JEDE: Universal Jersey Number Detector for Sports

Hengyue Liu and Bir Bhanu, *Life Fellow, IEEE*

***Abstract*— The rapid progress in deep learning-based computer vision has opened unprecedented possibilities in computing various high-level analytics for sports. Artificial intelligence techniques such as predictive analysis, automatic highlight generation, and assistant coaching have been applied to improve performance and decision-making for teams and players. To perform any high-level analysis from a game match, collecting the locations (where) and identities (who) of players is crucial and challenging. In this paper, a universal JErsey number DEtector (JEDE) for player identification is presented that predicts players' bounding boxes and keypoints, along with bounding boxes and classes of jersey digits and numbers in an end-to-end manner. Instead of generating digit proposals from pre-defined anchors, JEDE predicts more robust proposals guided by players' features and pose estimation. Moreover, a dataset is collected from soccer and basketball matches with annotations on players' bounding boxes and body keypoints, and jersey digits' bounding boxes and labels. Extensive experimental results and ablation studies on the collected dataset show that the proposed method outperforms the state-of-the-art methods by a large margin. Both quantitative and qualitative results also demonstrate JEDE's superior practicality and generalizability over different sports.**

***Index Terms*— Jersey number detection, player identification, player statistics, sports analytics, video analysis.**

I. Introduction

RECENTLY, there has been a tremendous growing interest in artificial intelligence (AI) technology for sports. Every aspect of sports, from the recruitment of athletes to the analysis of performance, from game planning to injury management, from audience experience to media, is empowered by AI. Not only industry has substantially explored new technologies for sports, but also academia has dramatically strengthened the research capacity on the topic. Among various AI applications, computer vision (CV) for sports has one of the most significant potentials which may have a huge impact on the way people view and consume sports content. For example, current tracking systems [1], [2] are deployed in stadiums and collect comprehensive game data on players, referees, and the ball in real-time. The basic statistics of players' moving direction, speed, and acceleration, and even more advanced statistics could be obtained via machine learning. Knowing players' locations, augmented reality (AR) can be applied in live broadcasting to provide entertainment enhancements such as ball movement diagrams, player identifications, scoring probabilities, *etc*. There are many other CV applications for sports, such as event detection [3]–[5], activity recognition [6], human pose estimation [7]–[9], human motion prediction [10], [11], automatic highlight generation [12], and image generative models [13]. Moreover, the research in CV for sports is not only about generating statistics, but also about scene understanding and human behavior analysis.

We have seen deep learning models [14] that defeat humans in many games like Go, Chess, and Atari. These games serve as perfect simulators for learning. Analogically for real-world applications, sports games are the perfect simulation environment for scene and human behavior understanding. Tuyls *et al*. [15] propose three foundational areas associated with soccer AI research: statistical learning, computer vision, and game theory. Computer vision models provide the complementary high-level and spatially-detailed features for the other two areas, while benefit from low-dimensional game-related statistics and metadata from them. Shih [16] proposes the content pyramid for sports video analytics, which consists of four layers: video, object, action, and conclusion. The object layer as the second lowest level, connects the raw data processing and higher-level analysis. Undoubtedly, object detection or player identification is the most important building block for sports video analysis. Traditional methods for player identification rely on hand-crafted features [17], [18] or face recognition [19]–[23], which are infeasible for complex scenes or different fields-of-view. Researchers also try to solve the problem by detecting the jersey number since it is the generic visual cue of identity. Early approaches [24]–[27] are based on optical character recognition (OCR) to extract and classify numbers. However, these methods are not robust to the challenges in broadcast sports videos, such as illumination changes [28], low jersey number resolution, viewpoint and camera movements [18], players' pose deformation, occlusion, motion blur [29], and stadium distractions [30], *etc*. Deep learning has been widely applied in CV, but there are only a few papers on jersey number classification [31], [32] or player detection [33]–[35], not to mention end-to-end jersey number detection. Previous work [31], [32] is only applicable for single-person images to perform image classification, but not for frames consisting of multiple players. In this paper, we propose a novel jersey number detection framework for player identification in sports videos, named as universal JErsey number DEtector (JEDE). It is a multi-stage detector, which predicts players' bounding boxes and pose estimations, with associated jersey digits' bounding boxes and classes all at once. The first stage extracts image features through

Manuscript received 24 March 2022; revised 2 June 2022; accepted 20 June 2022. Date of publication 30 June 2022; date of current version 28 October 2022. This work was supported in part by the Bourns Endowment and in part by SEVAai through the University of California, Riverside. This article was recommended by Associate Editor S. Gao. *(Corresponding author: Hengyue Liu.)*

The authors are with the Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521 USA (e-mail: hliu087@ucr.edu; bhanu@ee.ucr.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCSVT.2022.3187670.

Digital Object Identifier 10.1109/TCSVT.2022.3187670

a "backbone" network (*e.g.*, ResNet-50 [36]) and constructs a feature pyramid [37]; then, a Region Proposal Network (RPN) [38] is used for generating player candidate proposals. The second stage extracts features using RoIAlign [39] from each player's candidate box and performs classification, bounding-box regression, and human-body keypoint regression. In parallel with these detections, we add a branch that predicts the bounding boxes of digits of the jersey number within each player's bounding box. Both the player's features and corresponding keypoint predictions are used for generating digit proposals. More specifically, we model individual digit as an object, which is represented by the center and size of its bounding box. Within each player proposal, we regress the center and size heatmaps of the digits given the extracted player features and keypoint heatmaps. By conditioning on human pose information, the localization of digits is significantly improved. We then extract features from digit proposals and perform digit classification and bounding box regression. Finally, the digit detections are paired as number detections. Our framework is performed on a per-frame basis with fast inference speed, and no motion information is used. Besides the novel architecture, two data augmentation techniques called `CopypasteMix` and `SwapDigit` are proposed. `CopypasteMix` creates new training data by copying and pasting among images, while `SwapDigit` by swapping digit instances with data from other datasets such as Street View House Number (SVHN) [40]. This paper significantly extends our previous work [30] by re-designing the architecture, proposing new data augmentation methods, and providing more experimental results and ablation studies. The contributions of this paper are:

1) We tackle the player identification problem via jersey number detection that is more robust to real-world variations. We propose the first framework that can simultaneously predict players' bounding boxes, pose estimations, jersey digits' and numbers' bounding boxes and classes. The rich predictions provided by our framework are significant for higher-level analysis.
2) Unlike previous jersey number recognition frameworks, jersey number detection addressed in this paper is a challenging multi-player multi-digit detection problem. Our proposed model JEDE generates jersey number detections from instance-level digit localization and classification, which is much more accurate and reliable.
3) We collect a dataset consisting of 4477 images from soccer and basketball matches. There are 6054 labeled players with 5406 labeled human body pose, and 6293 labeled digits. Moreover, we propose data augmentation strategies named `CopypasteMix` and `SwapDigit` that effectively improve detection performance and robustness. We also explore pre-training on COCO [41] and SVHN [40] datasets, which further improve the results.
4) We conduct comprehensive evaluations, ablation studies, and comparisons of the proposed framework with the state-of-the-art methods for jersey number recognition, object detection, and scene text detection on the collected dataset. We also show that the proposed method is easily generalized on wild images across different sports with superb performance.

This paper is organized as follows: an overview of related research is presented in Section II. In Section III, the technical approach is explained in detail. In Section IV, comprehensive experimental results and ablations studies are presented and discussed in depth. Section V concludes this paper.

II. Related Work

This work is mainly focused on jersey number detection, which is also highly related to many general vision tasks such as person re-identification (Re-ID) [42], object detection [38], [43], multi-object tracking (MOT) [44], and scene text detection [45]–[47]. Reviewing all the related literature is beyond the scope of this paper, thus we only discuss the most relevant research on sports analysis.

A. Player Detection and Tracking

Player detection and tracking are important techniques that are required for sports video analysis, providing the spatial and temporal information about players. Player detection is the preliminary step for player identification that generates bounding boxes of players, while player tracking associates the bounding boxes between frames and assigns a tracking ID for each bounding box. Traditional methods rely on handcrafted features. For example, Lu *et al*. [27], [48] use the deformable part model (DPM) for player detection and then perform player classification based on handcrafted features and tracking information. Gerke *et al.* [49] augment Histogram of Oriented Gradients (HOG) features [50] with jersey color information to improve player detection performance. Modern deep-learning approaches adopt off-the-shelf object detectors and their variants for player detection [51]–[54]. For player tracking, off-the-shelf multi-object tracking algorithms are commonly used [55]–[58]. Sentioscope [59] is one example of such a system that maps the image to a modeled soccer field, performs player detection, builds a likelihood model based on appearance and motion, classifies teams based on jersey colors, and assigns identity tags to tracks.

B. Jersey Number Recognition

Jersey number recognition can be considered as the task of person identification (ID) in the context of sports broadcast videos where each player's ID is uniquely associated with the jersey number. Player identification can be performed directly based on the player's appearance or pose features [48], [58], [60]–[62], but re-training is required if the match roster or target sport changes. Jersey number recognition provides a relatively more general and robust solution to player identification. Most approaches can only perform jersey number recognition on images that only contain a single player. Traditional approaches before the dominance of deep learning usually first build an OCR system, and then classify numbers based on segmentation results. Šari *et al*. [25] introduce an OCR system to segment images in HSV color space with heavy pre-processing and post-processing. Ye *et al*. [24] combine tracking information of frames and a OCR system to predict jersey number based on voting. These OCR-based methods have limited flexibility and robustness on real-world

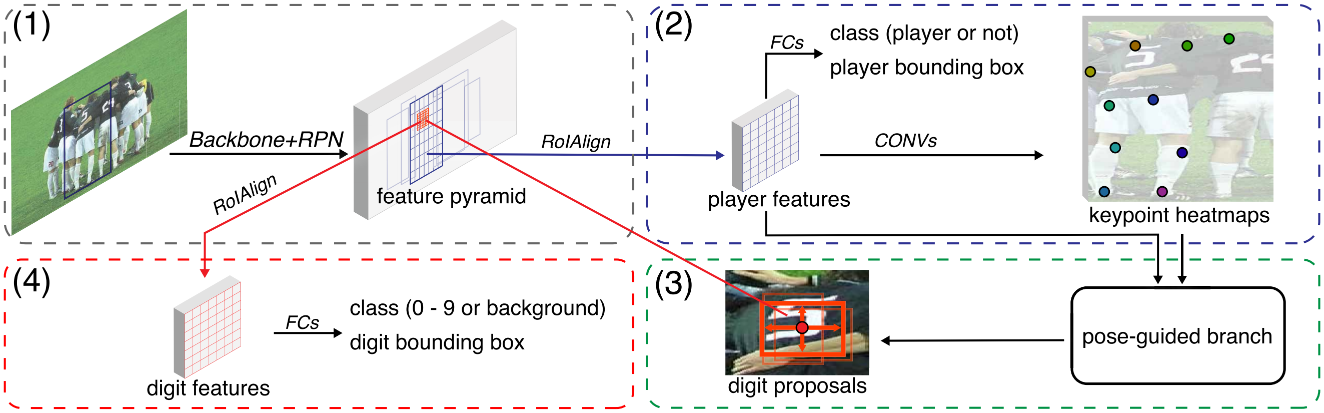


Fig. 1. The architecture of JEDE - it can be divided into four modules: (1) a backbone network that extracts features and constructs a feature pyramid (*e.g.*, ResNet-50 FPN) followed by a RPN (Section III-B.1); (2) a player branch that extracts features from player proposals generated from the RPN via RoIAlign, performs classification, bounding-box regression, and keypoints regression (Section III-B.2); (3) a pose-guided branch that predicts digit proposals from the pooled player's features and corresponding keypoint heatmaps (Section III-C); (4) a digit branch that extracts features from digit proposals, and then performs digit classification and bounding-box regression (Section III-D.1). Fully-connected layers are denoted by FCs, and convolutional layers by CONVs.

data. Switching to deep learning approaches, Gerke *et al.* [31] designs a neural network for jersey number recognition in cropped jersey number images. Li *et al*. [32] propose a framework that adopts Spatial Transformer Network (STN) [63] to refine jersey number features automatically, which is trained with additional labeled transformation quadrangles in a semi-supervised fashion. Some work takes the given sports field into consideration: Delannay *et al*. [64] create ground plane occupancy maps from multi-view detections to perform localization, followed by an OCR system with a Support Vector Machine classifier; Gerke *et al*. [65] combine the players' spatial constellation features and jersey number features from CNNs to achieve better per-game player recognition performance. These work make strong assumptions on the hidden pattern of player's movement and accurate inverse homography, which is not practical or generalizable for other sports.

