Self-supervised Learning of Spatial Object Positioning in Football

Matúš Baran*
Supervised by: Igor Jánoš†

Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Bratislava / Slovakia

Abstract

We introduce a pretext task for self-supervised learning of feature extraction on an unlabeled dataset of football images. The task is based on predicting the relative distance between two random crops from the same image, which requires the model to understand the spatial positioning of the objects and players in the image. We evaluate the feature extractor trained with the proposed pretext task on the SoccerNet action spotting challenge and compare it to the existing self-supervised method SimCLR. We demonstrate the effectiveness and generality of the proposed pretext task for learning relevant features of the football domain.

Keywords: Self-supervised, Feature extractor, Football

1 Introduction

Football arguably belongs among the most favorite sports in the world with millions of fans and players. With technological advances and improvements in machine learning algorithms, the tasks performed by humans have been automatized and simplified and this applies also to the football domain. There were many attempts to create a model that would understand the game to predict the winner [31, 32, 2], analyze the players [24, 23], or even substitute the role of a referee [3].

The recent works in self-supervised learning methods made huge advances in the field of computer vision by closing the gap to supervised learning [20], some of them even surpassing the supervised method [5]. The self-supervised methods like MoCo [21] and MoCov2 [8] proved to be very effective in extracting relevant features from the image by contrasting the features. Other works showed that the missing annotations in the dataset can be replaced by introducing a pretext task such as image rotation [19] or temporal frames shuffling [27]. The purpose of the pretext task is to force the model to learn relevant features on the prior layers that can then be transferred to other downstream tasks.

*xbaranm@stuba.sk
†igor.janos@stuba.sk

We introduce a pretext task for self-supervised feature extractor learning on the unlabelled dataset. The task is based on the spatial understanding of the image and does not rely on the batch size. We apply this task by training a feature extractor for the football domain on the unlabeled dataset and validate it by transferring the trained model to the downstream football task.

We consider the action spotting challenge from SoccerNet [15] as an appropriate task to evaluate our feature extractors. The goal of the task is to identify 17 football actions like a goal, foul, ball out of play, etc. in broadcasted football videos. The task allows us to exchange the used feature extractor while preserving the rest of the solution architecture. So by substituting the feature extractors, we can evaluate them with the resulting performance of the task.

To show the effectiveness of our method we compare the lightweight feature extractor model trained with our pretext task to the lightweight model trained with the existing self-supervised method SimCLR, and also to a bigger pre-trained model with substantially more parameters.

Our contributions are as follows:

- We introduce a pretext task based on the spatial understanding of the image content by predicting the relative distance between two random crops for the self-supervised learning of the feature extractor.
- We trained multiple feature extractors using the existing self-supervised method SimCLR and our method which we evaluated and compared using the SoccerNet action spotting challenge.

2 Related work

Many previous works focused on creating a pretext task that would replace the missing annotations. Noroozi and Favaro [28] created a pretext task inspired by the puzzle game jigsaw in which the original image is divided into nine evenly big crops and shuffled. The goal of the model is to solve the jigsaw puzzle by which the model learns features that are as representative and discriminative as possible.

Another pretext task which is based on the nine-part grid is defined as predicting a relative position of the crops

[14]. The nine crops are taken from the original image while preserving the grid structure with a little variance. The model is always given the central middle crop with one of the eight remaining neighbor crops. The model then has to predict the relative position of the second crop by specifying one of the eight directions represented by the numbers one to eight. It is therefore a classification task where only one option is correct. The distance between the crops is always relatively small as the crops are next to each other in contrast to our method where the crops are randomly sampled. This prevents learning features that are spread along the whole image from one end to another.

The contrastive methods SimCLR [6] and SimCLRv2 [7] rely on attracting the positive pairs represented by augmented views from the same image and repelling the negative pairs represented by augmented views from different images. This is done by applying the contrastive loss on the features extracted from the views while maximizing the similarity of the features from positive pairs and minimizing the similarity of negative pairs. The effectiveness of this method highly relies on big batch sizes which require adequate computational power and resources. As Lin et al. mentioned [26], there are cases where negative pairs from different images can be more similar than the positive pairs from the same image. For example, the two crops from opposite corners of the same image can both capture diametrally different content, and forcing them to have similar feature representations could be misleading.

