SoccerMaster

A Preprint

Most existing work in football analytics focuses on single-task models. In contrast, the authors propose a unified model designed to handle a wide range of soccer visual understanding tasks, from fine-grained perception (e.g., athlete detection) to higher-level semantic reasoning. Specifically, the model is pretrained in a supervised multi-task setting on the following tasks: (1) Athlete Detection, (2) Pitch Registration, (3) Event Classification, and (4) Vision–Language Alignment

Previous work and data

Previous works that tried to unify tasks via vision-language alignment often neglect dense spatial objectives during pretraining which results in mismatch between geometric perception and semantic reasoning. A lot of work from SoccerMaster paper adress data problem. Authors made automated data pipeline which is capable of generating highquality annotations from broadcast footage at scale. For Field registration authors use PnLCalib to estimate camera parameters, enabling projection between image and standardized pitch coordinates. For tracking authors use YOLOv8 fine-tuned on soccer data, to detect players, goalkeepers, and referees, and StrongSORT with features from PRTReID. For each detection they crop bbox and apply Qwen2.5-VL. After this, Post-processing refinement via SAM2 is applied. This technique uses SAM2 to refine segmentations, which helps in recovering detections that were missed by the initial detector and correcting instances where the identity of a tracked object might have switched. To enhance temporal consistency for jersey numbers and role classifications, the pipeline applies a majority voting strategy across tracklets. This means that for a given tracklet, the most frequently observed jersey number or role is chosen, making the annotation more stable over time. Short, fragmented trajectories are merged into longer, more complete tracklets. This is achieved by leveraging both ReID (re-identification) embeddings and the consistency of jersey numbers, similar to the approach in From broadcast to minimap. This helps in creating more robust and continuous tracking of athletes throughout a video.

Athlete detection

For each frame authors aim to detect a set of athlete instances where each instance is described by $a_i = (b_i, r_i, n_i)$. b_i – represents the 2D bounding box of the detected athlete; r_i – represents role(goalkeeper, player, referee); n_i – represents jersey number. Notably, when the detected athlete's role (r_i) is goalkeeper or referee, or when the jersey number is not visible due to occlusion or viewpoint issues, the corresponding number (n_i) takes the value null

Pitch registration

The objective is to identify and localize critical structural elements of the soccer pitch, including keypoints (e.g., intersection between side lines) and lines (e.g., middle line and goal crossbar). For each frame the set of lines and keypoints is detected

Event classification

Event classification aims to recognize key actions within video segments. As input it gets video clip and as output the model predicts an event label(across the set of 24 event categories). It is important to note that the model is designed to classify a single event per clip, where the clip is already focused on that event

Vision-language alignment

The whole tasks basically aims to learn connections between video content and textual commentary. Commentary generation is also built upon this training

Camera calibration

The task is to predict camera params for later usage in homography

Multiple object tracking

For each detected athlete is assigns id to each detection, thereby constructing trajectories that maintain consistent identities throughout the video. Team affiliation is determined through clustering on tracklet-averaged ReID embeddings and pitch coordinates mapped via the estimated camera parameters.

The model itself

Given a soccer video segment visual encoder extracts spatial end semantic features. After that, task specific heads are used

$$\{\mathcal{A}, id, \mathcal{K}, \mathcal{L}, K, R, t, e, \} = \Psi(features)$$

where Ψ is alignment head computes the similarity (s) between video semantic features and text features of textual commentary extracted by a pretrained text encoder(semantic features and). Visual encoder inherits ViT, incorporates a TimeSformer-like design to enable spatiotemporal attention and capture temporal features from videos. To balance performance and efficiency, we use spatial attention in the first s layers and apply spatiotemporal attention only in the final st layers. The visual feature extraction consists of 3 stages:

- (1) token embedding
- (2) spatial encoding
- (3) spatiotemporal encoding

(1) Process each frame following the tokenization procedure of ViT

(2) The initial token sequence passes through spatial transformer blocks, which process individual frames separately. The SoccerMaster visual encoder processes input video segments in stages. Initially, the token sequence $z^{(0)}$ undergoes L_s spatial transformer blocks. These blocks operate independently on each frame, employing a standard transformer architecture comprising multi-head self-attention, LayerNorm, and a feedforward network. The spatial self-attention mechanism, central to this stage, is defined as:

$$z_{t,i}^{(l+1)} = \text{SpatialAttn}(z_{t,i}^{(l)}, \{z_{t,j}^{(l)}\}_{j=1}^{N}) \quad (6)$$

In this formulation, each token at a spatial position i and temporal position t attends solely to other tokens within the same temporal position t, effectively restricting information exchange to individual frames. The output of the L_s-th spatial attention block yields the extracted spatial features, denoted as $F_{\text{spa}} = z^{(L_s)} \in \mathbb{R}^{T\times h\times w\times d}$, which retain fine-grained spatial details for all frames.