C. Jersey Number Detection

There is limited work on jersey number detection due to significant challenges like human pose deformation, camera view changes, motion blur, and various illumination conditions. Traditional OCR-based methods [24], [25] can only perform single jersey number detection on single-player images with close-up views. Our previous work [30] explores deep-learning-based multi-player multi-digit jersey number detection, and proposes a pose-guided R-CNN that still has some limitations. It requires associations between player and digit bounding boxes, where wrong associations may occur in crowded scenes. It does not work well on images with a wider field-of-view due to insufficient training data. In this paper, JEDE addresses these challenges and limitations. The major extension over [30] lies in the pose-guided branch and data augmentation. The re-designed pose-guided branch directly predicts digit proposals from each player proposal instead of using RPN, so no association of bounding boxes is needed. It provides more accurate and robust digit proposals based on the player's features and keypoints. The proposed data augmentation strategies `CopypasteMix` and `SwapDigit` introduce more training data variations that significantly alleviate the problem of limited data as compared to [30]. As a result, the proposed framework JEDE achieves the state-of-the-art results and outperforms pose-guided R-CNN by a large margin. The contributions of the paper are summarized in Section I.

III. Technical Approach

R-CNN and its variants [38], [39], [66]–[68] are flexible, general, and extensible for many computer vision tasks, such as object detection, instance segmentation, human pose estimation, and panoptic segmentation. This flexibility provides more capabilities for sports analysis that involves more complex scene dynamics. In this section, we explain our overall framework and individual modules of our proposed method.

A. Problem Setup

A jersey number is defined as the unique number on the player's uniform to identify players. In our work, only the number printed on the back is considered since it typically exists for most team sports. As a jersey number consists of a sequence of at most two digits in most sports [69], we only consider detecting the number with a maximum length of two digits. The task is then to predict the bounding box and class of any visible and recognizable digit instance on the back of the jersey in an image. We formulate jersey number detection as a multi-step approach: player detection, digit detection, and jersey number detection. Player detection is based on two-stage Faster R-CNN; digit detection is a top-down approach performed on each Region-of-Interest (RoI) of detected players; we then generate jersey number candidates based on the predicted digits. For jersey number recognition, previous work [31], [32] simply treats it as a number classification task, while our approach performs per-instance digit classification and association within each player's RoI.

The overall architecture of JEDE is presented in Fig. 1. Inspired by Mask R-CNN [39], the framework consists of four main components: a feature pyramid network (FPN) [37] as the backbone, followed by a region proposal network (RPN) [38] for generating player proposals; a player branch for player/background classification, bounding boxes regression, and pose estimation; a pose-guided branch for generating digit proposals; a digit branch for digit classification and bounding boxes regression. The final jersey numbers are generated from digit detections as a post-processing step which will be discussed in Section III-D.2.

B. Backbone, RPN and Player Branch

1) Backbone and RPN: Similar to scene text detection, jersey number detection is challenging because of varying sizes and fonts of jersey numbers in sports broadcasting. The scale of a player changes with the change of the camera and its viewpoint. Therefore, the scale of jersey numbers also changes in a wide range. To capture high-level semantic features at all scales, a feature pyramid [37] is constructed from ResNet [36] features. RPN is used to generate player proposals for the subsequent player and pose-guided branches. We use 5 scales of anchors {32, 64, 128, 256, 512}, and 3 aspect ratios {0.5, 1, 2} following Faster R-CNN [37], [38]. As shown in section IV later, we achieve similar results with faster inference speed by removing the anchor size of 32.

2) Player Branch: The player branch includes three tasks: binary classification (player *vs.* background), bounding box regression, and keypoints regression. Given the player proposals from RPN, RoIAlign [39] is used for extracting features. We keep the same Mask R-CNN [39] heads (small prediction networks) with pre-trained weights for faster convergence, where the pooling size is 7 × 7 (pixels in feature maps) for classification and bounding box regression, and 14 × 14 for keypoints regression. For human body keypoint detection, we predict a mask of shape 17 × 56 × 56 for each player RoI, where there are 17 types of person keypoints following the COCO dataset [41], and the output feature side length is 56. Please refer to Mask R-CNN [39] for more details.

C. Pose-Guided Branch

For jersey number detection task, previous work [30] has demonstrated that better jersey number localization can be achieved given human pose information. Though Faster R-CNN is capable of regressing jersey number or digit bounding boxes directly, there are limitations under more difficult scenarios. For example, varying jersey patterns, fonts, hash marks, and commercial banners introduce difficulties in generating satisfactory proposals for RPN. To tackle these problems, we introduce a pose-guided branch for refining digit localization conditioned on the player detection and pose estimation. Each digit proposal is generated based on the regressions of its center and bounding box size within a player proposal. We also provide a theoretical analysis on why the human pose helps in digit localization in the supplementary material.

1) Design: We consider a single player proposal for illustration. Given the player's bounding box predicted from the player branch, the player features are pooled from the feature pyramid as one input to the pose-guided branch. Another input is the regressed keypoint heatmaps. Since the feature dimensions may be different, we use small fully convolutional networks (FCNs) for adjusting feature dimensions. Specifically, a convolution kernel with stride of 2 is used for downsampling, and bilinear interpolation is used for upsampling, if needed. The player features generated by the FCN (two 3 × 3 64-channel convolutional layers by default) is denoted by F_{player}. Depending on the configurations, the spatial dimension of F_{player} may remain the same or increase to have higher-resolution features for digit localization. The keypoint heatmaps are downsampled spatially to have larger reception fields via a FCN, capturing more semantic features from the pose estimations. We name the resulting features as F_{kpts}. Both features are then fused as $F = \textsf{fusion}(F_{\text{player}}, F_{\text{kpts}}) \in \mathbb{R}^{C \times M \times M}$, where $\textsf{fusion}$ is the function that combines the two input features, C is the output number of channels, and M is the output feature side length. The feature fusion can be either $\textsf{concatenation}$, $\textsf{addition}$, or $\textsf{multiplication}$. The ablation study on fusion methods is provided in Section IV-F. Optionally, positional information can be considered as additional features. We adopt the extended 2D version of positional embeddings [70], and concatenate them with F_{kpts}. We can also concatenate the embeddings with F which is less effective as shown in the ablation study (Section IV-F).

2) Output and Ground-Truth Generation: We then regress heatmaps for digit center, center offset, and size respectively. The fused features F will be fed into 3 FCN prediction heads, each of which consists of four 3×3 64-channel convolutional layers. Since there are at most 2 digits for a jersey number for most of the sports, predicting two-channel center heatmaps $\mathbf{O} \in \mathbb{R}^{2 \times M \times M}$ is sufficient. For a single-digit jersey number, the ground-truth (GT) center is only defined on the first channel. As for a two-digit jersey number, the left digit center is defined on the first channel, and the right digit center on the second channel. Specifically, we define the GT digit center class $d \in \{0, 1\}$ and the bounding box (x_0, y_0, x_1, y_1) where (x_0, y_0) and (x_1, y_1) denote the coordinates of the left-top and right-bottom corners. The center is computed as $\mathbf{o} = (o_x, o_y) = ((x_0+x_1)/2, (y_0+y_1)/2)$. We then need to map the digit center coordinates into the feature scale. Given the corresponding player's bounding box $(x_0^p, y_0^p, x_1^p, y_1^p)$, we compute the relative coordinates with respect to the player's bounding box as $(o_x - x_0^p, o_y - y_0^p)$. The feature-to-bounding-box width and height ratios are computed as

$$r_w = M/(x_1^p - x_0^p),\ r_h = M/(y_1^p - y_0^p), \tag{1}$$

respectively. Finally, the digit center in the $M \times M$ feature grid is

$$(o'_x, o'_y) = (r_w \cdot (o_x - x_0^p), r_h \cdot (o_y - y_0^p)). \tag{2}$$

For regression of the GT digit center, we quantize the coordinates, and assign the value of 1 at $(\lfloor o'_x \rfloor, \lfloor o'_y \rfloor)$ and 0 otherwise, where $\lfloor \cdot \rfloor$ is the floor function. To obtain more positive training samples, the object center will be modulated by a bivariate Gaussian distribution along the x-axis and y-axis following Law [71] and Zhou *et al.* [72]. Given the size of the bounding box $(w, h) = (x_1 - x_0, y_1 - y_0)$, the feature-scale size is $(w', h') = (r_w \cdot w, r_h \cdot h)$. Based on the feature-scale bounding box size, and the desired minimum

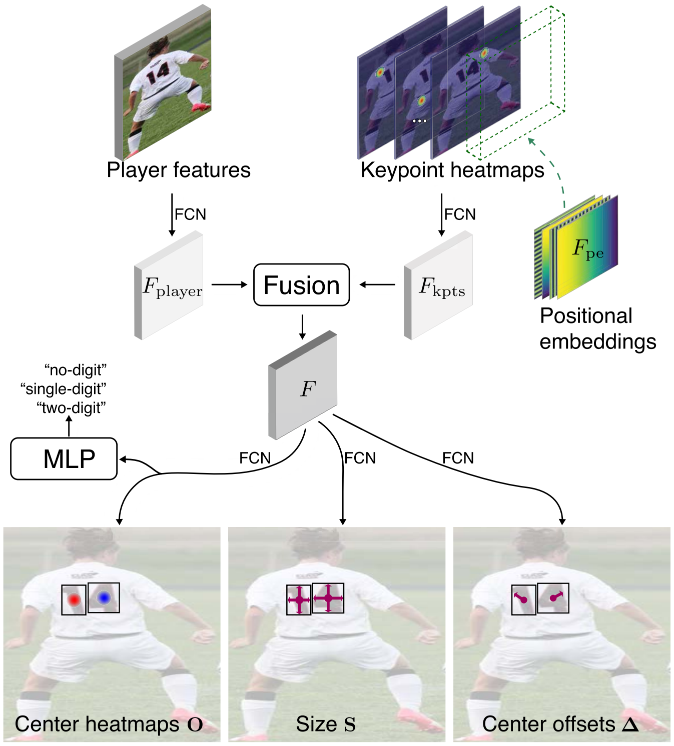


Fig. 2. The architecture of the pose-guided branch. In this example, a two-digit jersey number is predicted. the center of digit "1" is predicted on the first channel of **O** shown as the red dot, and the center of digit "4" is predicted on the second channel of **O** shown as the blue dot. The predictions of size and center offset are location-aware and class-agnostic.

Intersection over Union (IoU) denoted by min_iou, the Gaussian standard deviations σ_x and σ_y are derived as:

$$(\sigma_x, \sigma_y) = \frac{1}{3}(a, b) = \frac{1}{3}\lfloor \frac{1 - \sqrt{min\_iou}}{\sqrt{2}} \cdot (w', h') + 1\rfloor, \quad (3)$$

where σ_x and σ_y control the spread of the distribution, and a and b are the semi-minor axes along the x-axis and y-axis respectively. The value of min_iou is chosen in the range of $(0, 1)$, such that for any point (x, y) in the ellipse region $R = \{(x, y)|\frac{x^2}{a^2} + \frac{y^2}{b^2} \leq 1\}$, when using it as a center to create a bounding box of size (w, h), the bounding box has at least min_iou IoU with the GT. Then, for a digit of class d, its GT center heatmaps are computed as

$$\mathbf{O}_{d,x,y} = \begin{cases} \exp\big(- \frac{\|x - \lfloor o'_x \rfloor\|_2^2}{2\sigma_x^2} - \frac{\|y - \lfloor o'_y \rfloor\|_2^2}{2\sigma_y^2}\big), \forall (x, y) \in R \\ 0 \quad \text{otherwise.} \end{cases} \quad (4)$$

In addition to center heatmaps, we regress digit center offsets $\Delta \in \mathbb{R}^{2\times M\times M}$ for recovering from downsampled and discretized coordinates. The first and second channels of Δ represent the offsets along x-axis and y-axis respectively. The offset target at the digit center is $\Delta_{\lfloor o'_x \rfloor, \lfloor o'_y \rfloor} = (o'_x - \lfloor o'_x \rfloor, o'_y - \lfloor o'_y \rfloor)$, and 0 on all other locations. For regression of size $\mathbf{S} \in \mathbb{R}^{2\times M\times M}$, the size target at $(\lfloor o'_x \rfloor, \lfloor o'_y \rfloor)$ is the feature-scale width and height (w', h') and 0 otherwise.