Giancola et al. [15] proposed a benchmark dataset for football action spotting. Later the authors extended the SoccerNet dataset [11] and provided a baseline using their own NetVLAD++ [18] model. The authors provide the annotated dataset along with annual challenges [16, 10] doing which they promote the use of neural networks in the football domain.

Action spotting is a challenge to identify certain football actions within the temporal window of their occurrence in the video. It is a popular challenge with many submissions [30, 22, 13, 9] competing for the best result. We consider the action spotting task as the appropriate form of evaluation of our feature extractor as it focuses on the most interesting and common actions in football.

3 Data collection

Despite the recent advances in football dataset annotation [17], manual annotations are still needed. Therefore we decided to attack the problem of insufficient size and number of annotated datasets in the football domain by using a self-supervised method and train the model on unlabeled football data. As the process of annotating is often costly and always very time-consuming, there will be no need for the dataset to contain the annotations. In this case, we trade off the missing annotations for a larger dataset size.

When training an unsupervised or self-supervised model, a large dataset is a must. Therefore getting as much valid data as possible was our top priority. We focused on the replays of professional football matches and extracted the frames from these videos. Football is a dynamic sport where a lot can happen in a nick of time so we choose the frequency of the extraction to be two frames per second. This resulted in the final 12,085,293 images in the unlabeled dataset. As the main source of the videos was YouTube, we named the dataset YF (YoutubeFootball).

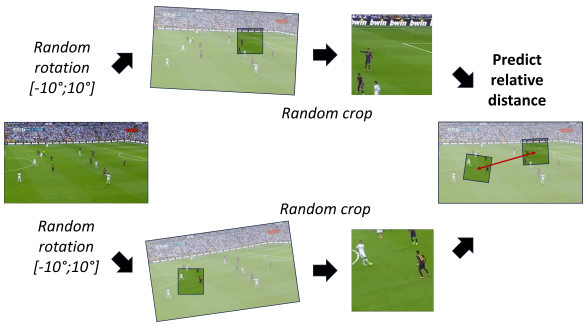

Figure 1: Illustration of our pretext task that is used for self-supervised training.

The images in the dataset do not strictly have to be consecutive as there is no additional information about which image is the start or the end of some video. So image *N + 1* does not have to be subsequent to image *N*. This fact constrains the pretext task to not rely on any temporal information which makes the pretext task more generic and applicable to other domains.

As we do not possess the author rights to the videos we can only publish the scripts for the image extraction and not the whole dataset.

4 Our pretext task

Most of the existing methods are trained and benchmarked on datasets [12, 25] that have very little in common with football. The fact that the YF dataset consists of football images only can be used as an advantage when creating the new self-supervised method.

Our pretext task focuses on understanding the spatial positioning of the objects in the images by predicting their relative distance from each other. This is done by extracting two random crops of the same size from the same image and measuring the relative distance between their centers.

Before the crops are taken from the image, the image is rotated by a random degree. For each crop, a new random number is used from the interval from -10 to 10 degrees. The rotation is done around the center of the image. After the rotation is applied, the resulting image is still rectangular, but a dark background is created to fill the blank spaces around the rotated edges. To end up with the image containing only the valid content of the image a crop that represents the largest possible rectangle that excludes the

Figure 2: After the rotation is applied, corner spaces around the image are filled with default dark color. To end up purely with valid data containing the content of the image a crop is performed, representing the largest possible rectangle with content omitting the filled spaces created by rotation.

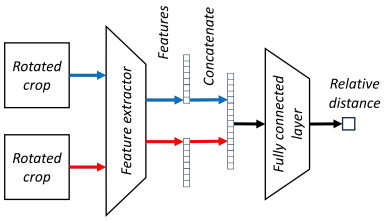

Figure 3: Architecture of our pretext task. The features are concatenated into the dense layer which outputs the relative distance.

dark background from the rotation needs to be performed. To better understand this process, figure 2 visually shows the adapted solution to effectively end up with valid data after augmenting the original image.

