2014) also shows that this is feasible. For breakaway/crash detection, experiments revealed that simple spatio-temporal analysis across all riders will already provide satisfying results.

## 3 TEAM & RIDER DETECTION

### 3.1 Skeleton and Pose Detection

The proposed team and rider detection methodology both start from the output of a skeleton recognition algorithm (such as OpenPose<sup>1</sup>, tf-pose<sup>2</sup> and AlphaPose<sup>3</sup>). Figure 3 shows an example of the skeleton detection (front and side view) of these algorithms – tested in our lab set-up. In order to measure the accuracy of each of the available pose estimation libraries, tests were performed in which ground truth annotations of the rider joints are

compared to the algorithms’ output. As can be seen in the results shown in Figure 4, none of these skeleton trackers is outperforming the others in all situations, but AlphaPose and OpenPose are definitely outperforming tf-pose. An evaluation on a dataset of Tour de France footage with OpenPose also provided satisfying results on the typical filming angles in cycling live broadcasts.

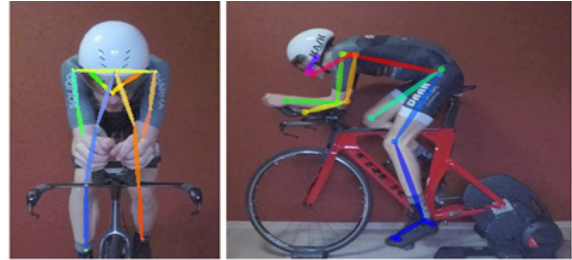


Figure 3: Rider skeleton detection (lab set-up).

The skeleton detection is providing the keypoints, i.e., the main joints of the rider’s body. From the keypoint locations (i.e., pixel coordinates) we can detect the pose and orientation of the rider. If the left shoulder is left of the right shoulder, then it is most likely to be a frontal shot. If the left shoulder is on the right of the right shoulder, then the frame was most likely shot from a rear perspective. Based on this information, different techniques can be selected for further identification. For instance, if we see a rider from the back, face detection will not work, but a combination of number and team recognition will make it possible to detect the rider from that side. If we have a frontal view of the rider, the number will of course not be visible, but now the face recognition algorithm can take over. If available, sensor data can help to limit the number of candidate riders that can be expected in a particular shot or frame.

In addition to detection of the orientation of the rider, skeleton detection can also be used for shot type classification. Based on the number of detected skeletons and their size/location in the video footage, a close-up shot can easily be distinguished from a longshot or landscape view, as is shown in Figure 5. Furthermore, scene changes can also be detected by analysing the skeleton size/location changes over time. As a result, we know exactly when it is safe to start and stop tracking (if needed) and we can also easily further crop the video into logical story units.

Finally, we also use the skeleton output to crop out the faces and upper body regions of the riders – results of this step are shown in Figure 6. In this way the and