3) Losses: Regressions are performed for the three output heatmaps of the pose-guided branch. We use Gaussian focal loss [71]–[73] for digit center heatmaps with default hyper-parameters $\alpha = 2$ and $\gamma = 4$ for weight balancing. Let $\hat{\mathbf{O}}$ be the predicted center heatmaps, then the pixel-wise loss $\mathcal{L}_{\mathbf{O}_{d,x,y}}$ is defined as:

$$\mathcal{L}_{\mathbf{O}_{d,x,y}} = -\begin{cases} (1 - \hat{\mathbf{O}}_{d,x,y})^{\alpha} \log(\hat{\mathbf{O}}_{d,x,y}) & \text{if } \mathbf{O}_{d,x,y} = 1 \\ (\hat{\mathbf{O}}_{d,x,y})^{\alpha}(1 - \mathbf{O}_{d,x,y})^{\gamma} \log(1 - \hat{\mathbf{O}}_{d,x,y}) & \text{o/w.} \end{cases} \quad (5)$$

The offset and size targets are only defined at the GT digit center locations where $\mathbf{O}_{d,x,y} = 1$, and regressed via L1 loss as $\mathcal{L}_{\Delta_{x,y}}$ and $\mathcal{L}_{\mathbf{S}_{x,y}}$. The overall digit detection objective is

$$\mathcal{L}_{det} = \frac{1}{N}\sum_{d,x,y}\left(\mathcal{L}_{\mathbf{O}_{d,x,y}} + \lambda_{\Delta}\mathcal{L}_{\Delta_{x,y}} + \lambda_{\mathbf{S}}\mathcal{L}_{\mathbf{S}_{x,y}}\right), \quad (6)$$

where N is the total number of digits within the player proposal; λ_{Δ} and $\lambda_{\mathbf{S}}$ are hyper-parameters for weight balancing. We empirically set $\lambda_{\Delta} = 1$ and $\lambda_{\mathbf{S}} = 1$ for all experiments.

4) Decoding Digit Proposals: With the output digit center heatmaps $\hat{\mathbf{O}}$, center offsets $\hat{\Delta}$, and digit bounding box sizes $\hat{\mathbf{S}}$, we need to decode the digit proposals for both training and inference. To get the digit centers, we follow the same step in [72]. Specifically, a `sigmoid` function is applied to the predicted center heatmaps $\hat{\mathbf{O}}$ such that the values are mapped into the range of [0, 1]. Then a 3 × 3 max pooling is applied to center heatmaps for filtering duplicate detections. The value of $\hat{\mathbf{O}}_{c,x,y}$ is considered as the measurement of the detection score. Then the top peaks in center heatmaps can be extracted as the detected digit centers $\{\hat{\mathbf{o}}^i\}_{i=1}^{K}$, where $\hat{\mathbf{o}}^i = (\hat{o}_x^i, \hat{o}_y^i)$ and K is the hyper-parameter to control how many digit proposals to keep per player proposal. During training, we set $K = 100$ where the top 50 digit proposals are kept plus 50 random proposals since we need negative training examples for digit classifications. For inference, we set $K = 20$ with the top 20 proposals for balancing the accuracy and inference speed.

To get the corresponding center offset and bounding box size, we gather the values at the detected digit center $(\hat{o}_x, \hat{o}_y)$ from $\hat{\Delta}$ and $\hat{\mathbf{S}}$, namely the center offset $\hat{\Delta}_{\hat{o}_x,\hat{o}_y} = (\hat{\delta}_x, \hat{\delta}_y)$ and bounding box size $\hat{\mathbf{S}}_{\hat{o}_x,\hat{o}_y} = (\hat{w}', \hat{h}')$. The width and height ratios, $\hat{r}_w$ and $\hat{r}_h$, can be computed via Equation 1 given the predicted player proposal $(\hat{x}_0^p, \hat{y}_0^p, \hat{x}_1^p, \hat{y}_1^p)$. Finally, the digit bounding box $(\hat{x}_0, \hat{y}_0, \hat{x}_1, \hat{y}_1)$ can be recovered as

$$\begin{aligned} \hat{x}_0 &= \hat{x}_0^p + (\hat{o}_x + \hat{\delta}_x - \hat{w}'/2)/\hat{r}_w, \\ \hat{y}_0 &= \hat{y}_0^p + (\hat{o}_y + \hat{\delta}_y - \hat{h}'/2)/\hat{r}_h, \\ \hat{x}_1 &= \hat{x}_0^p + (\hat{o}_x + \hat{\delta}_x + \hat{w}'/2)/\hat{r}_w, \\ \hat{y}_1 &= \hat{y}_0^p + (\hat{o}_y + \hat{\delta}_y + \hat{h}'/2)/\hat{r}_h. \end{aligned} \quad (7)$$

To obtain jersey number detections, we group the digit detections based on the digit center class, and determine the jersey number length. We add a small multilayer perceptron (MLP) parallel to the output layer of the center prediction head, for classifying the number length ("no-digit", "single-digit", and "two-digit"). The MLP consists of a MaxPool layer (downsampling by a factor of 2) and 3 fully-connected (FC) layers. The number length is denoted by $l \in \{0, 1, 2\}$, which will be used for generating jersey number detection.

D. Digit Branch and Jersey Number Detection

1) Digit Branch: We sample both positive and negative (ratio of 1:3) digit proposals from the output of the pose-guided branch for training, then extract 7 × 7 digit features via RoIAlign [39]. The digit branch architecture is the same as the player branch except that there are 11 output classes for digit classification, including 10 digit classes and 1 background class. The predicted digit class is denoted by c with a confidence score u, which will be used for generating jersey number detections.

2) Jersey Number Detection: We predict the jersey number length $\hat{l}$ for each player proposal as discussed in Section III-C.4. Then within each player proposal, there are multiple digit detections denoted by $\mathbf{B}_{\text{digit}} = \{\hat{\mathbf{b}}^i\}_{i=1}^K$, where $\hat{\mathbf{b}}^i = (\hat{x}_0^i, \hat{y}_0^i, \hat{x}_1^i, \hat{y}_1^i, \hat{c}^i, \hat{u}^i)$, with the predicted digit center class $\hat{d}^i$ from the pose-guided branch. The jersey number detection is generated based on the predicted number length $\hat{l}$. For $\hat{l} = 0$, we simply discard all the bounding boxes; for single-digit case $\hat{l} = 1$, all the digit detections of $\hat{c}^i = 0$ are considered as number detections. For a two-digit number where $\hat{l} = 2$, we union digit bounding boxes pair-wisely and use the multiplication of corresponding digit class scores as the jersey number score. The jersey number class is the concatenation ($\oplus$) of digit classes for two-digit numbers. The top 100 jersey number detections are selected (`topk`) for evaluation purpose. The process to obtain the jersey number detections $\mathbf{B}_{\text{number}}$ is described in Algorithm 1.

Algorithm 1 Jersey Number Detection

```
Input: Digit detections $\mathbf{B}_{\text{digit}}$, jersey number length $\hat{l}$
Output: Jersey number detections $\mathbf{B}_{\text{number}}$
if $\hat{l} = 0$ then
    $\mathbf{B}_{\text{number}} \leftarrow \varnothing$
else if $\hat{l} = 1$ then
    $\mathbf{B}_{\text{number}} \leftarrow \mathbf{B}_{\text{digit}}$
else if $\hat{l} = 2$ then
    $\mathbf{B}_{d=0} \leftarrow \{\hat{\mathbf{b}}^i | \hat{d}^i = 0\}$, $\mathbf{B}_{d=1} \leftarrow \{\hat{\mathbf{b}}^j | \hat{d}^j = 1\}$
    $\mathbf{B}_{\text{number}} \leftarrow \{(\ \mathtt{min}(\hat{x}_0^i, \hat{x}_0^j), \mathtt{min}(\hat{y}_0^i, \hat{y}_0^j),$
                      $\mathtt{max}(\hat{x}_1^i, \hat{x}_1^j), \mathtt{max}(\hat{y}_1^i, \hat{y}_1^j),$
                      $\hat{c}^i \oplus \hat{c}^j, \hat{u}^i \times \hat{u}^j\ ) \ |$
                      $\hat{\mathbf{b}}^i \in \mathbf{B}_{d=0} \land \hat{\mathbf{b}}^j \in \mathbf{B}_{d=1}\}$
end if
$\mathbf{B}_{\text{number}} \leftarrow \mathtt{topk}(\mathbf{B}_{\text{number}})$
return $\mathbf{B}_{\text{number}}$
```

E. Data Augmentations

During training, we employ multi-scale training by randomly resizing the images to several predetermined scales. To further improve both the player and jersey number detection performance, we introduce three data augmentation methods specifically designed for the jersey number detection task in this section. The jersey number detection is highly dependent on the player detection performance. The more accurate player bounding boxes are predicted, the better digit localization performance is thanks to the proposed pose-guided branch. However, the digit classification is challenging due to many factors such as motion blur, clothing deformation, *etc*. We collected cropped player images from several soccer matches with annotations of players' bounding boxes, keypoints, and digit bounding boxes. However, the collected data does not cover enough scene variations and range of jersey numbers. A jersey number may correlate to its popularity and a player's position in certain sports, but the digit distribution suffers significant bias given the limited data. Sufficient data for training both player detection and digit classification is needed.

1) Pretraining: To create a general jersey number detection framework that works for most sports, we need more data besides our dataset. We incorporate the Street View House Number (SVHN) dataset [40] and COCO Keypoints dataset [41] for pretraining. SVHN contains images of numbers with annotated digit bounding boxes, and COCO contains images of persons with annotated bounding boxes and keypoints. Specifically, during each training iteration, we randomly select data from SVHN and COCO with equal probabilities. The backbone network is trained on both datasets for obtaining robust player and digit feature representations; RPN and player branch are only trained on COCO for generating robust person detection and keypoint estimation. As for the digit branch, the digit features are pooled from the ground-truth digit bounding boxes and used for training the digit classifier. During pretraining, the pose-guided branch is unused.

2) CopyPasteMix: As discussed in our previous work [30], our collected images are enlarged patches cropped from whole video frames. Each image contains at least one player with annotations. Inspired from recent work on data augmentation [74]–[78], we propose a data augmentation method called `CopyPasteMix` that provides more variability to the training data. It copies random number of training images and pastes onto a background image filled in black. The source images are resized randomly, and each image is pasted in order of largest to smallest onto a random location. If one image has an IoU over 0.5 with the previously pasted image, we regenerate the target location and retry. For any two images with IoU less or equal to 0.5, we perform a linear blending of the two images with equal weights. After all the sources images are pasted, we adjust the ground-truth annotations accordingly. We then train on the resulting synthetic image. By using `CopyPasteMix`, we have more training targets at different scales and locations per image. Empirically, the number of images used to construct the synthetic image is randomly selected between 1 and 5.

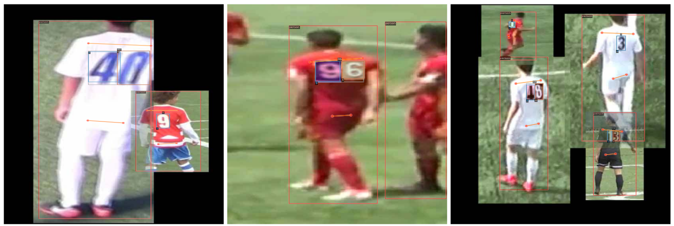
3) SwapDigit: We also design another augmentation method called `SwapDigit`. We borrow the data from SVHN, such that the RoI of a digit in our training images is randomly replaced with a cropped digit RoI in SVHN. By using `SwapDigit`, we effectively mitigate the lack of digit annotations problem such that each digit class can be trained with enough data for the digit classifier. It is worth noting that `CopyPasteMix` and `SwapDigit` can be used simultaneously for maximizing data efficiency and performance gain.