After the images are rotated and cropped to contain the biggest possible content, random coordinates are selected to represent the center of the final crop in each of the rotated images. The range that the coordinates are taken from is calculated so that the randomly taken coordinate is not located near the edge of the image which would result in an incomplete image crop since the part of the crop could exceed the rotated image. This technique ensures that the final crop will always contain valid data. On the other hand, the rotation of the image and the aforementioned cropping result also in omitting some valid parts of the original image that will not be used when performing the final crops. While this is true in most cases, in a case when the rotation degree is zero the full image is available for the final crop and no data is omitted before the final crop.

Since both of the crops are extracted from the images that could be rotated by a different degree, the coordinates of their centers are recomputed to match the exact same points in the original non-rotated image. The relative distance between the crops is computed as the distance between the centers of the crops divided by the size of the crop, all in pixel units. So if two crops were both from the non-rotated images(rotation angle zero degrees) and were right next to each other meaning they have one common edge, their relative distance would be exactly one.

This relative distance is created for every image during the training so the pseudo-labels are created on the fly and the task for the model is to predict this relative distance. Figure 1 illustrates our pretext task and figure 3 illustrates the architecture of the model using our pretext task.

Our method is different from the previous position prediction pretext task [14] as it offers more variations in the resulting pseudo-label because the predicted value is not limited by some set of values. Also, the crops are taken randomly and there is no restriction on their positioning, meaning that they can be next to each other, or one under the other, or anywhere in the image. The gap between them also varies so the model must learn not only local similarities when the crops are right next to each other but also be aware of the global context when the crops are on the opposite corners of the image. There is also no restriction on whether the crops can overlap or not.

To be able to accurately predict the distance between two parts of the image some knowledge about the context must be known that can be derived from the content of the two crops. In football, the positioning on the pitch is very important. It can say a lot about the style of the play of one team or the current situation in the game, whether is the team attacking or defending. The positioning of the players is very important also because of the football rules. Mainly because of one particular rule, which is offside [4]. In football, a player is offside if they are closer to the opponent's goal line than both the ball and the second-last opponent when the ball is played to them. So the understanding of the positioning is even more important in the football domain.

Therefore using our proposed pretext task the model should learn to understand the complex positioning of the players and the ball on the pitch. This however applies not only to the game itself but also to replays from other perspectives and other actions connected to the game as substitutions, in-game medical treatment, and many more. So when the model is trained to be relatively accurate when predicting the relative distance between two crops from the football image, it must possess some deeper knowledge and understanding of the football positioning itself. This implies selecting more valuable features from the early layers and in the end better feature extraction.

To make the task more challenging the rotation of the image by up to ten degrees is applied. When looking at the images in the figure 2 we can visually understand what

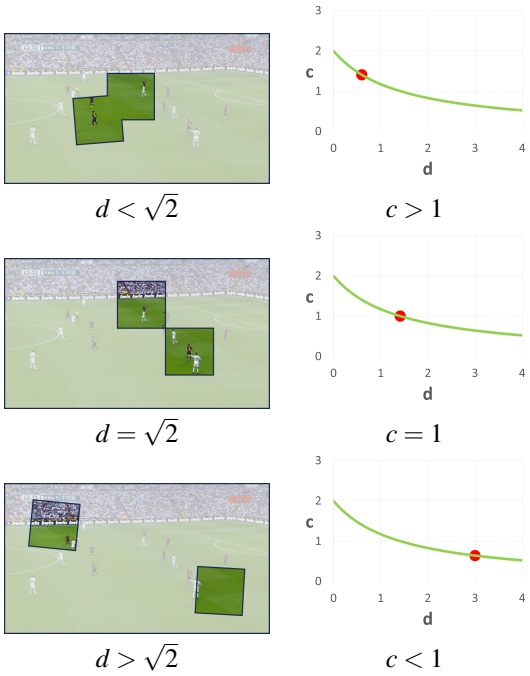

Figure 4: Illustration of possible placement of the crops, where d represents the distance and c represents the coefficient.

is happening in the picture even if it is slightly rotated. In convolutional neural networks, however, even a slight rotation can cause different outcomes as the convolutional filters are very sensitive to rotated input [29, 1]. By rotating the input crops, this sensitivity is attacked and the neural network is forced to learn and understand the images more in a way that humans understand them.