<sup>1</sup> <https://github.com/CMU-Perceptual-Computing-Lab/openpose>

<sup>2</sup> <https://github.com/ildoonet/tf-pose-estimation>

<sup>3</sup> <https://github.com/MVIG-SJTU/AlphaPose>

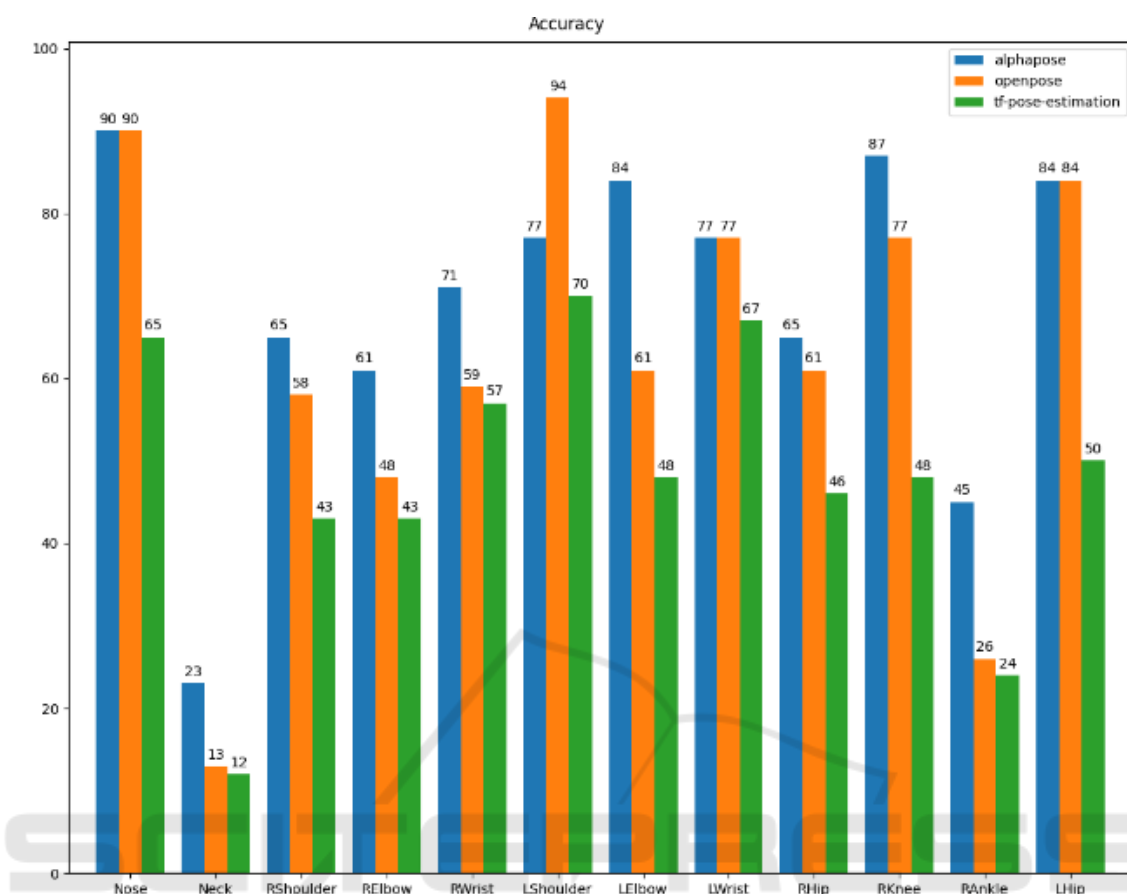


Figure 4: Accuracy of skeleton trackers for each of the rider's joints.



Figure 5: Results of pose detection and shot type estimation on a Tour de France 2019 broadcast.



Figure 6: Results of skeleton-based upper-body extraction.

accuracy of the next steps (face, number and team recognition) is improved a lot as background noise non-rider information (e.g. numbers/text of other objects) are limited to the bare minimum.

Since the skeleton detection output is used in several of our building blocks, its higher computational cost is spread across all these steps, making it a very interesting building block that can

probably be even further reused in other types of automatic sports analysis.

### 3.2 Team Recognition

The proposed team recognition algorithm is based on a CNN-powered sequential model that is developed



with KERAS/Tensorflow<sup>4</sup> and trained on a dataset of rider images from 18 UCI World Tour teams. The CNN outputs a collection of class probabilities for with KERAS/Tensorflow<sup>5</sup> and trained on a dataset of rider images from 18 UCI World Tour teams. The CNN outputs a collection of class probabilities for each rider upper body region, as shown in Figure 7 (in this example the rider is recognized as being a member of Team Jumbo-Visma with 94%). The current accuracy of the model is approximately 80%, however, the current dataset for training is still limited. By extending the dataset (e.g. by data augmentation or web scraping) and taking into account additional information (e.g. text recognition of team sponsors or jersey numbers) we should be able to detect most team jerseys with high accuracy. Furthermore, the current model does not take into account the orientation of the rider. Future work will also evaluate if a separate model for front, side and rear views would further increase the accuracy of the team recognition. Finally, based on the specific accuracies for each team (shown in Figure 8) and the confusion matrices that explain which teams are swapped most frequently, we can further improve our workflow in the future.