Among all the discussed data augmentation methods, only translation or scaling of bounding boxes are involved. The human body keypoints are changed accordingly with respect

TABLE I

DATASET STATISTICS. THE COLLECTED STATISTICS FROM THE SECOND LEFT TO THE RIGHTMOST COLUMN ARE: THE NUMBER (#) OF IMAGES, THE NUMBER OF ANNOTATED DIGITS, THE NUMBER OF ANNOTATED PLAYERS, THE NUMBER OF PLAYERS WITH ANNOTATED KEYPOINTS, THE MEAN AND STANDARD DEVIATION OF THE BOUNDING BOX SIZE OF PLAYERS AND DIGITS (IN PIXELS), AND THE BOUNDING BOX AREA RATIO OF DIGIT TO PLAYER

Video	# Images	# Digits	# Players	# Kpts	Image width	Image height	Player width	Player height	Digit width	Digit height	Area ratio
1	1030	1940	1323	1062	224.00 ± 0.00	224.00 ± 0.00	100.13 ±25.78	172.18 ± 27.71	20.06 ± 5.69	24.08 ± 6.59	0.03 ± 0.01
2	706	1347	768	732	173.42 ± 30.58	281.64 ±48.28	95.14 ±30.25	242.57 ± 54.12	15.58 ± 3.87	28.94 ± 6.01	0.02 ± 0.01
3	418	829	468	420	224.46 ± 37.58	389.36 ±80.58	154.35 ±46.72	343.98 ± 97.66	28.56 ±10.66	53.08 ±16.22	0.03 ± 0.01
4	1413	2180	1769	1472	225.23 ± 42.41	376.15 ±79.79	139.77 ±52.17	324.87 ±102.18	22.01 ± 7.43	44.69 ±12.62	0.02 ± 0.01
5	910	2779	1726	1720	640.00 ± 0.00	360.00 ± 0.00	82.38 ±23.07	160.83 ± 39.89	10.16 ± 2.97	19.71 ± 3.12	0.02 ± 0.01
Total	4477	9075	6054	5406	301.01 ±174.63	324.19 ±84.19	110.21 ±45.33	235.77 ±102.45	17.61 ± 8.36	31.06 ±14.81	0.02 ± 0.01



CopyPasteMix SwapDigit CopyPasteMix+SwapDigit

Fig. 3. Example augmented images (left to right) using `CopyPasteMix`, `SwapDigit`, and `CopyPasteMix+SwapDigit`.

to the players' bounding boxes. We do not want to undermine the geometric relationship between the human pose and digit locations, so no other transformation is carried out. Some examples of augmented images are shown in Figure 3. In Section IV, we show significant performance gains by using the proposed augmentation methods, and investigate the individual and combinatorial effects of the augmentations in Section IV-F. A theoretical analysis on the effectiveness of the proposed data augmentation methods is provided in the supplementary material.

IV. EXPERIMENTS

To validate the effectiveness of JEDE, we conduct experiments and compare with other state-of-the-art methods on our collected dataset as there is no publicly available dataset. We conduct evaluations on both jersey digit and number detection tasks. Following the standard COCO [41] metrics for object detection, we report the bounding box AP and AR (averaged precision and recall across IoU thresholds from 0.5 to 0.95 with an interval of 0.05), AP_{50}, AP_{75}, AR_{50}, and AP_{75} where the IoU threshold is denoted by the subscript.

A. Dataset

There is no publicly available jersey number dataset with instance-level annotations. To identify players in sports scenes, predicting the jersey numbers can be more robust than from other visual information like face and gait. It is natural to consider the jersey number detection as a top-down process where player localization provides robust prior information for digit localization. Human pose information can also be useful by providing implicit constraints on digit locations. Previous work [31], [32] only investigates image-level jersey number recognition which is not practical for real-world applications where multiple players are involved. Instead of performing player identity classification directly, digit detection provides more accurate visual cues of players' identities. To continue the advances towards the ultimate goal of automatic sports analysis, we introduce a new dataset that addresses three core research problems in detection for sports: player detection, player pose estimation, and digit detection.

We choose soccer and basketball, two of the most popular team sports, for creating the dataset. To collect data with sufficient variations, the game match videos are selected based on different jersey colors, jersey number colors, and jersey number fonts. The soccer data is collected from four matches. The recording device used is a single Canon XA10 video camera which is installed on a pole that is 15 feet high, and 10 to 20 feet away from the horizontal baseline of the soccer field. For better video qualities on jersey numbers, the camera operator is allowed to pan and zoom accordingly. Next, the video frames are enlarged by a factor of 2. An off-the-shelf person detector (*e.g.*, OpenPose [79]) is applied to get players' bounding boxes. The image is cropped around each bounding box with a padding of 150 pixels and a random shift within 20 pixels to create data variations. Besides the soccer sport, we also collect frames from one basketball match. To increase the diversity and add more challenging training data, the basketball frames are enlarged by a factor of 2 and divided into 4 large patches of equal size. After the data collection is completed, the images are labeled via VGG Image Annotator [80]. For each player in an image, we annotate its bounding box and legible digits. For players with digit annotations, 4 human body keypoints (left shoulder, right shoulder, left hip and right hip) are also annotated.

In total, there are 4477 labeled images, including annotations of 6054 players with 5406 of them are labeled with keypoints, and 9075 digits. We list the statistics of the collected dataset in Table I. There are large variations in scales within each collected video, and even larger across videos. The digit to player bounding box area ratio is only around 2% as shown in the table. The relatively small scale of digits makes jersey number detection even a more challenging task. Some example image and digit class distributions for each video are shown in Figure 4. The differences in image appearance and digit distribution are significant. For a fair evaluation on the unbalanced dataset, we perform k-fold cross validation where the training and testing data are divided by videos for examining the generalization power of JEDE.

B. Implementation Details

Our implementation is built upon the codebase Detectron2 [81] and PyTorch [82]. All experiments are conducted on a workstation with two Nvidia 1080 Ti GPUs. The model

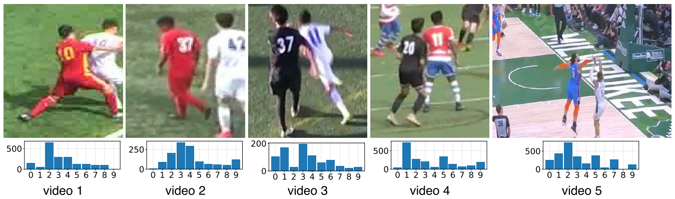


Fig. 4. Example images from the collected dataset for each video labeled in numerical ascending order. Images are resized for illustration. The second row shows the histogram of digit annotations for each video.

is trained in parallel and evaluated on a single GPU. We set the hyper-parameters following Mask R-CNN [39]. A fixed random seed is used, and no data balancing strategy is applied in all experiments for fair comparisons. We first perform 4-fold cross validation on the collected soccer data. Then, we conduct cross-domain evaluations by training on soccer and testing on basketball data, and vice versa.

We first implement a JEDE baseline with a ResNet-50-FPN backbone [37], [39]. The positional embeddings are not included, and only multi-scale training is used during training. We then implement an augmented version of JEDE baseline by adding the positional embeddings and using the proposed augmentation methods. Specifically, we pretrain the model on COCO and SVHN simultaneously with equal sampling probabilities, where `CopyPasteMix` with a maximum of 5 images is applied for SVHN data. Then, we train on our dataset using both `CopyPasteMix` and `SwapDigit` with the same hyper-parameters of the baseline model.

1) Training: The input images are resized such that their longer edge is no more than 800 pixels. For the short edge, we use multi-scale training such that a random scale is selected during each iteration. We follow some common pixel values for the shorter edge [38], [39] to choose from: 480, 512, 544, 576, 608, and 640. For each image, there are 512 sampled player RoIs with a ratio of 1:3 of positive samples (RoIs with IoU $\geq$ 0.5 over GT player bounding boxes) to negatives. For each detected player, we sample 100 digit proposals from the pose-guided branch. Within the predicted center heatmaps of each player, we sample the top 50 locations with the highest scores as detected centers, and randomly sample other 50 from the rest as negative samples. The maximum number of sampled digit proposals per image is set as 256. We train the whole framework for 50k iterations, with a mini-batch size of 4 (2 images per GPU on 2 GPUs) and learning rate of 0.0002. The learning rate is decreased by 10 at the 40k-th iteration. We use a weight decay of 0.0001 and momentum of 0.9.

2) Inference: The test images are resized such that their longer edge is no more than 800 pixels while the shorter edge is at least 480 pixels. Top 1000 player proposals from RPN are kept. We run the player branch on these proposals followed by non-maximum suppression to get the top 100 player bounding boxes ranked by classification scores. The keypoints regression and pose-guided branch are applied to these selected boxes. For each player instance, we keep the top 20 digit proposals. Finally, all the digit proposals are fed into the digit branch to obtain the final classifications and bounding boxes, followed by non-maximum suppression.

TABLE II

JERSEY DIGIT DETECTION RESULTS

Fold	Method	AP	AP_{50}	AP_{75}	AR	AR_{50}	AR_{75}
1	Faster R-CNN	25.26	42.01	27.52	40.00	41.42	41.42
	Cascade R-CNN	25.10	35.41	31.79	39.97	40.32	40.32
	TridentNet	26.06	40.30	31.68	38.89	40.85	40.85
	Pose-guided R-CNN	27.03	44.05	30.55	40.22	42.15	42.15
	JEDE Baseline	30.28	55.96	28.98	43.96	44.67	44.67
	JEDE Augmented	**51.08**	**80.10**	**60.09**	**64.24**	**65.33**	**65.33**
2	Faster R-CNN	49.61	71.79	61.05	67.48	69.09	69.09
	Cascade R-CNN	54.97	76.00	67.00	69.86	70.60	70.60
	TridentNet	56.07	77.68	70.79	69.97	70.60	70.60
	Pose-guided R-CNN	48.33	71.01	60.23	65.43	67.07	67.07
	JEDE Baseline	49.46	73.45	61.31	66.6	67.47	67.47
	JEDE Augmented	**59.70**	**86.28**	**74.73**	**70.89**	**71.23**	**71.23**
3	Faster R-CNN	55.79	76.69	67.44	64.74	67.30	67.30
	Cascade R-CNN	**60.01**	77.93	**72.53**	**72.27**	**74.59**	**74.59**
	TridentNet	57.50	75.35	70.07	67.09	68.54	68.54
	Pose-guided R-CNN	46.74	75.32	53.76	55.89	57.14	57.14
	JEDE Baseline	48.80	77.65	54.38	57.85	59.33	59.33
	JEDE Augmented	56.85	**88.44**	65.76	64.70	66.28	66.28
4	Faster R-CNN	49.17	65.64	60.86	66.33	68.57	68.57
	Cascade R-CNN	62.09	76.18	73.59	72.91	**74.52**	**74.52**
	TridentNet	59.29	75.94	72.65	68.30	69.16	69.16
	Pose-guided R-CNN	51.45	71.27	63.33	65.67	67.01	67.01
	JEDE Baseline	53.32	73.80	65.46	66.93	68.02	68.02
	JEDE Augmented	**67.15**	**91.65**	**82.81**	**73.02**	74.25	74.25

C. Digit Detection Results

In this sub-section, we evaluate the jersey digit detection performance with thorough comparisons of JEDE to pose-guided R-CNN [30], and state-of-the-art object detectors such as Faster R-CNN [38], Cascade R-CNN [83], and TridentNet [84]. Serving as competitive methods, Cascade R-CNN includes a sequence of detectors that improve the detection quality, while TridentNet is robust to object scale variations.

The results of JEDE models are listed and compared with other methods in Table II. For each cross validation result, the fold number indicates which test video is used, *e.g.* fold 1 means that we train on videos 2, 3, and 4, then test on video 1. Our models achieve the state-of-the-art results with substantial improvements. JEDE baseline already outperforms Faster R-CNN over most metrics, and the augmented JEDE further improves the results. Fold 1 and 4 involve less training data and more testing data, and we see more performance gains for JEDE. For example of fold 1, JEDE baseline has 5.02 points improvement in AP over Faster R-CNN, and 3.25 points over pose-guided R-CNN. The results prove the effectiveness of the pose-guided branch. The digit localization can be improved by extracting contextual information from the pose features with limited training data. Moreover, the model trained with augmentations achieves massive gains such that AP is doubled compared with Faster R-CNN. Both precision and recall are dramatically improved over the baseline model, demonstrating that we have more accurate bounding box predictions and better digit classifiers. This highlights that our proposed augmentation methods are capable of diversifying the training data without any extra cost. It can be seen that the results of JEDE for fold 3 are slightly lower than Cascade R-CNN. Since fold 3 only contains 418 testing images, whose size is relatively small compared with other folds, it is a natural disturbance for a such small performance gap. The augmented JEDE still outperforms Cascade R-CNN by 10.51 AP_{50}.