Another advantage of our pretext task is its generality. Since it does not rely on any particular information related directly to football it can be applied to other datasets as well.

4.1 Customised loss function

Predicting the relative distance of the crops from the image can be a demanding task when the crops are from opposite parts of the image since the content on one side could be wholly different than the content on the other side. There could be not so many if any clues in the crops for predicting the right distance between such crops. On the other hand, it is much easier to predict the distance if the crops are overlapping and have some common parts. The parts in common could hint that the crops are close to each other and by the size of the overlapping part, it could easily be determined how far away are the centers of the crops.

Because it is not always the same difficulty to predict the distance of the crops based only on the content of the crops without any context, we scale the loss calculated from the predicted distance based on the distance of the crops. If the crops are close to each other or even overlapping, the

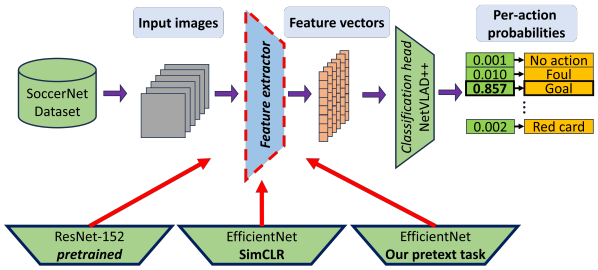

Figure 5: Action spotting pipeline with various feature extractor models. The classification head predicts the per-class probabilities for each action.

neural network should easily determine their relative distance and therefore it will be additionally penalized if it makes a mistake in such an "easy" case. If the crops are far away from each other it is way more difficult to predict the exact distance between the crops and therefore if the neural network makes a mistake in such a "hard" case the resulting mistake will be reduced.

Figure 4 shows three scenarios that can occur when creating the crops from the image. In the first case, the crops are overlapping and their relative distance is less than $\sqrt{2}$. In the second case, the crops have exactly one corner in common and their relative distance is equal to $\sqrt{2}$. In the third case, the crops have no area in common and their relative distance is greater than $\sqrt{2}$.

To adjust the loss or the mistake that the neural network makes, a coefficient is used which is calculated with the formula 1. The coefficient is dependent on the distance of the crops. The α and β are coefficients with default values $\sqrt{8}$ and $\sqrt{2}$ respectively and d is the relative distance of the crops. The final loss (l) is calculated as the product of the distance error (e) and the coefficient (c) as can be seen in the formula 2. The smaller the distance between the crops is the bigger the coefficient is and therefore the final loss will be also bigger. When the distance is bigger, the coefficient and the final loss will be smaller.

$$c = \frac{\sqrt{\alpha}}{d + \sqrt{\beta}} \tag{1}$$

$$l = e * c \tag{2}$$

The default values for the coefficients are set according to the illustration in the figure 4. In the second case when the crops are diagonally next to each other, the computed coefficient will have value 1 and therefore will not affect the final loss. This serves as a reference scenario where it should be reasonably difficult to predict the distance of the crops. If the crops are closer to each other, the coefficient will be greater than 1 and if the crops are further from each other the coefficient will be less than 1.

5 Evaluation metric

To be able to evaluate the trained feature extraction models and compare our method to the existing self-supervised method we need a qualitative metric that will give us some score for both models. As the pretext tasks used in these methods were different we could not use the training or validation loss as a valid metric for comparison. Instead, we used the action spotting task from SoccerNet which takes the broadcast videos of professional football matches and evaluates the precision of identifying specific football actions.

The model must predict the exact timestamp when the action occurs and the prediction must land within a tolerance δ around the ground truth anchor. The tolerance varies from 5 to 60 seconds with 5-second steps. Recall, precision, and Average Precision (AP) are computed for each given class and a mean Average Precision (mAP) is computed across all classes. An average-mAP is computed across all δ tolerances. The average-mAP metric, together with the *average-mAP visible* for visible actions and *average-mAP unshown* for actions that happen out of the camera range, are used for the evaluation of the models' performances in the SoccerNet action spotting.