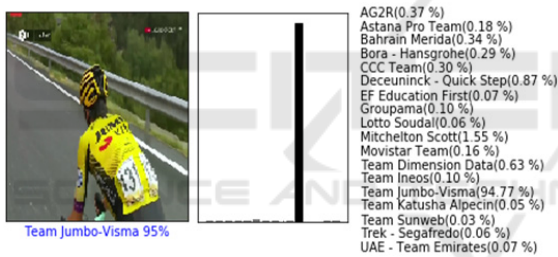


Figure 7: Results of team recognition algorithm.

### 3.3 Number Recognition

For both the dorsal and lateral view of riders we use number recognition to identify the riders. Similar to the team recognition, we start from the upper body region of the rider, which is fed to the number recognition module. Once a bib number is recognized we can link it to the name of the rider using the race contestant list. Important to mention is that this bib recognition approach is immediately applicable to other sports too.

The text extraction itself uses Microsoft Azure<sup>6</sup>, which has state-of-the-art text detection and recognition models. Azure offers an API that accepts HTTP requests. For text extraction, two separate calls

<sup>4</sup> [https://www.tensorflow.org/api\\_docs/python/tf/keras](https://www.tensorflow.org/api_docs/python/tf/keras)  
<sup>5</sup> [https://www.tensorflow.org/api\\_docs/python/tf/keras](https://www.tensorflow.org/api_docs/python/tf/keras)  
<sup>6</sup> <https://docs.microsoft.com/en-us/azure/cognitive-service/computer-vision/concept-recognizing-text>

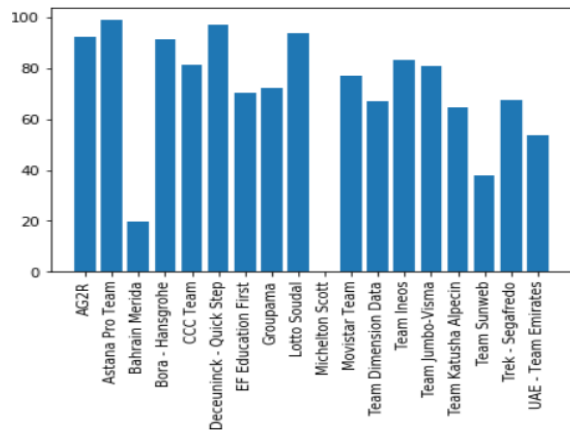


Figure 8: Accuracy of team recognition for each team.

are required: one that supplies the images to Azure and another that retrieves the text recognition results. It is important to note that multiple calls can be made asynchronously, allowing this pipeline element to be multi-threaded. An example of the information extracted by the text extraction module is shown in Figure 9. The JSON output contains the detected number, its bounding box, skeleton ID and confidence score.



Figure 9: Results of number recognition algorithm.

The biggest downside of using Azure is the fact that it is a pay-per-call service. The amount of calls to the Azure servers per hour of video is 7200 times the framerate of the video (if each frame should be analysed). However, since subsequent frames are correlated, we can limit the text analysis to a couple of frames per second, which is still resulting in a lot of calls to Azure's services. As each call is billed separately, the economical attractiveness of using commercial text-recognition APIs is rather low. Furthermore, Azure also produces better results the higher the size of the text relative to the entire frame. Accuracy would be at its highest, if we launch a call for each upper body region instead of for each frame. This, however, would even further increase the amount of calls. Thus, a number of alternatives have been explored.

A well-known text extraction alternative is Tesseract Optical Character Recognition (OCR), which is available as a Python library<sup>7</sup>. Tesseract, however, was not able to detect the majority of the bib numbers in our test sequences. Even after intensive pre-processing/optimizing the images the library still did not manage to extract the numbers. So it was decided to abandon Tesseract as a candidate to replace Azure.

