Supplementary material
ARTHuS: Adaptive Real-Time Human Segmentation in Sports through Online Distillation

A. Cioppa[1,*], A. Deliège[1,*], M. Istasse[2], C. De Vleeschouwer[2] and M. Van Droogenbroeck[1]
[1]University of Liège, [2]University of Louvain, Belgium
[*]These authors contributed equally
{anthony.cioppa,adrien.deliege}@uliege.be

Code and video available at https://drive.google.com/drive/folders/1FFdZYel3s8tL5YgLc6EQyZObRg2AMpDo?usp=sharing

A. Description of the datasets

In this section, we provide further details about the datasets used for training, validating and testing the offline distillation process that produces the networks $\mathcal{S}_{\text{pretrained}}$ of Section 3.2.

We use the main camera stream for both the soccer and basketball videos. This camera has a wide angle of view and is shown most of the time on television because it usually provides an excellent overview of the ongoing match. Hence, this camera is often used for sports analysis. Figure 4 of the main paper shows four examples of images taken from this camera.

Regarding the soccer dataset, we use the following eight matches from the UEFA Euro 2016 competition: 1. Germany vs Slovakia; 2. Belgium vs Wales; 3. Croatia vs Portugal; 4. France vs Ireland; 5. Northern Ireland vs Wales; 6. Poland vs Portugal; 7. Switzerland vs Poland; 8. Hungary vs Belgium. We also use one match from the 2013 Belgian Jupiler Pro League, FC Bruges vs Anderlecht, in order to test the networks on a match from a different competition, for the reasons detailed in the paper.

Regarding the basketball dataset, we use the following eight matches from the 2019 LNB Jeep Elite competition: 1. Bourg-en-bresse vs Cholet; 2. Le Portel vs Monaco; 3. Lyon-Villeurbanne vs Cholet; 4. Le Mans vs Châlons-Reims; 5. Gravelines-Dunkerque vs Le Portel; 6. Landerneau vs Montpellier; 7. Dijon vs Strasbourg; 8. Cholet vs Boulazac.

B. Obtaining $\mathcal{S}_{\text{pretrained}}$

The two video datasets are used for training two separate instances of $\mathcal{S}_{\text{pretrained}}$, one for each sport, by usual offline knowledge distillation. We collect a set $\mathcal{X}$ of images by selecting one frame every four seconds in each video and compute their corresponding approximated ground truth $\mathcal{T}(\mathcal{X})$ using the teacher network $\mathcal{T}$. We split the dataset into three sets: a training set $\mathcal{D}_{\text{train}} = (\mathcal{X}_{\text{train}}, \mathcal{T}(\mathcal{X}_{\text{train}}))$ containing the images subsampled from the first six matches, a validation set $\mathcal{D}_{\text{val}} = (\mathcal{X}_{\text{val}}, \mathcal{T}(\mathcal{X}_{\text{val}}))$ containing the images of the seventh match and a test set $\mathcal{D}_{\text{test}} = (\mathcal{X}_{\text{test}}, \mathcal{T}(\mathcal{X}_{\text{test}}))$ containing the images of the eighth match. $\mathcal{S}_{\text{pretrained}}$ is trained on $\mathcal{D}_{\text{train}}$ using the Adam optimizer with a batch size of 1, the weighted cross entropy loss (see pytorch.org) and a learning rate of 10^{-4}. We stop its training when its performances on $\mathcal{D}_{\text{val}}$, computed after each epoch, start decreasing. The good performances of $\mathcal{S}_{\text{pretrained}}$ on an unseen game, assessed on $\mathcal{D}_{\text{test}}$, confirmed that $\mathcal{S}_{\text{pretrained}}$ could be used as such for the experiments reported in the paper.

C. About the networks

This section provides some details about the implementations of the networks used in the paper.

TinyNet [1] This network is the real-time segmentation network used for most of the experiments of the paper. We adapted the original implementation available at https://orbi.uliege.be/handle/2268/222427 from TensorFlow to PyTorch. In our work, we train it on fine-grained segmentation masks (provided by $\mathcal{T}$) rather than on rectangular blobs (manually annotated). The achitecture of the network can be found in Figure 1. One way to speed up the inference time, is to compute the segmentation on batches of images rather than one image at a time, authorizing a small delay of on the output stream. Let us note that real-time inference is still possible with batches of size 1.