The process of training a classification model for action spotting is illustrated in figure 5. The process consists of extracting the features from the videos and training the classification head on the extracted features. The feature extraction is a separate process that allows for modifying it by substituting the feature extractor which then yields different feature vectors.

No architectural or other changes are needed for the classification head which is every time trained from scratch on the given features. The result achieved by the classification head therefore relies on the extracted features. So when the result of a classification head trained on features extracted by one model is better than the result of a classification head trained on features extracted by the second model, we can say that the first feature extraction model is better than the second.

Figure 5 shows the integration of feature extractors trained with the SimCLR method and also our pretext task into the SoccerNet training pipeline. The values of per-action probabilities on the output of the two figures are illustrative but they symbolize that the classification head trained on different features yields different predictions which end up in different accuracy and precision.

By comparing the average-mAP of the classification heads, which is the metric used in SoccerNet action spotting, we were able to compare the performance and ability of the feature extractors to extract the relevant features from the football videos. As the architecture of the classification head remains always the same, its average-mAP is used as the qualitative metric for evaluating the feature extractors.

6 Results

We trained multiple feature extractors on the YF dataset using the existing self-supervised method SimCLR and our pretext task. The augmentations used in the SimCLR method were random horizontal flip, random resized crop, color jitter, random grayscale, and Gaussian blur, similar to the original paper. We trained the SimCLR models with a learning rate of 10^{-4}, batch size of 120, and cosine annealing scheduler without restart. We used the NT-Xent loss with a temperature of 0.7.

As for our pretext task, we used the same batch size as with SimCLR, but we used a constant learning rate of 10^{-4} together with adjusted MSE loss as discussed in the section 4.1.

The training time of one feature extractor trained on the subset of the YF dataset was about one week for both the SimCLR and our method. Training on the whole dataset took two to three weeks for each feature extractor. All training runs were executed using one NVIDIA RTX3090 GPU.

Throughout the training, we performed evaluations on the SoccerNet action spotting task, which we used as a metric for the evaluation of feature extractors for the football domain.

As can be seen in table 1, the feature extractors trained on the YF dataset did not outperform the pretrained feature extractors from the ImageNet, however, the feature extractor trained with the SimCLR method did not get behind by much, as the difference between the best model is only 2.71%. The feature extractor trained with our pretext task did not perform badly neither. The a_mAP score of 41.13% did prove that the method helps to learn to extract relevant features for the football domain, however, it does not reach the level of the existing self-supervised method. Further research focusing on finding the optimal hyperparameters of our pretext task could improve its performance.

In table 2 we show the per-class results of the NetVLAD++ model trained on features extracted by the best extractor trained with the SimCLR and our pretext task. For reference, we show also the results of the model trained on the features provided by SoccerNet that were extracted using ResNET-152 and features extracted with the pretrained Efficient-B0 model. The table shows that the model trained on the features extracted with the SimCLR model outperformed the pretrained EfficientNet-B0 in 4 classes and even outperformed the pretrained ResNET-152 in one class. The feature extractor trained with our method did not reach the best result in any class and the result margins were similar to the a_mAP results.

7 Future work

Our pretext task uses the customized loss function that contains two hyperparameters *alpha* and *beta*, which in-

Backbone	# params	Dataset	Train method	# Images seen /	unique	Head	a_mAP all	a_mAP visible	a_mAP unshown
ResNET-152*	60M	ImageNet	*pretrained*	- /	-	NetVLAD++	**52.73**	**59.07**	**36.59**
EfficientNet-B0	5.3M	ImageNet	*pretrained*	- /	-	NetVLAD++	52.17	58.79	35.61
EfficientNet-B0	5.3M	YF[000-002.sqsh]	SimCLR	110 625 000 /	375 000	NetVLAD++	**49.02**	53.87	**33.38**
EfficientNet-B0	5.3M	YF	SimCLR	69 806 310 /	11 634 385	NetVLAD++	48.84	**54.63**	33.00
EfficientNet-B0	5.3M	YF[000-002.sqsh]	**Our pretext**	56 625 000 /	375 000	NetVLAD++	39.13	44.74	29.96
EfficientNet-B0	5.3M	YF	**Our pretext**	255 956 470 /	11 634 385	NetVLAD++	41.13	46.14	30.41

Table 1: Best results of various feature extractors. The first run(marked with an asterisk *) is executed on the provided features from the SoccerNet. Other runs are executed with the use of a smaller model EfficientNet-B0. YF[000-002.sqsh] represents a small subset of the YF dataset.