Another promising OCR implementation, is that of KerasOCR<sup>8</sup> which uses the CRAFT Character Region Awareness model for Text Detection (Baek et al., 2019) and the Keras CRNN model for text recognition<sup>9</sup>. Some preliminary testing has already shown that the KerasOCR model produces similar results to those of Microsoft Azure, making it a viable replacement option. The main limitation of KerasOCR, however, is that it is even slower than Azure. However, this could be solved by multi-threading the text extraction module.

Once the text recognition module has detected the bib numbers in the frame, they are mapped onto the riders that were detected during the skeleton and pose estimation. This mapping is done by verifying whether the bib number is located on the back of a skeleton. Once this mapping is completed, the bib numbers are resolved based on the contestant info (as discussed before). To keep track of the results gathered by both the bib number resolver and face recognition throughout a sequence, a shared dictionary is used. This shared dictionary holds the most likely label (the name of the rider in question) and the confidence score of that label for each skeleton ID. The main reason the dictionary is shared is that the face recognition module and the bib number resolver module share this data structure. This means that when face recognition assigns a label to a certain skeleton, this label can be corrected afterwards by the bib number resolver. The shared dictionary effectively allows both modules to correct each other and work together rather than next to each other.

### 3.4 Face Recognition

In order to recognize the riders in the live broadcasts we first need to have a dataset with the faces of all contestants. An automatic workflow was developed to automatically create such a dataset based on the

Wikidata of the riders and their pictures on Wikimedia Commons. To extract the faces from the Wikimedia images (and from the broadcasted video frames too) we use a face detection algorithm that detects and encodes the faces and their facial features. Based on these encodings, the recognition can then be performed.

A number of different approaches to perform the face detection exist. Histogram of Oriented Gradients (HOG) based face detection (Cerna et al., 2013) is a technique that remains popular due to its low computational complexity. Although this detection method performs quite well on perfectly frontal images, it often fails when it is used to detect faces shot in a non-frontal position. CNN-based face detection is an alternative approach whose accuracy has been shown to be less prone to skewed images (Vikas, 2018). A Python version of the C++ library dlib<sup>10</sup> contains such kind of face detector and is used in our set-up. Furthermore, it also provides an encoder for facial features in images and a face landmark detection algorithm. The encodings can be used to calculate distances between faces and thus produce a metric to compare faces, while the landmarks are useful to distinguish different face expressions (e.g. suffering, nervousness or concentration).

The face encoder from dlib is a CNN heavily inspired by the ResNet34 architecture (He, 2016), with a few layers removed and the number of filters per layer halved. The model was trained on a combination of several publicly available datasets such as Facescrub and the VGG ones. A shortcoming of the scraped images from WikiMedia Commons is that some rider images also contain other people. To resolve this problem and filter out faces that do not belong to that specific rider, the pipeline applies face clustering. Based on the assumption that the rider in question will be represented more than any other person, the largest cluster should correspond to the rider's encoded faces.

The recognition step compares the encoded faces gathered during the initialization process with face encodings extracted from the video frames. The encoded face from the dataset with lowest distance is selected as the detected rider. The used distance metric is the norm of the difference of the 128 features describing the face encodings. The documentation of the face recognition library which produced the encodings states that a distance lower than 0.6 indicates that the faces are belonging to the same

<sup>7</sup> <https://pypi.org/project/pytesseract/>

<sup>8</sup> <https://keras.io>

<sup>9</sup> <https://github.com/kurapan/CRNN>

<sup>10</sup> <https://github.com/ageitgey/face-recognition>



Figure 10: Results of face recognition algorithm.

individual. However, based on our experiments, we found that a threshold of 0.4 resulted in better matching. The JSON output of the recognition step is shown in Figure 10. For each detected rider it contains the skeleton ID, rider name, and confidence score.