TABLE III

JERSEY NUMBER DETECTION RESULTS

Fold	Method	AP	AP_{50}	AP_{75}	AR	AR_{50}	AR_{75}
1	Mask TextSpotter V3	-	2.43	0.25	-	35.84	3.72
	SwinTextSpotter	5.28	6.85	6.44	14.71	16.55	16.58
	Pose-guided R-CNN	13.15	18.35	14.67	27.43	28.02	28.02
	JEDE Baseline	18.07	26.87	20.93	35.69	35.99	35.99
	JEDE Augmented	**37.12**	**50.69**	**45.84**	**50.29**	**50.63**	**50.63**
2	Mask TextSpotter V3	-	0.77	0.11	-	11.49	1.65
	SwinTextSpotter	10.54	12.27	12.00	14.64	18.54	18.54
	Pose-guided R-CNN	32.23	42.27	39.73	47.85	48.84	48.84
	JEDE Baseline	36.02	43.59	42.69	50.26	50.61	50.61
	JEDE Augmented	**36.12**	**45.60**	**44.08**	**54.61**	**54.65**	**54.65**
3	Mask TextSpotter V3	-	1.80	0.48	-	29.24	7.88
	SwinTextSpotter	11.29	13.83	13.49	23.53	26.55	26.63
	Pose-guided R-CNN	43.12	55.12	52.36	58.42	60.12	60.12
	JEDE Baseline	45.17	59.03	55.42	**62.38**	**63.83**	**63.83**
	JEDE Augmented	**48.27**	**61.72**	**57.51**	57.85	59.31	59.31
4	Mask TextSpotter V3	-	0.21	0.12	-	5.65	3.27
	SwinTextSpotter	9.73	12.82	11.35	17.33	19.64	19.87
	Pose-guided R-CNN	33.76	40.92	38.45	44.77	46.58	46.58
	JEDE Baseline	36.35	44.42	42.84	47.95	48.27	48.27
	JEDE Augmented	**42.87**	**52.82**	**51.53**	**53.18**	**54.19**	**54.19**

D. Jersey Number Detection Results

For jersey number detection evaluations, we use the same metrics as they are in digit detection. Naturally, the precision is less than it is in digit detection, since both digits must be detected corrected for a two-digit jersey number to be counted as a true positive detection. We report the results of JEDE models, and compare with the pose-guided R-CNN and the state-of-the-art scene text detection frameworks Mask TextSpotter V3 [45], and SwinTextSpotter [47]. We use their open-sourced codebases and modify the output number of classes to 11 ("0" -"9" and "background"). The model Mask TextSpotter V3 is initialized with their pre-trained weights, and SwinTextSpotter is trained from scratch. Both models are trained with the same settings as in Section IV-B.

Table III compares our results to Mask TextSpotter V3, SwinTextSpotter, and pose-guided R-CNN. For Mask TextSpotter V3, we only report AP_{50}, AP_{75}, AR_{50}, and AR_{75}. JEDE outperforms other methods in every fold by a large margin. For example in fold 1, the baseline JEDE achieves 26.87 AP_{50}, which shows 24.44 points improvement over Mask TextSpotter V3 and 20.02 points improvement over SwinTextSpotter. The augmented JEDE further pushes the performance with 50.69 AP_{50}. It is worth noting that both Mask TextSpotter V3 and SwinTextSpotter have poor AP but fair AR results. This indicates that these methods are able to detect jersey number bounding boxes but barely classify them correctly. To examine this behavior, we conduct experiments by modifying our detection branches similar to commonly used scene text detection methods. Specifically, we detect jersey number bounding boxes directly where the union of digit bounding boxes is considered for a two-digit number. Then we add the sequence modeling (*e.g.*, Bidirectional LSTM [85], [86]) for pooled jersey number features as in [45], [87], [88]. The number classification is performed per column of the features that can be trained using Connectionist Temporal Classification (CTC) [89]. This modified model achieves 13.71 AP_{50} which is much lower than the JEDE baseline. The unsatisfied results can be justified by the sequence modeling of jersey numbers. Scene text detection relies on lexicons and contextual information between characters, while there is no such dependence between digits in a jersey number. Moreover, there are not enough jersey numbers for training a robust recurrent model for sequence classification. Traditional scene text detection frameworks are probably not suitable for directly detecting jersey numbers unless heavy adaptations are applied.

In Figure 5, we provide the qualitative comparisons between JEDE and other methods. Only detections with a confidence score over 0.2 are shown. JEDE models consistently perform better under different conditions. The pose-guided branch provides a strong regularization for the digit locations as seen from columns 7 and 8 in the figure. JEDE models predict accurate digit bounding boxes, while other methods predict wrong locations on arms or hips. It can also be seen that JEDE is more robust to low-resolution images (columns 1 and 2). The last column shows a failure case where an extreme pose is presented: "9" is not recognized correctly due to rare deformations. JEDE Augmented still predicts an accurate digit bounding box, while other methods generate some false positive detections. It further demonstrates that the pose-guided branch provides more reliable digit proposals. Nevertheless, there are some failure cases for the proposed method as well. For example, in the last row of Figure 5: in the 4-th image, "46" is not detected due to partial occlusion; in the 9-th image, "39" is not recognized since "single-digit" is predicted from the jersey number length classifier.

E. Cross-Domain Results

It is well known that deep learning vision models are highly dependent on large labeled datasets. Even though a trained model can success on one dataset, its performance often declines significantly on new data or new domain. This problem is referred to as dataset shift [90] where training and testing data distributions are different. It can be observed from Tables II and III that the detection performance differs among each testing fold, although both training and testing data are collected from soccer matches. The visual changes between soccer matches are substantial, not to mention the domain shift from soccer to other sports, and vice versa. Thus, we perform two cross-domain experiments between the soccer and basketball data, without using any transfer learning [91] or domain adaptation [92] technique.

The first experiment is training on soccer and testing on basketball data ($S \rightarrow B$). During testing, we resize the input image such that the longer edge is no more than 1600 pixels, and the short edge is at least 960 pixels. The results are shown in Table IV. Although models trained without augmentations suffer from the domain shift and limited data, JEDE Baseline still outperforms other methods significantly. With augmentations, JEDE achieves much better performance with 30.34 digit AP_{50} and 21.60 number AP_{50}. We further show the qualitative results in Figure 6. For SwinTextSpotter, we also use their pre-trained model to perform text detection directly as shown in the 4-th row of Figure 6. JEDE Augmented still achieves the best detection results overall. Pre-trained SwinTextSpotter can achieve fair detection results, but cannot distinguish between jersey numbers and other texts. In some difficult conditions as shown in Figure 6, JEDE may fail, such as in the second

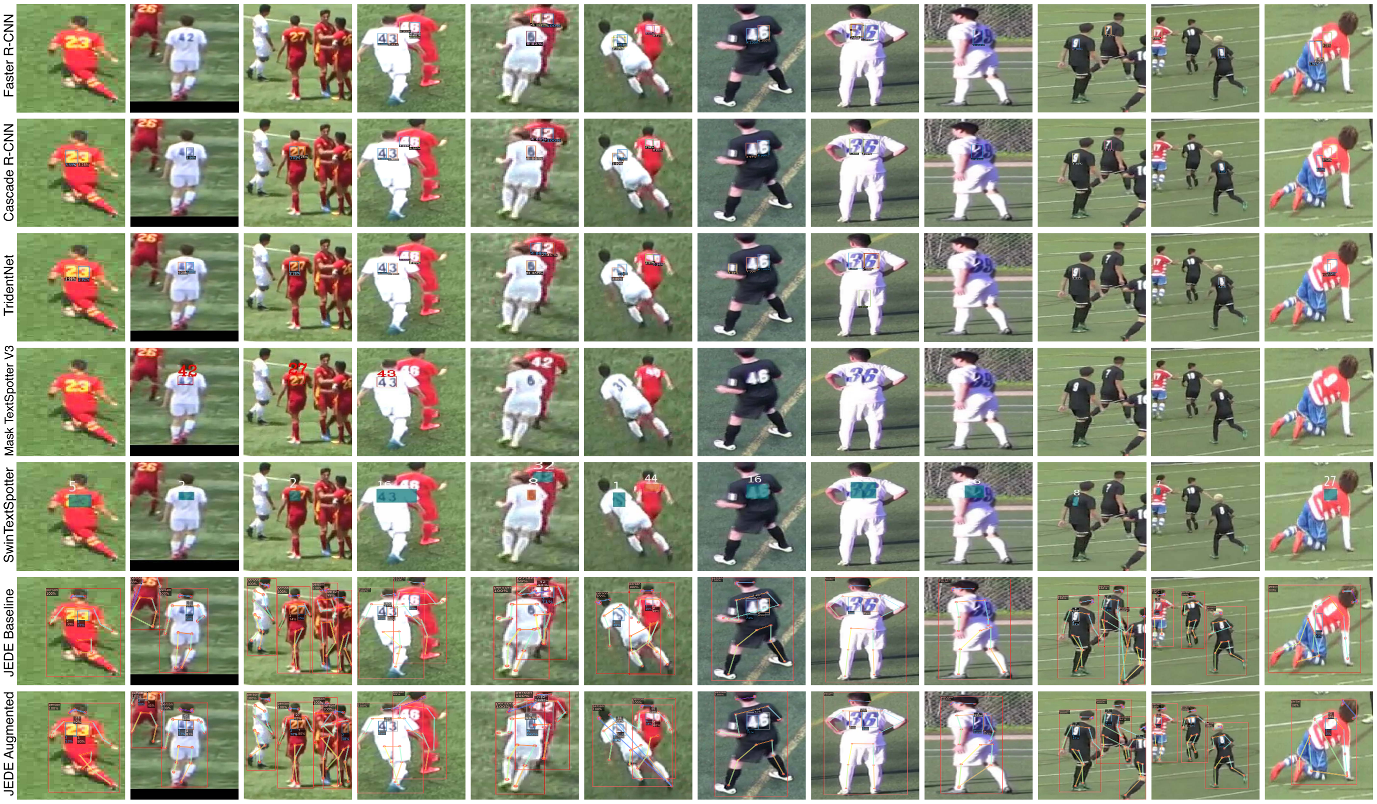


Fig. 5. Qualitative comparisons between JEDE and other methods. Each bounding box is labeled with the predicted class and score, if available. The digit class is labeled under the left bottom corner of the bounding box for all methods (rows 1, 2, 3, 6, 7), and the jersey number is labeled above the box for JEDE models (rows 6, 7). Only jersey numbers are labeled for Mask TextSpotter V3 and SwinTextSpotter. Images are resized for illustration.

TABLE IV

PERFORMANCE COMPARISON FOR S→B TASK

Method	AP	AP_{50}	AP_{75}	AR	AR_{50}	AR_{75}
Faster R-CNN	0.00	0.01	0.01	0.12	0.15	0.16
Cascade R-CNN	0.01	0.01	0.01	0.16	0.19	0.19
TridentNet	0.00	0.00	0.00	0.02	0.02	0.02
Pose-guided R-CNN	0.04	0.32	0.00	0.25	0.28	0.28
JEDE Baseline	0.09	0.46	0.00	0.36	0.37	0.37
JEDE Augmented	**10.17**	**30.34**	**3.91**	**19.97**	**24.06**	**24.07**
Mask TextSpotter V3	-	0.00	0.00	-	0.08	0.00
SwinTextSpotter	0.05	0.08	0.08	0.04	0.05	0.05
Pose-guided R-CNN	0.01	0.07	0.00	0.05	0.05	0.05
JEDE Baseline	0.02	0.09	0.00	0.07	0.07	0.07
JEDE Augmented	**9.90**	**21.60**	**7.71**	**20.05**	**22.56**	**22.56**

TABLE V

PERFORMANCE COMPARISON FOR B→S TASK

Method	AP	AP_{50}	AP_{75}	AR	AR_{50}	AR_{75}
Faster R-CNN	1.49	3.80	0.98	3.04	3.46	3.46
Cascade R-CNN	1.46	2.94	1.33	3.85	4.01	4.01
TridentNet	0.14	0.65	0.00	0.84	0.84	0.84
Pose-guided R-CNN	1.51	4.02	1.07	3.60	3.83	3.83
JEDE Baseline	0.63	2.54	0.15	1.94	2.11	2.11
JEDE Augmented	**9.09**	**32.54**	**1.85**	**15.81**	**16.29**	**18.61**
Mask TextSpotter V3	-	0.98	0.16	-	**17.96**	3.03
SwinTextSpotter	0.00	0.00	0.00	0.06	0.10	0.11
Pose-guided R-CNN	0.20	0.78	0.05	0.48	0.50	0.50
JEDE Baseline	0.23	0.87	0.06	0.56	0.58	0.58
JEDE Augmented	**8.40**	**20.24**	**5.22**	**13.97**	14.02	**14.02**

image, “77” is not detected due to motion blur; and in the 4-th image, “12” is not detected due to small digit scale.