Feature extractor	**# params**	**SoccerNet-v2**	visible	unshown	Ball out	Throw-in	Foul	Ind. free-kick	Clearance	Shots on tar.	Shots off tar.	Corner	Substitution	Kick-off	Yellow card	Offside	Dir. free-kick	Goal	Penalty	Yel. → Red	Red card
ResNET-152 [pretrained]	60M	52.7	59.1	36.6	74.9	58.5	73.4	69.2	36.0	39.2	40.0	56.7	70.1	68.5	64.5	43.8	57.9	79.7	54.9	3.9	5.2
EfficientNet-B0 [pretrained]	5.3M	52.2	58.8	35.6	70.2	58.7	66.7	67.3	38.7	36.5	40.0	54.2	68.7	67.5	63.2	44.3	58.4	80.9	62.8	5.4	3.4
EfficientNet-B0 [SimCLR]	5.3M	49.0	53.9	33.4	62.4	51.8	65.6	69.1	29.0	36.6	39.0	56.3	68.9	63.6	61.9	43.5	52.0	79.3	39.3	13.3	1.7
EfficientNet-B0 [Our pretext]	5.3M	41.1	46.1	30.4	36.0	39.6	49.5	60.3	22.0	33.6	36.6	52.2	66.3	58.9	54.0	38.3	39.7	77.0	33.6	0.6	1.0

Table 2: Mean average precision of the NetVLAD++ model on features extracted by various feature extractors on SoccerNet action spotting.

fluence the training process of the model. Initial values of these hyperparameters that were used are not optimized and future work could include finding the optimal values of these hyperparameters, which could lead to better performance of the method. Identically the optimal value for the maximal rotation of the image could improve the method and lead to better results.

The comparison of a lightweight feature extractor trained with our pretext task to the existing self-supervised method and a substantially bigger model showed the potential of our method. Models with more parameters tend to achieve better results because of their higher learning capacity, so possible future work includes training a bigger feature extractor using our method and comparing it to the ResNET-152 originally used in SoccerNet.

Since our pretext task does not rely on any information about the dataset or the football domain, it can be used in other domains and downstream tasks as well. An example can be the replay grounding task from SoccerNet which also uses extracted features from the SoccerNet dataset to identify replayed actions in broadcasted football matches or another non-football-related downstream task such as image classification benchmark in ImageNet.

All feature extractors in this work were evaluated on the SoccerNet action spotting using only the NetVLAD++ classification head. Using other models as a classification head can yield even better results in the SoccerNet action spotting challenge.

8 Conclusion

In this paper, we introduced a pretext task for self-supervised learning on an unlabelled dataset that is based on the spatial understanding of the image content. We trained multiple feature extractors with both our and an existing self-supervised method SimCLR which we evaluated on the SoccerNet dataset using the action spotting task.

With the same model representing the classification head and only varying the backbone, we showed that our method achieved an a_mAP of 41.13% in the action spotting task, which is 7.89% less compared to the existing self-supervised method SimCLR. The performance gap between the lightweight EfficientNet-B0 model trained with both SimCLR and our pretext task and a substantially bigger ResNET-152 model is relatively small compared to the number of parameters and learning capacity that they dispose of.

We hypothesize that using a bigger model together with our method can achieve even better results and overcome the pretrained ResNET-152 in the action spotting task. We show that our pretext task does not rely on any information about the dataset and therefore can be applied to other domains as well.

References

[1] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6, 2017.

[2] Azrel Aiman Azeman, Aida Mustapha, Nazim Razali, Aziz Nanthaamomphong, and Mohd Helmy Abd Wahab. Prediction of football matches results: Decision forest against neural networks. In

2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pages 1032–1035, 2021.