ICNet [4] This well-known network, state-of-the-art among real-time networks on the Cityscapes dataset [2], is

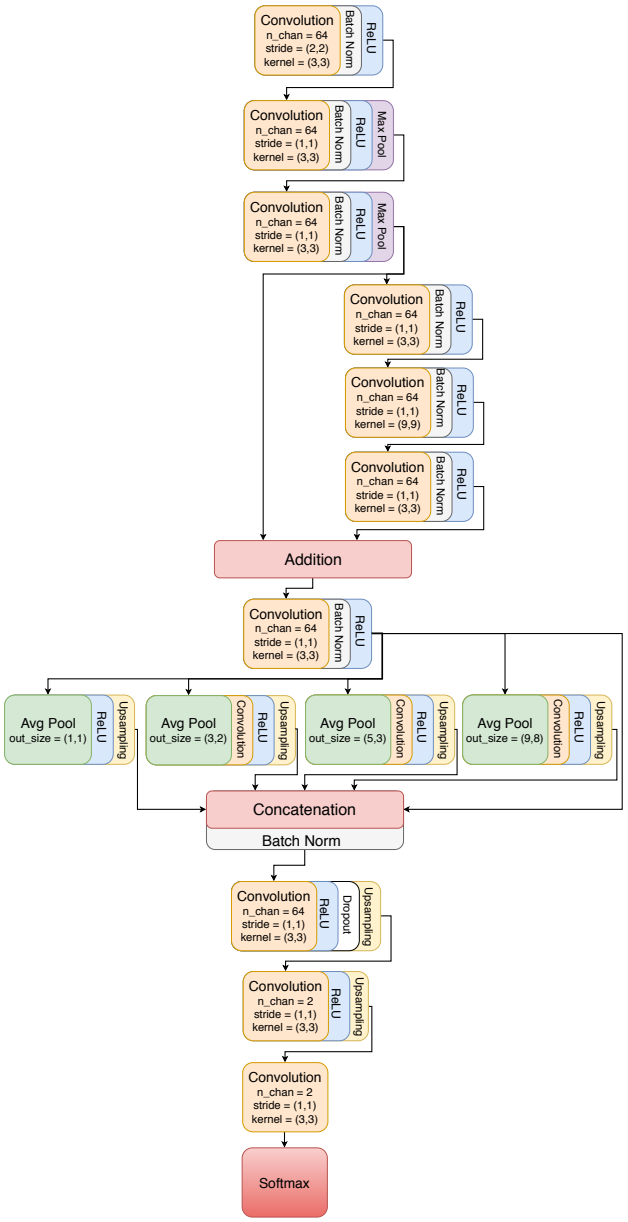

Figure 1. Architecture of the real-time segmentation network TinyNet [1]. It is a lightweight version of PSPNet 101 with 100 times less parameters.

used for a comparison with TinyNet. We adapted the implementation available at https://github.com/hellochick/ICNet-tensorflow from TensorFlow to PyTorch. By default, this network outputs predictions for a large variety of classes. Since we only have two classes of interest in this work, humans or background, we modified the last layer of the network so that it outputs two numbers for each pixel, after which a softmax is applied, as in TinyNet.

Mask R-CNN [3] This network is chosen as the slow but effective "universal" teacher network $\mathcal{T}$ of our experiments. We used the PyTorch implementation available at https://github.com/facebookresearch/maskrcnn-benchmark. This network outputs several bounding boxes predictions with corresponding labels and segmentation masks inside the boxes. In order to select only interesting humans for our online training process, we keep the segmentation masks provided inside the boxes whose label corresponds to "human" and that intersect the field mask.

D. Additional experiments

D.1. Analysis of a "failure" case

It can be seen in Figures 3 and 6 of the paper that a drop in F_1 score occurs around $15-20$ minutes in the soccer test match and 20-25 minutes in the basketball match. We further comment this "failure" case in this section, as one might interpret it as a possible limitation of ARTHuS.

By looking at the soccer video at that moment, it can be seen that an unusual scene occurs, which shows a remote part of the field where bear-looking mascots walk and photographs, staff members and spectators sit, as depicted in Figure 2. Fortunately, it appears that $\mathcal{S}^{\text{seg}}$ still perfectly segments the player and the referees, which are the only humans that are of interest in this scene. Hence, the drop in performances can be explained by the large number of false positives and false negatives related to the mascots and the other people external to the game on the bottom and right side of the frame, which we added in the corrected masks (and for which $\mathcal{T}$ is also confused). If we evaluate the F_1 score without taking them into account, it goes back up to the same levels of performance as those achieved for the rest of the match. For the baskeball video, the scene is a timeout, in which the players are grouped together and discuss strategies, as can be seen in Figure 2, which is a difficult case to handle for Mask R-CNN (similarly to Figure 5 of the paper).

Therefore, we can say that the drops in performances are due to unusual scenes coupled with the evaluation method itself, rather than an actual struggle of $\mathcal{S}^{\text{seg}}$ to segment the interesting humans of the scene.