The other experiment is training on basketball and testing on soccer data (B→S). During testing, we resize the input image such that the longer edge is no more than 400 pixels, and the short edge is at least 240 pixels. `CopyPasteMix` is not applied for JEDE Augmented. The results are shown in Table V. We observe worse results compared with the ones on S→B, since there is much less training data (but more testing data) for basketball domain (Table I). Nevertheless, JEDE Augmented still achieves the best results among all the methods with number AP_{50} being over 20 times better than Mask TextSpotter V3. By comparing the JEDE Baseline and Augmented results, it implies that `SwapDigit` significantly mitigate the problem of limited training data, suggesting the high practicality of the proposed data augmentation strategies. The cross-domain results demonstrate that JEDE models have better generalizability over other methods.

F. Ablation Study

We conduct a number of ablations to analyze JEDE, and show the digit detection results in this section unless otherwise specified. All the experiments are based on JEDE Baseline. Best results are in bold.

1) Backbone: Table VI shows a comparison of JEDE results, number of parameters (in Million), giga floating point operations (GFlops), and inference frame per second (FPS) of various backbones. The metrics are measured on a Nvidia GTX 1080 Ti GPU with a batch size of 1, and the GFlops and FPS are averaged over all testing images. We observe that the results do not benefit from much deeper networks such as ResNet-101 and ResNeXt-101 due to overfitting on limited

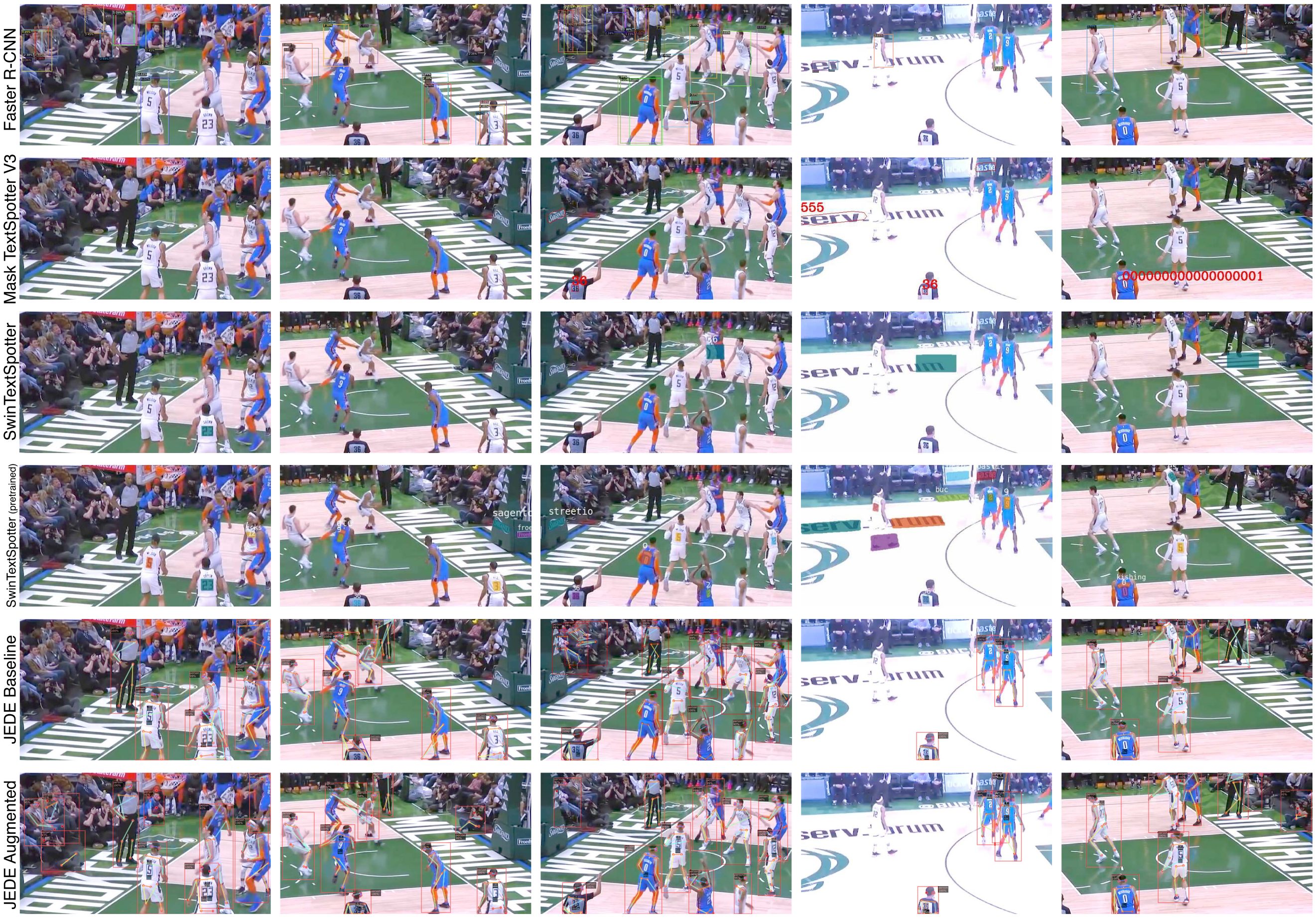


Fig. 6. Qualitative comparison on the cross-domain task S→B. Faster R-CNN, Mask TextSpotter V3, and SwinTextSpotter achieve poor performances with false positive detections that predict players, body parts, or texts as digits. JEDE Baseline performs well on player detection and pose estimation with fair digit classification performance. JEDE Augmented is much more robust to recognize digits thanks to the proposed data augmentation methods.

TABLE VI
ABLATION ON BACKBONE NETWORKS

Backbone	AP	AR	Params	GFlops	FPS
ResNet-18	11.58	23.51	**67.4M**	52.5	**24.55**
ResNet-50	25.15	38.80	80.5M	60.9	20.37
ResNet-50[†]	**25.33**	**40.14**	80.5M	**52.0**	22.68
ResNet-101	24.50	38.85	93.2M	77.7	17.54
ResNeXt-101-32x8d	19.62	35.98	144.0M	119.2	10.76

[†] The smallest anchors of size 32 are removed.

training data. The JEDE Augmented with ResNet-101 slightly outperforms it with ResNet-50 by 0.08 as shown in Table II. It indicates that the proposed data augmentation methods are more effective than increasing the model size. We also observe notable speed and performance improvements by removing the anchor size of 32. Since small proposals are simple negative examples (background) that do not contribute much to the loss, the model acquires more effective proposals during training by removing small anchors. We also compute the inference speed of Mask TextSpotter V3 and SwinTextSpotter with the same settings since the jersey number detection task is time-sensitive. On average, Mask TextSpotter V3 achieves 5.04 FPS and SwinTextSpotter achieves 2.46 FPS, while JEDE with ResNet-50 backbone achieves 22.68 FPS as a comparison.

TABLE VII
GT BOUNDING BOX MINIMUM OVERLAP: SMALLER VALUE GIVES BETTER RESULTS

min_iou	AP	AR
0.1	**26.46**	**38.95**
0.3	25.79	38.69
0.5	26.12	38.77

2) GT Minimum IoU: As discussed in Section III-C.2, the hyper-parameter min_iou controls the Gaussian blob size of the digit centers. The larger min_iou is, the smaller the blob size is. Table VII shows the results of using different values of min_iou. It can be observed that smaller min_iou gives better results because more training samples are included while training center heatmaps regression.

3) GT Heatmap Spatial Size: The spatial size of the GT heatmaps is important for predicting the center locations of digits. Hypothetically, higher resolution the heatmaps have, more accurate the prediction will be due to less coordinate discretization. We perform the sensitivity analysis of the heatmap size and show the results in Table VIII. The input player features are interpolated to the desired size. As expected, using the largest size achieves slightly better AP. However, simply up-sampling features does not improve the results significantly. We further perform the experiment by pooling 56 × 56 player

TABLE VIII

GT Heatmap Spatial Size: Larger Size Gives Better Results

Size	AP	AR
14×14	24.15	35.68
28×28	22.90	36.06
56×56	24.53	35.73
$56 \times 56^{\ddagger}$	**28.82**	**39.96**

‡ Directly pooled without interpolation.

TABLE IX

Input to Pose-Guided Branch: Fusion of Both Features Gives Better Results

Input Features	AP	AR
Keypoint	12.77	24.02
Player	20.81	31.64
Both	**23.55**	**34.81**

TABLE X

Feature Fusion Methods: Concatenation Gives Better Results

Fusion method	AP	AR
Multiply	20.79	33.50
Sum	21.40	33.76
Concatenate	**23.07**	**36.43**

TABLE XI

Positional Embeddings (PE): Concatenation w/ Keypoint Heatmaps Gives Better Results

Concat w/	AP	AR
No PE	23.07	36.43
Fused features	24.81	35.89
Keypoint heatmaps	**25.29**	**37.96**

TABLE XII

Normalization Layers: GN Provides Better Results Using a Small Batch Size

Norm	AP	AR
None	21.44	32.14
BN	22.12	34.58
GN	**27.73**	**42.51**

features directly. As a result, AP is significantly improved by 4.67 compared with using 14×14.

4) Keypoint vs. *Player Features:* We investigate the effectiveness of the pose-guided branch. There are two default input features, the keypoint heatmaps and pooled player features. We conduct ablation experiments by 1) only using the keypoint features, 2) only using the player features, and 3) using the fusion of both features. The results are shown in Table IX. Using both features significantly outperforms only using one set of features. By adding the keypoint features, AP is improved by 2.74 points. It validates the effectiveness of the pose-guided branch that pose information is helpful for localizing digits.

5) Fusion Methods: We also investigate the fusion methods of the player features and keypoint features, and show the results in Table X. Concatenation has better performance because it provides more feature transformations for the branch to reduce the semantic difference between the player features and the keypoint heatmaps.

6) Positional Embeddings: We examine the effectiveness of concatenating positional embeddings (PE) with different features, fused features or keypoint heatmaps, and show the results in Table XI. Adding the positional embeddings improves the results, and concatenating with keypoint heatmaps outperforms with fused features by 0.48 AP. One possible reason is that both PE and keypoint heatmaps provide spatial information and make learning easier.

7) Normalization Layers: We further perform an ablation study on the normalization layers used in the pose-guided branch. Two commonly used feature normalization methods are selected, namely Batch Normalization (BN) [93] and Group Normalization (GN) [94], and the results are shown in Table XII. The results are improved by using normalization layers, and GN performs better than BN due to training with a small batch size of 4. We expect that better results can be achieved by using a larger batch size.

8) Number Length Classification: As discussed in Section III-D.2, we use a MLP to predict the jersey number length. We investigate three possible features to be considered as its input: center heatmaps, the second last features in the center prediction head (features before the center heatmaps), and the fused features. The jersey number detection results are shown in Table XIII. Using the 2nd last features gives the best AP and AR. Using the center heatmaps has worse performance due to the loss of contextual information of the player, which is useful for number length classification.

9) Digit RoI Pooling Resolution: One of the hyper-parameters of the digit branch is the digit RoI pooling resolution. We conduct experiments with different resolutions and shown the results in Table XIV. It can be observed that larger pooling size gives better performance. Higher resolution provides more fine-grained visual features that are helpful for differentiating similar digits like “1” and “7”.

10) Augmentations: Besides the ablation on architectures, we also perform the effects of data augmentations on JEDE, including random crop with a maximum of 30% crop rate, pre-train on COCO and SVHN, `CopyPasteMix`, and `SwapDigit`. For JEDE Baseline, no augmentation is conducted to balance the uneven distribution of digits for fair comparison. The results are shown in Table XV. Each data augmentation method improves the results compared to JEDE Baseline. `CopyPasteMix` achieves the largest gain of AP by 17.80, while random crop only improves AP by 1.72. Moreover, combining `CopyPasteMix` and `SwapDigit` with pre-trained weights gives the best AP gain of 28.36, and AR gain of 31.21. The results demonstrate that data augmentation is one of the key factors for improving the performance given limited data. For jersey number detection task, our specially designed augmentation methods have stronger regularization capability than general methods like random crop.