[3] Arik Badami, Mazen Kazi, Sajal Bansal, and Krishna Samdani. Review on video refereeing using computer vision in football. In *2018 IEEE Punecon*, pages 1–8, 2018.

[4] The International Football Association Board. Laws of the game 23/24. Accessed: 7-1-2024.

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *CoRR*, abs/2006.09882, 2020.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020.

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. Big self-supervised models are strong semi-supervised learners. *CoRR*, abs/2006.10029, 2020.

[8] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020.

[9] Anthony Cioppa, Adrien Deliège, Silvio Giancola, Bernard Ghanem, Marc Van Droogenbroeck, Rikke Gade, and Thomas B. Moeslund. A context-aware loss function for action spotting in soccer videos. *CoRR*, abs/1912.01326, 2019.

[10] Anthony Cioppa, Silvio Giancola, Vladimir Somers, Floriane Magera, Xin Zhou, Hassan Mkhallati, Adrien Deliège, Jan Held, Carlos Hinojosa, Amir M. Mansourian, Pierre Miralles, Olivier Barnich, Christophe De Vleeschouwer, Alexandre Alahi, Bernard Ghanem, Marc Van Droogenbroeck, Abdullah Kamal, Adrien Maglo, Albert Clapés, Amr Abdelaziz, Artur Xarles, Astrid Orcesi, Atom Scott, Bin Liu, Byoungkwon Lim, Chen Chen, Fabian Deuser, Feng Yan, Fufu Yu, Gal Shitrit, Guanshuo Wang, Gyusik Choi, Hankyul Kim, Hao Guo, Hasby Fahrudin, Hidenari Koguchi, Håkan Ardö, Ibrahim Salah, Ido Yerushalmy, Iftikar Muhammad, Ikuma Uchida, Ishay Be'ery, Jaonary Rabarisoa, Jeongae Lee, Jiajun Fu, Jianqin Yin, Jinghang Xu, Jongho Nang, Julien Denize, Junjie Li, Junpei Zhang, Juntae Kim, Kamil Synowiec, Kenji Kobayashi, Kexin Zhang, Konrad Habel, Kota Nakajima, Licheng Jiao, Lin Ma, Lizhi Wang, Luping Wang, Menglong Li, Mengying Zhou, Mohamed Nasr, Mohamed Abdelwahed, Mykola Liashuha, Nikolay Falaleev, Norbert Oswald, Qiong Jia, Quoc-Cuong Pham, Ran Song, Romain Hérault, Rui Peng, Ruilong Chen, Ruixuan Liu, Ruslan Baikulov, Ryuto Fukushima, Sergio Escalera, Seungcheon Lee, Shimin Chen, Shouhong Ding, Taiga Someya, Thomas B. Moeslund, Tianjiao Li, Wei Shen, Wei Zhang, Wei Li, Wei Dai, Weixin Luo, Wending Zhao, Wenjie Zhang, Xinquan Yang, Yanbiao Ma, Yeeun Joo, Yingsen Zeng, Yiyang Gan, Yongqiang Zhu, Yujie Zhong, Zheng Ruan, Zhiheng Li, Zhijian Huang, and Ziyu Meng. Soccernet 2023 challenges results, 2023.

[11] Adrien Deliège, Anthony Cioppa, Silvio Giancola, Meisam Jamshidi Seikavandi, Jacob V. Dueholm, Kamal Nasrollahi, Bernard Ghanem, Thomas B. Moeslund, and Marc Van Droogenbroeck. Soccernet-v2 : A dataset and benchmarks for holistic understanding of broadcast soccer videos. *CoRR*, abs/2011.13367, 2020.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[13] Julien Denize, Mykola Liashuha, Jaonary Rabarisoa, Astrid Orcesi, and Romain Hérault. Comedian: Self-supervised learning and knowledge distillation for action spotting using transformers, 2023.

[14] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. *CoRR*, abs/1505.05192, 2015.

[15] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. *CoRR*, abs/1804.04527, 2018.