D.2. ARTHuS when $\mathcal{S}_{\text{pretrained}}$ already generalizes well

As mentionned, $\mathcal{S}_{\text{pretrained}}$ is trained on six matches from the Euro 2016 competition in the case of soccer. It is interesting to check if this network generalizes well to another game of the same competition and to examine the effect of training it online with ARTHuS. To do so, we evaluate the online training with TinyNet and report the performances on the eighth match of the soccer dataset in Figure 3. As can be seen, all curves reach about the same performances

Figure 2. Frame taken around the 20-th minute of the soccer test match (top) and the 24-th minute of the basketball test match (bottom), used to analyze the slight drops in the F_1 score at these moments. They can be explained by the unusual nature of the scene, which involves mascots and spectators close to the field in the case of soccer and a time-out in the case of basketball, and the evaluation method itself, since the humans of interest in this scene, *i.e.* the player and the referees, are still correctly segmented by $\mathcal{S}^{\text{seg}}$.

with a slight advantage for those produced by ARTHuS after 15 minutes. This indicates that, even when the network generalizes well, there is an interest in retraining it online since the performances can only increase during the match.

D.3. Tuning of the learning rate

As for any gradient-based learning algorithm, the learning rate may influence the results. Hereafter, we compare the results obtained with two learning rates when TinyNet is retrained online from a pre-trained network. As can be seen in Figure 4, a higher learning rate rapidly leads to better performances but also involves much more drops of performances throughout the match. This is why we use the lower learning rate of 10^{-5} in the experiments of the paper.

D.4. Other camera views

We also tested our method on another camera view. In this section, we show the result on one of the close-up cameras. This type of camera shows the players from a close-up point of view, which results in much bigger silhouettes compared to the main camera. Figure 5 shows the result of our method when trained from scratch after 20 minutes of

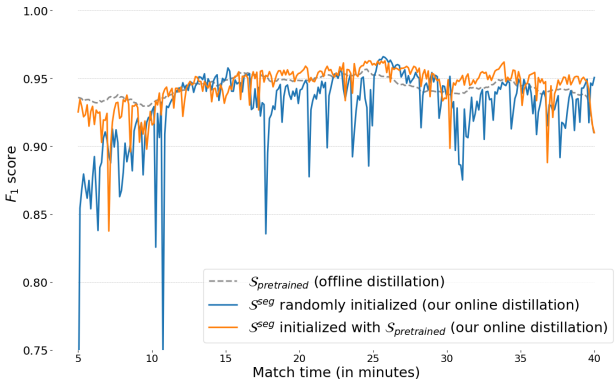

Figure 3. Evolution of the performances of several variants of distilled models through their F_1 score, computed with respect to the masks provided by $\mathcal{T}$, for a soccer match that is taken from the same competition as the training set. All curves reach approximately the same performances, with a slight advantage for adaptive networks produced by ARTHuS when initialized from $\mathcal{S}_{\text{pretrained}}$.

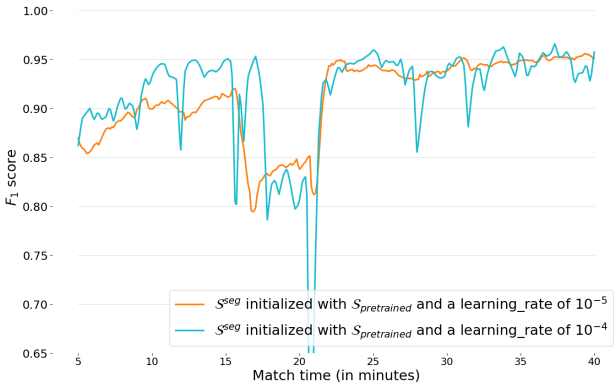

Figure 4. Comparison of two different learning rates for ARTHuS when the network is trained online from a pre-trained network. A higher learning rate allows to reach better performances rapidly but is more prone to accidental drops of performances.

online training time. As can be seen, the players are still correctly segmented, which indicates that our method can be applied as is with different camera views. This is an encouraging result about the robustness of our method to various sports scenes. It also implies that there might be no need to manually annotate any frames, regardless of the camera, in order to have a working multi-camera system for human segmentation in sports.

References

[1] A. Cioppa, A. Deliège, and M. Van Droogenbroeck. A bottom-up approach based on semantics for the interpretation of the main camera stream in soccer games. In *Int. Workshop on Comput. Vision in Sports (CVsports), in conjunction with CVPR*, pages 1846–1855, Salt Lake

Figure 5. Results of our method on another soccer camera obtained with TinyNet trained from scratch.

City, UT, USA, June 2018.

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Int. Conf. Comput. Vision and Pattern Recogn. (CVPR)*, pages 3213–3223, Las Vegas, NV, USA, June 2016.

[3] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2018.

[4] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia. ICNet for real-time semantic segmentation on high-resolution images. In *Eur. Conf. Comput. Vision (ECCV)*, volume 11207 of *Lecture Notes Comp. Sci.*, pages 418–434, 2018.