Better results could be achieved by re-sampling [95], but the testing data distribution will still be very different from

TABLE XIII

ABLATION ON THE INPUT FEATURES FOR NUMBER LENGTH CLASSIFICATION

Input features	AP	AR
Center heatmaps	11.11	20.97
2nd last features	**12.75**	**23.80**
Fused features	12.66	20.16

TABLE XIV

DIGIT ROI POOLING RESOLUTION: LARGER RESOLUTION GIVES BETTER RESULTS

Pooling Res.	AP	AR
7 × 7	20.79	35.34
14 × 14	23.19	35.22
28 × 28	**24.99**	**36.60**

TABLE XV

ABLATION ON DATA AUGMENTATIONS

Random Crop	Pre-train	`CopyPasteMix`	`SwapDigit`	AP	AR
				21.44	32.14
✓				23.16	35.90
	✓			30.37	40.73
		✓		39.24	57.13
			✓	30.36	44.57
	✓	✓		45.95	60.42
	✓	✓	✓	**49.80**	**63.35**

training data in the cross validation. This is the reason why `SwapDigit` is implemented for mitigating the problem of unbalanced data. By using the SVHN dataset which is well-balanced for digit annotations, the detection performance is much improved for those digit classes that are trained insufficiently. We provide a per-class comparison of JEDE Baseline and Augmented as shown in Figure 7. It can be observed that larger relative improvements are achieved for those digit classes that have a low AP.

We also evaluate and compare the player bounding box detection mean average precision ($\mathrm{AP}^{\mathrm{player}}$) and keypoint detection mean average precision ($\mathrm{AP}^{\mathrm{kpts}}$) for each experiment following the standard metrics [41]. We only report on the four categories of keypoints for which annotations are available in our dataset. The results are shown in Table XVI. It can be observed that JEDE Augmented achieves better keypoint detection performance in each experiment, especially when less training data is available in the B→S task. The testing images for each fold are cropped frames with low variances, and thus the data distribution difference between training and testing are small. With data augmentations, we provide a stronger regularization by training with more instances at different scales. As a result, JEDE Augmented receives less training data that are similar to the testing data. This suggests why JEDE Augmented achieves slightly lower $\mathrm{AP}^{\mathrm{player}}$ in some experiments compared with the baseline.

11) Toward a Universal Jersey Number Detector: To fully exploit the capability of JEDE, we develop a slightly larger model using the best hyper-parameters found in the ablation studies. We use larger input image size with a maximum of 1589 pixels for the longer edge. We first pre-train the model on SVHN and COCO, and then train using `CopyPasteMix` and `SwapDigit`. To prevent "forgetting" [96] the pre-trained weights for player detection, we train on COCO and our whole dataset (including soccer and basketball data) concurrently with an equal sampling probability for 150k iterations (3× longer than all other experiments). The learning rate is decreased by 10 at the 120k-th iteration. Since we use all the data for training, we only show the qualitative results on images collected from the Internet. We choose several popular team sports for visualizations as in Figure 8. The qualitative results demonstrate the remarkable generalizability of JEDE across different sports, even though is it only trained on soccer and basketball domains. There are limitations of the proposed framework as shown in the last image in the figure, where the numbers on the players' heads are not detected. For uncommon poses and digit locations which do not present during training, JEDE can only reject low-confidence detections.

TABLE XVI

COMPARISON OF JEDE BASELINE AND AUGMENTED ON PLAYER DETECTION AND HUMAN POSE ESTIMATION RESULTS

Experiment	Method	$\mathrm{AP}^{\mathrm{player}}$	$\mathrm{AP}^{\mathrm{kpts}}$
Fold 1	JEDE Baseline JEDE Augmented	76.52 **80.44**	94.88 **97.13**
Fold 2	JEDE Baseline JEDE Augmented	**81.44** 79.57	99.45 **99.58**
Fold 3	JEDE Baseline JEDE Augmented	**83.29** 75.49	91.27 **96.31**
Fold 4	JEDE Baseline JEDE Augmented	**80.59** 76.45	96.45 **98.03**
S→B	JEDE Baseline JEDE Augmented	**41.76** 41.42	49.61 **52.46**
B→S	JEDE Baseline JEDE Augmented	61.83 **70.41**	65.86 **83.05**

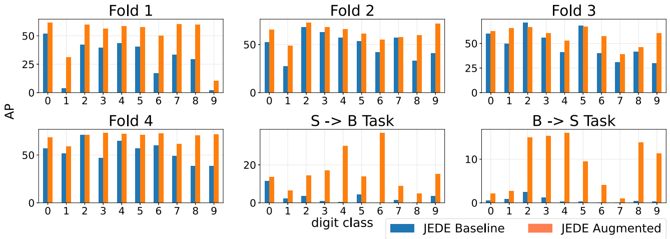


Fig. 7. Per-class AP comparisons between JEDE Baseline and Augmented for each fold and cross-domain task.

12) Limitations and Solutions: JEDE achieves great performances on player detection, player pose estimation, and jersey number detection. However, the detections are not perfect as shown in the visualizations. We summarize the limitations and their solutions as follows:

- Our approach is data-driven, and the performance still suffer from the lack of extensive data. If more annotations on sports-related poses and jersey numbers are available, better results can be achieved.
- The jersey number detection performance relies on the accuracy of jersey number length prediction. If the length is predicted incorrectly, the predicted jersey number will be wrong. As discussed in Section IV-D, if more training data of digits are provided, the number length predictor

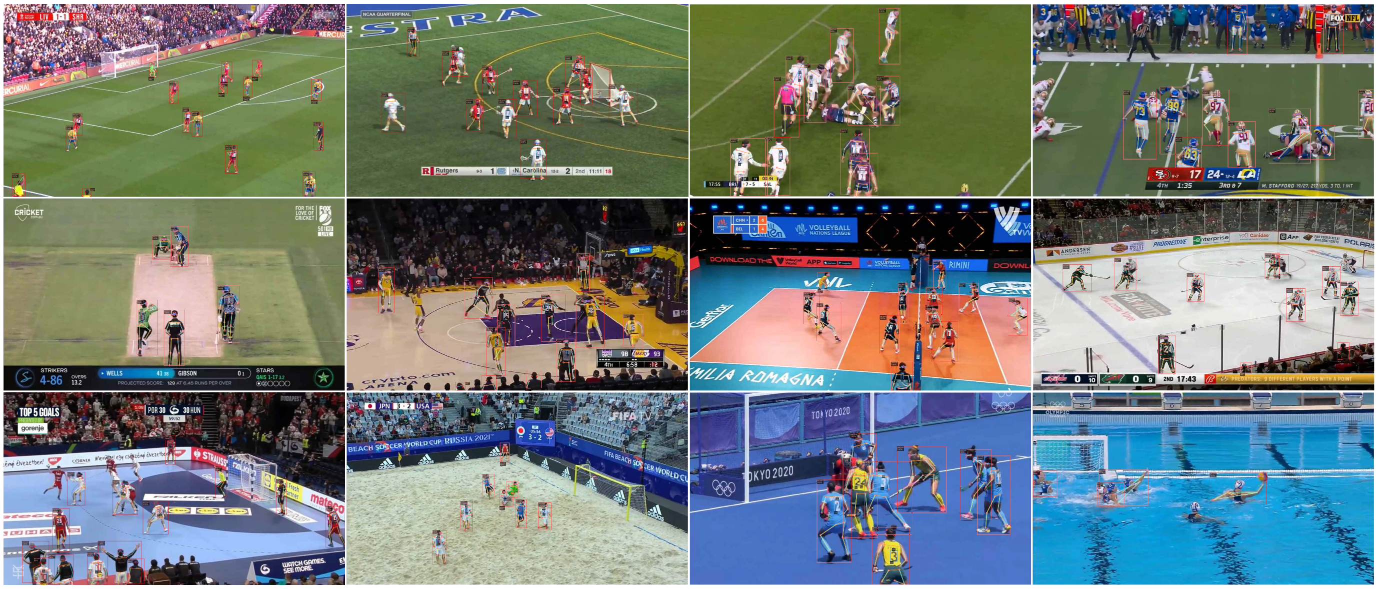


Fig. 8. Qualitative results for images in the wild. Sports from left to right, top to bottom are: soccer, lacrosse, rugby, American football, cricket, basketball, volleyball, ice hockey, handball, beach soccer, hockey, and water polo.

can be replaced with a sequence decoder like the one used in Mask TextSpotter V3 [45] and SwinTextSpotter [47].

- It is difficult for JEDE, object detection, or scene text detection framework to handle blur, occlusion, and small scale. JEDE can be deployed for real-time sports analysis. If the detection fails in a particular frame, we can still obtain correct detections in future frames. Moreover, in future work, JEDE can be extended for video analysis by integrating tracking for handling fast-changing conditions in sports fields.

V. Conclusion

In this work, a universal jersey number detector (JEDE) was proposed as an end-to-end solution for automated sports analysis that performs player detection, human pose estimation, jersey digit detection, and jersey number detection simultaneously. A dataset was collected that consists of 4477 images from soccer and basketball matches with annotations of 6054 player bounding boxes, 5406 poses, and 9075 digit bounding boxes. Exhaustive evaluations and comparisons were performed on this dataset. By conditioning digit detection on player's features and pose information, JEDE outperformed the state-of-the-art methods by a large margin. Moreover, to overcome the problem of insufficient data, data augmentation techniques `CopyPasteMix` and `SwapDigit` were proposed that significantly improved the results without extra inference cost. Extensive ablation studies were performed that showed how individual modules, hyper-parameters, and augmentations affect the performance of jersey number detection. Finally, the strong generalization capability of the proposed framework was demonstrated by showing the superior qualitative results across many sport domains.

Acknowledgment

The authors would like to thank Ashwini Shandilya and Eric Ebert from SEVAai Inc. for their support, Don Ebert for providing the dataset that is used in this paper, and their colleague Alex Shin for providing image annotations.

References

[1] *Sportlogiq*. Accessed: Mar. 1, 2022. [Online]. Available: https://sportlogiq.com/

[2] *S. Spectrum*. Accessed: Mar. 1, 2022. [Online]. Available: https://www.secondspectrum.com/

[3] F. Chen and C. De Vleeschouwer, "Formulating team-sport video summarization as a resource allocation problem," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 2, pp. 193–205, Feb. 2011.

[4] M. Tavassolipour, M. Karimian, and S. Kasaei, "Event detection and summarization in soccer videos using Bayesian network and Copula," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 291–304, Feb. 2013.

[5] Z. Wang, J. Yu, and Y. He, "Soccer video event annotation by synchronization of attack–defense clips and match reports with coarse-grained time information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 5, pp. 1104–1117, May 2016.

[6] R. Li and B. Bhanu, "Fine-grained visual dribbling style analysis for soccer videos with augmented dribble energy image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 2439–2447.

[7] J. Wang *et al.*, "Deep 3D human pose estimation: A review," *Comput. Vis. Image Understand.*, vol. 210, Sep. 2021, Art. no. 103225.

[8] Y. Yuan, S.-E. Wei, T. Simon, K. Kitani, and J. Saragih, "SimPoE: Simulated character control for 3D human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7159–7169.

[9] L. Jin *et al.*, "Single-stage is enough: Multi-person absolute 3D pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, Jun. 2022, pp. 13086–13095.

[10] X. Shu, L. Zhang, G.-J. Qi, W. Liu, and J. Tang, "Spatiotemporal co-attention recurrent neural networks for human-skeleton motion prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3300–3315, Jun. 2021.

[11] R. Zhang, X. Shu, R. Yan, J. Zhang, and Y. Song, "Skip-attention encoder–decoder framework for human motion prediction," *Multimedia Syst.*, vol. 28, no. 2, pp. 413–422, Apr. 2022.

[12] P. Shukla *et al.*, "Automatic cricket highlight generation using event-driven and excitement-based features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1881–1889.

[13] R. Theagarajan, F. Pala, X. Zhang, and B. Bhanu, "Soccer: Who has the ball? Generating visual analytics and player statistics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1830–1838.

[14] J. Schrittwieser *et al.*, "Mastering Atari, Go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.

[15] K. Tuyls *et al.*, "Game plan: What AI can do for football, and what football can do for AI," *J. Artif. Intell. Res.*, vol. 71, pp. 41–88, May 2021.

[16] H.-C. Shih, "A survey of content-aware video analysis for sports," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 5, pp. 1212–1231, May 2017.