[16] Silvio Giancola, Anthony Cioppa, Adrien Deliège, Floriane Magera, Vladimir Somers, Le Kang, Xin Zhou, Olivier Barnich, Christophe De Vleeschouwer, Alexandre Alahi, Bernard Ghanem, Marc Van Droogenbroeck, Abdulrahman Darwish, Adrien Maglo, Albert Clapés, Andreas Luyts, Andrei Boiarov, Artur Xarles, Astrid Orcesi, Avijit Shah, Baoyu Fan, Bharath Comandur, Chen Chen, Chen Zhang, Chen Zhao, Chengzhi Lin, Cheuk-Yiu Chan, Chun Chuen Hui, Dengjie Li, Fan Yang, Fan Liang, Fang Da, Feng Yan, Fufu Yu, Guanshuo Wang, H. Anthony Chan, He Zhu, Hongwei Kan, Jiaming Chu, Jianming Hu, Jianyang Gu, Jin Chen, João V. B. Soares, Jonas Theiner, Jorge De Corte, José Henrique Brito, Jun Zhang, Junjie Li, Junwei Liang, Leqi Shen, Lin Ma, Lingchi Chen, Miguel Santos Marques, Mike Azatov, Nikita Kasatkin, Ning Wang, Qiong Jia, Quoc Cuong Pham, Ralph Ewerth, Ran Song, Rengang Li, Rikke Gade, Ruben Debien, Runze Zhang, Sangrok Lee,

Sergio Escalera, Shan Jiang, Shigeyuki Odashima, Shimin Chen, Shoichi Masui, Shouhong Ding, Sin-wai Chan, Siyu Chen, Tallal El-Shabrawy, Tao He, Thomas B. Moeslund, Wan-Chi Siu, Wei Zhang, Wei Li, Xiangwei Wang, Xiao Tan, Xiaochuan Li, Xiaolin Wei, Xiaoqing Ye, Xing Liu, Xinying Wang, Yandong Guo, Yaqian Zhao, Yi Yu, Yingying Li, Yue He, Yujie Zhong, Zhenhua Guo, and Zhiheng Li. Soccernet 2022 challenges results. In *Proceedings of the 5th International ACM Workshop on Multimedia Content Analysis in Sports*, MM '22. ACM, October 2022.

[17] Silvio Giancola, Anthony Cioppa, Julia Georgieva, Johsan Billingham, Andreas Serner, Kerry Peek, Bernard Ghanem, and Marc Van Droogenbroeck. Towards active learning for action spotting in association football videos, 2023.

[18] Silvio Giancola and Bernard Ghanem. Temporally-aware feature pooling for action spotting in soccer broadcasts. *CoRR*, abs/2104.06779, 2021.

[19] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728, 2018.

[20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *CoRR*, abs/2006.07733, 2020.

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *CoRR*, abs/1911.05722, 2019.

[22] James Hong, Haotian Zhang, Michaël Gharbi, Matthew Fisher, and Kayvon Fatahalian. Spotting temporally precise, fine-grained events in video, 2022.

[23] Xin Hu. Football player posture detection method combining foreground detection and neural networks. *Scientific Programming*, 2021:4102294, June 2021.

[24] Mihnea Bogdan Jurca and Ion Giosan. A modern approach for positional football analysis using computer vision. In *2022 IEEE 18th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 275–282, 2022.

[25] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[26] Wenye Lin, Yifeng Ding, Zhixiong Cao, and Hai tao Zheng. Establishing a stronger baseline for lightweight contrastive models, 2023.

[27] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Unsupervised learning using sequential verification for action recognition. *CoRR*, abs/1603.08561, 2016.

[28] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. *CoRR*, abs/1603.09246, 2016.

[29] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks, 2015.

[30] João V. B. Soares and Avijit Shah. Action spotting using dense detection anchors revisited: Submission to the soccernet challenge 2022, 2022.

[31] Johannes Stübinger, Benedikt Mangold, and Julian Knoll. Machine learning in football betting: Prediction of match results based on player characteristics. *Applied Sciences*, 10(1), 2020.

[32] Ekansh Tiwari, Prasanjit Sardar, and Sarika Jain. Football match result prediction using neural networks and deep learning. In *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pages 229–231, 2020.