Following the spatial encoding, learnable temporal positional embeddings ($P_{\text{tem}} \in \mathbb{R}^{T\times d}$) are introduced to the spatial features (F_{spa}) to incorporate temporal ordering. A TimeSformer-like approach [5] is then employed, extending the remaining L_{st} attention blocks to spatiotemporal attention. Each of these blocks alternately performs temporal and spatial attention.

For a token at temporal position t and spatial position i, temporal attention is expressed as:

$$z_{t,i}^{(l+\frac{1}{2})} = \text{TemporalAttn}(z_{t,i}^{(l)}, \{z_{t',i}^{(l)}\}_{t'=1}^{T}) \quad (7)$$

This ensures interaction among all tokens at the same spatial position across different temporal positions, without exchanging information across distinct spatial positions. Subsequently, a spatial attention layer is applied:

$$z_{t,i}^{(l+1)} = \text{SpatialAttn}(z_{t,i}^{(l+\frac{1}{2})}, \{z_{t,j}^{(l+\frac{1}{2})}\}_{j=1}^{N}) \quad (8)$$

After passing through L_s spatial and L_{st} spatiotemporal attention blocks, an MAP head (attention pooling) is applied across the spatial dimensions of the final features, $z^{(L_s+L_{st})}$. This yields the final semantic features, denoted as $F_{\text{sem}} = \text{MAP}(z^{(L_s+L_{st})}) \in \mathbb{R}^{T\times d}$, which effectively capture global dynamic semantic information. This hierarchical design enables the model to capture both spatial details and temporal dynamics, resulting in unified representations for various downstream tasks

Pretrain

Visual encoder with a multi-task framework, on spatial perception tasks (e.g., athlete detection, pitch registration) and semantic reasoning tasks (e.g., event classification, vision-language alignment), with lightweight output heads.

For athlete detection they use a lightweight Deformable DETR decoder-like structure to perform attention between learnable queries and the extracted spatial features (Fspa) to obtain object-level features, which are then passed through three linear layers that predict the athlete bounding box (b_i), player role (r_i), and jersey number (n_i), respectively

For pitch registration, two convolutional networks with the same idea of heatmap as in PnLCalib is used to predict keypoint and endpoint heatmaps for each frame. Specifically, each head uses convolutional layers containing PixelShuffle to perform progressive upsampling on features, ultimately using MSE losses between predicted lines keypoints and GT lines keypoints.

For event classification, the event classification head adopts a two-layer transformer encoder, followed by temporal average pooling and a linear classifier, which predicts soccer event based on semantic features

For Vision-language alignment, alignment head first performs temporal average pooling on semantic reasoning features(semantic) for video representations and computes similarity with text embeddings from SigLIP2 with SigLIP loss optimization

For any next task, model may be adopted by adding task-focused head.

OVERALL, the model SoccerMaster adopts a hierarchical vision transformer architecture, initialized with weights from siglip2large-patch16-51 with 16 spatial transformer blocks and 8 spatiotemporal transformer blocks, with a hidden dimension size equals 1024

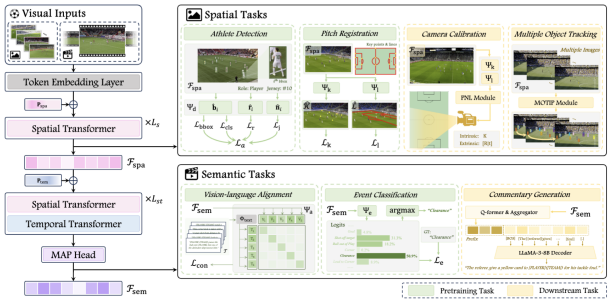

Figure 3. **SoccerMaster Architecture.** (a) The architecture of SoccerMaster, which encodes both soccer videos and images through spatial and temporal attention modules to generate semantically rich representations. (b) The pretraining tasks and downstream adaptations of SoccerMaster across both spatial perception and semantic understanding tasks.