The proposed face recognition still has some issues due to the specific circumstances in which the recognition needs to be performed. First of all, contestants always wear helmets and often sunglasses - both covering a part of their face and thus limiting the amount of 'visible' face features. Second, cyclists have the tendency to look downward, rather than straight forward, which has a negative impact on performance as well. Finally, the video quality is sometimes rather low, which impacts the face recognition significantly. However, for professional broadcasts this should not be an issue, as the pipeline should ultimately run on their raw footage, which has sufficiently high quality. Furthermore, additional data streams (such as GraceNote<sup>11</sup> data) could be used to improve the recognition. GraceNote produces data streams which give detailed, real-time information of the state of the race. This includes information about the different groups. This means that if there is a group of two riders and the system only manages to identify one, then the other rider can still be labelled. Similarly, if the system identifies two riders in the same frame, which are in different groups, it could re-adjust its predictions.

## 4 RESULTS

The entire pipeline has been tested on both cyclocross and Tour de France race footage. An example of the cyclocross results are shown in Figure 11. Once riders are detected, they are tracked by skeleton matching between consecutive frames. In this example, all 5 riders are detected correctly. However, this is not

<sup>11</sup> <https://www.gracenote.com/scores-and-statistics/>

always the case – i.e., we still get some errors in more challenging sequences. Future work, as discussed in the conclusions section, will focus on resolving them by improving/extending some building blocks.

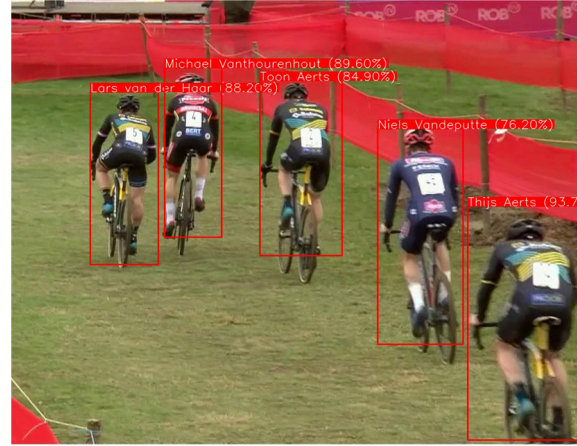


Figure 11: Cyclocross rider recognition.

As illustrated in Table 1, the text extraction and bib number resolver modules are the major bottlenecks for the proposed pipeline. A deeper analysis of the problem reveals that there is a large waiting time between the moment when the image is posted to Microsoft Azure and when the results from text extraction are available. Due to its single-threaded nature, the system currently has to wait for the results from Azure, before it can start processing a new frame. By multi-threading this piece of the pipeline, the process could be speeded up by a factor of five, as nearly 90% of the current total processing time of the text recognition module is spent actively polling the results from Azure.

Table 1: Processing time of the individual building blocks.

Module	Time	Percentage
Pose Estimation	9.64s	0.65%
Shot Boundary Detection	68.96s	4.62%
Shot Type Detection	2.9s	0.19%
Face Detection/Recognition	70.55s	4.73%
Text Extraction/Bib Number Resolver	1340.97s	89.82%

## 5 CONCLUSIONS

This paper proposes a video processing system that can be used to automatically annotate cycling race broadcasts. Skeleton-based rider pose detection is used to extract rider regions and analyse them with the team, number and/or face recognition building blocks. Evaluation on both cyclocross and road



cycling races show that there is certainly potential in this novel methodology, however, improvement is still possible.

Data augmentation could offer a possible solution to combat some of the current face recognition problems. Data augmentation is a machine learning technique in which a data sample is transformed into several versions, which should all typically be labelled the same. By augmenting images from WikiMedia Commons, for example, into versions with and without glasses, the face recognition algorithm should be able to recognize riders more easily, regardless of the fact if a racer is wearing glasses or not. To determine the location of the glasses onto the face of the contestant, pose estimation could be used to extract the location of the eyes and ears of the participant. This process is illustrated with an example in Figure 12.



Figure 12: Augmenting the face of a cyclist with glasses.