[17] M. Beetz, S. Gedikli, J. Bandouch, B. Kirchlechner, N. V. Hoyningen-Huene, and A. Perzylo, "Visually tracking football games based on TV broadcasts," in *Proc. Int. Joint Conf. Artif. Intell.*, 2007, pp. 2066–2071.

[18] W.-L. Lu, J.-A. Ting, K. P. Murphy, and J. J. Little, "Identifying players in broadcast sports videos using conditional random fields," in *Proc. CVPR*, Jun. 2011, pp. 3249–3256.

[19] M. Bertini, A. Del Bimbo, and W. Nunziati, "Player identification in soccer videos," in *Proc. 7th ACM SIGMM Int. Workshop Multimedia Inf. Retr.*, 2005, pp. 25–32.

[20] M. Bertini, A. Bimbo, and W. Nunziati, "Matching faces with textual cues in soccer videos," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2006, pp. 537–540.

[21] M. Bertini, A. Del Bimbo, and W. Nunziati, "Automatic detection of player's identity in soccer videos using faces and text cues," in *Proc. 14th Annu. ACM Int. Conf. Multimedia*, 2006, pp. 663–666.

[22] L. Ballan, M. Bertini, A. D. Bimbo, and W. Nunziati, "Soccer players identification based on visual local features," in *Proc. 6th ACM Int. Conf. Image Video Retr.*, 2007, pp. 258–265.

[23] Z. Mahmood, T. Ali, S. Khattak, L. Hasan, and S. U. Khan, "Automatic player detection and identification for sports entertainment applications," *Pattern Anal. Appl.*, vol. 18, no. 4, pp. 971–982, Nov. 2015.

[24] Q. Ye, Q. Huang, S. Jiang, Y. Liu, and W. Gao, "Jersey number detection in sports video for athlete identification," *Proc. SPIE*, vol. 5960, pp. 1599–1606, May 2006.

[25] M. Šari, H. Dujmi, V. Papi, and N. Roži, "Player number localization and recognition in soccer video using HSV color space and internal contours," *Int. J. Elect. Comput. Eng.*, vol. 2, no. 7, pp. 1408–1412, 2008

[26] S. Messelodi and C. M. Modena, "Scene text recognition and tracking to identify athletes in sport videos," *Multimedia Tools Appl.*, vol. 63, no. 2, pp. 521–545, Mar. 2013.

[27] C.-W. Lu, C.-Y. Lin, C.-Y. Hsu, M.-F. Weng, L.-W. Kang, and H.-Y.-M. Liao, "Identification and tracking of players in sport videos," in *Proc. 5th Int. Conf. Internet Multimedia Comput. Service*, 2013, pp. 113–116.

[28] K. Akila, S. Chitrakala, and S. Vaishnavi, "Survey on illumination condition of video/image under heterogeneous environments for enhancement," in *Proc. 3rd Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Jan. 2016, pp. 1–7.

[29] K. Okuma, D. G. Lowe, and J. J. Little, "Self-learning for player localization in sports video," 2013, *arXiv:1307.7198*.

[30] H. Liu and B. Bhanu, "Pose-guided R-CNN for Jersey number recognition in sports," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 2457–2466.

[31] S. Gerke, K. Müller, and R. Schafer, "Soccer Jersey number recognition using convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 17–24.

[32] G. Li, S. Xu, X. Liu, L. Li, and C. Wang, "Jersey number recognition with semi-supervised spatial transformer network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1864–18647.

[33] A. Lehuger, S. Duffner, and C. Garcia, "A robust method for automatic player detection in sport videos," *Compression et Représentation des Signaux Audiovisuels*, vol. 4, pp. 1–5, Nov. 2007.

[34] D. Acuna, "Towards real-time detection and tracking of basketball players using deep neural networks," in *Proc. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4–9.

[35] K. Lu, J. Chen, J. Little, and H. He, "Light cascaded convolutional neural networks for accurate player detection," in *Proc. Brit. Mach. Vis. Conf.*, 2017, p. 173.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[37] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[38] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[39] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. ICCV*, 2017, pp. 2961–2969.

[40] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," 2013, *arXiv:1312.6082*.

[41] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.

[42] A. Deliege *et al.*, "SoccerNet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 4508–4519.

[43] A. Nady and E. Hemayed, "Player identification in different sports," in *Proc. 16th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2021, pp. 653–660.

[44] Q. Liang, W. Wu, Y. Yang, R. Zhang, Y. Peng, and M. Xu, "Multi-player tracking for multi-view sports videos with improved K-shortest path algorithm," *Appl. Sci.*, vol. 10, no. 3, p. 864, Jan. 2020.

[45] M. Liao, G. Pang, J. Huang, T. Hassner, and X. Bai, "Mask TextSpotter v3: Segmentation proposal network for robust scene text spotting," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 706–722.

[46] Z. Raisi, M. A. Naiel, G. Younes, S. Wardell, and J. S. Zelek, "Transformer-based text detection in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 3162–3171.

[47] M. Huang *et al.*, "SwinTextSpotter: Scene text spotting via better synergy between text detection and text recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, Jun. 2022, pp. 4593–4603.

[48] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy, "Learning to track and identify players from broadcast sports videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1704–1716, Jul. 2013.

[49] S. Gerke, S. Singh, A. Linnemann, and P. Ndjiki-Nya, "Unsupervised color classifier training for soccer player detection," in *Proc. Vis. Commun. Image Process. (VCIP)*, Nov. 2013, pp. 1–5.

[50] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, vol. 1, no. 1, pp. 886–893.

[51] L. Zhang, Y. Lu, G. Song, and H. Zheng, "RC-CNN: Reverse connected convolutional neural network for accurate player detection," in *Proc. Pacific Rim Int. Conf. Artif. Intell.* Cham, Switzerland: Springer, 2018, pp. 438–446.

[52] T. Guo, K. Tao, Q. Hu, and Y. Shen, "Detection of ice hockey players and teams via a two-phase cascaded CNN model," *IEEE Access*, vol. 8, pp. 195062–195073, 2020.

[53] M. Pobar and M. Ivasic-Kos, "Active player detection in handball scenes based on activity measures," *Sensors*, vol. 20, no. 5, p. 1475, Mar. 2020.

[54] M. Şah and C. Direkoğlu, "Review and evaluation of player detection methods in field sports," *Multimedia Tools Appl.*, pp. 1–25, Jun. 2021. [Online]. Available: https://link.springer.com/article/10.1007/s11042-021-11071-z#citeas, doi: 10.1007/s11042-021-11071-z.

[55] M. Manafifard, H. Ebadi, and H. A. Moghaddam, "A survey on player tracking in soccer videos," *Comput. Vis. Image Understand.*, vol. 159, pp. 19–46, Jun. 2017.

[56] R. Zhang, L. Wu, Y. Yang, W. Wu, Y. Chen, and M. Xu, "Multi-camera multi-player tracking with deep player identification in sports video," *Pattern Recognit.*, vol. 102, Jun. 2020, Art. no. 107260.

[57] R. Theagarajan and B. Bhanu, "An automated system for generating tactical performance statistics for individual soccer players from videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 2, pp. 632–646, Feb. 2021.

[58] K. Vats, P. Walters, M. Fani, D. A. Clausi, and J. Zelek, "Player tracking and identification in ice hockey," 2021, *arXiv:2110.03090*.

[59] S. Baysal and P. Duygulu, "Sentioscope: A soccer player tracking system using model field particles," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 7, pp. 1350–1362, Jul. 2016.

[60] A. Senocak, T.-H. Oh, J. Kim, and I. S. Kweon, "Part-based player identification using deep convolutional representation and multi-scale pooling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1813–18137.

[61] T. Feng, K. Ji, A. Bian, C. Liu, and J. Zhang, "Identifying players in broadcast videos using graph convolutional network," *Pattern Recognit.*, vol. 124, Apr. 2022, Art. no. 108503.

[62] A. Chan, M. D. Levine, and M. Javan, "Player identification in hockey broadcast videos," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113891.

[63] M. Jaderberg *et al.*, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2015, pp. 2017–2025.

[64] D. Delannay, N. Danhier, and C. De Vleeschouwer, "Detection and recognition of sports(wo)men from multiple views," in *Proc. 3rd ACM/IEEE Int. Conf. Distrib. Smart Cameras (ICDSC)*, Aug. 2009, pp. 1–7.

[65] S. Gerke, A. Linnemann, and K. Müller, "Soccer player recognition using spatial constellation features and Jersey number recognition," *Comput. Vis. Image Understand.*, vol. 159, pp. 105–115, Jun. 2017.

[66] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[67] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[68] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6399–6408.

[69] Wikipedia contributors. Number (sports). Accessed: Nov. 11, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Number_(sports)

[70] Z. Wang and J.-C. Liu, "Translating math formula images to LaTeX sequences using deep neural networks with sequence-level training," *Int. J. Document Anal. Recognit.*, vol. 24, no. 1, pp. 63–75, Nov. 2020.

[71] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 734–750.

[72] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.

[73] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[74] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1301–1310.

[75] N. Dvornik, J. Mairal, and C. Schmid, "Modeling visual context is key to augmenting object detection datasets," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 364–380.

[76] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6023–6032.

[77] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[78] G. Ghiasi *et al.*, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2918–2928.

[79] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7291–7299.

[80] A. Dutta, A. Gupta, and A. Zissermann. (2016). *VGG Image Annotator (VIA)*. [Online]. Available: http://www.robots.ox.ac.U.K./~vgg/software/via/

[81] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. (2019). *Detectron2*. [Online]. Available: https://github.com/facebookresearch/detectron2

[82] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Dec. 2019, pp. 8026–8037.

[83] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High quality object detection and instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1483–1498, May 2021.

[84] Y. Li, Y. Chen, N. Wang, and Z.-X. Zhang, "Scale-aware trident networks for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6054–6063.

[85] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[86] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[87] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017.

[88] J. Baek *et al.*, "What is wrong with scene text recognition model comparisons? Dataset and model analysis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4715–4723.

[89] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 369–376.

[90] J. Qui nonero-Candela, M. Sugiyama, N. D. Lawrence, and A. Schwaighofer, *Dataset Shift in Machine Learning*. Cambridge, MA, USA: MIT Press, 2009.

[91] F. Zhuang *et al.*, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jul. 2020.

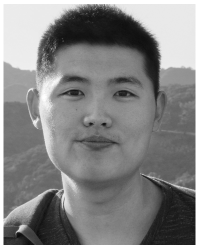
[92] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, Oct. 2018.

[93] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[94] Y. Wu and K. He, "Group normalization," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[95] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.

[96] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of Learning and Motivation*, vol. 24. New York, NY, USA: Academic, 1989, pp. 109–165.



Hengyue Liu received the B.S. degree from the Beijing University of Posts and Telecommunications, Beijing, China, and the M.S. degree in electrical engineering from the University of Southern California. He is currently pursuing the Ph.D. degree in computer vision with the Visualization and Intelligent Systems Laboratory (VISLab), University of California, Riverside, CA, USA. His research interests include object detection, scene graph generation, and mobile vision.

Bir Bhanu (Life Fellow, IEEE) received the B.S. degree (Hons.) from IIT-BHU, the M.E. degree (Hons.) from BITS, Pilani, the S.M. and E.E. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, and the M.B.A. degree from the University of California, Irvine, CA, USA. He is the Bourns Endowed University of California Presidential Chair in engineering, the Distinguished Professor of electrical and computer engineering, and the Founding Director of the Interdisciplinary Center for Research in Intelligent Systems from 1998 to 2019 and the Visualization and Intelligent Systems Laboratory, University of California, Riverside (UCR), CA, USA, since 1991. He is the Founding Professor of electrical engineering with UCR and served as its first Chair from 1991 to 1994. He has been the cooperative Professor of computer science and engineering since 1991, bioengineering since 2006, and mechanical engineering since 2008. Recently, he has served as the Interim Chair of the Department of Bioengineering from 2014 to 2016. He also served as the Director of the National Science Foundation Graduate Research and Training Program in video bioinformatics with UCR. Prior to joining UCR in 1991, he was a Senior Honeywell Fellow with Honeywell Inc. He has published extensively and has 18 patents. His research interests include computer vision, pattern recognition and data mining, machine learning, artificial intelligence, image processing, image and video database, graphics and visualization, robotics, human–computer interactions, and biological, medical, military, and intelligence applications. He is a fellow of AAAS, IAPR, SPIE, NAI, and AIMBE.