Another improvement could be switching to a more advanced face recognition algorithm, such as DeepFace<sup>12</sup> or FaceNet<sup>13</sup>. However, as these algorithms are slower than our current dlib approach and they would require some adjustments for real-time processing. A final suggestion which could improve the accuracy of the face recognition module, would be to use a spatio-temporal face detection model. The current model uses a frame-by-frame approach, meaning it does not fully exploit the temporal correlation between subsequent frames.

Finally, OCR-based recognition of sponsor names, as shown in Figure 13, could definitely help to further improve the team recognition and will be investigated in future work too.

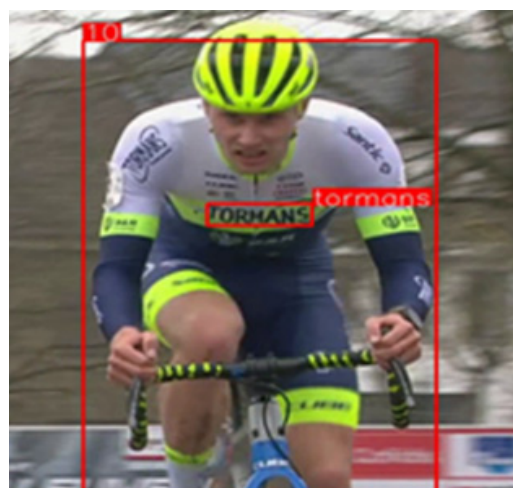


Figure 13: OCR-based recognition of sponsor names.

## ACKNOWLEDGEMENTS

This research is funded by the imec ICON project DAIQUIRI (Agentschap Innoveren en Ondernemen project nr. HBC.2019.0053).

## REFERENCES

- Lombardo, J., Broughton, D., 2017, Going gray: Sports TV viewers skew older, *Sports Business Journal* (June 5<sup>th</sup>).
- Qian, X., Wang, H., Liu, G., Hou., X., 2012, HMM based soccer video event detection using enhanced mid-level semantic, *Multimedia Tools Appl.* 60(1), 233-255.
- Tejero-de-Pablos, A., Nakashima, Y., Sato, T., Yokoya, N., Linna, M. and Rahtu, E., 2018, Summarization of User-

<sup>12</sup> <https://github.com/serengil/deepface>

<sup>13</sup> <https://github.com/davidsandberg/facenet>



- Generated Sports Video by Using Deep Action Recognition Features, *IEEE Transactions on Multimedia*, 20(8), pp. 2000-2011.
- Liu, H., Bhanu, B., 2019. Pose-Guided R-CNN for Jersey Number Recognition in Sports. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.
- Mahmood, Z., Ali, T., Khattak, S., Hasan, L., Khan, S., 2015. Automatic player detection and identification for sports entertainment applications. *Pattern Analysis and Applications*. 18. 10.1007/s10044-014-0416-4.
- Langer, S., Müller, R., Schmid, K., Linnhoff-Popien, C., 2020. Difficulty Classification of Mountainbike Downhill Trails Utilizing Deep Neural Networks. *Machine Learning and Knowledge Discovery in Databases*. 10.1007/978-3-030-43887-6\_21.
- Verstockt, S., Slavkovikj, V., De Potter, P. and Van de Walle, R., 2014, Collaborative Bike Sensing for Automatic Geographic Enrichment: Geoannotation of road and terrain type by multimodal bike sensing, *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 101-111, 10.1109/MSP.2014.2329379.
- Baek, Y., Lee, B., Han, D., Yun, S, Lee, H., 2019, Character Region Awareness for Text Detection, IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Cerna, L., Camara-Chavez, G., Menotti, D., 2013. Face detection: Histogram of oriented gradients and bag of feature method, *International Conference on Image Processing, Computer Vision, and Pattern Recognition (ICCV)*.
- Vikas, G., 2018, Face Detection – OpenCV, Dlib and Deep Learning (C++/Python) <https://www.learnopencv.com/face-detection-opencv-dlib-and-deep-learning-c-python>
- He, K., Zhang, X., Ren, S., Sun, J., 2016, Deep Residual Learning for Image Recognition, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 10.1109/CVPR.2016.90.