Jersey Number Recognition Using Joint Detection and Recognition

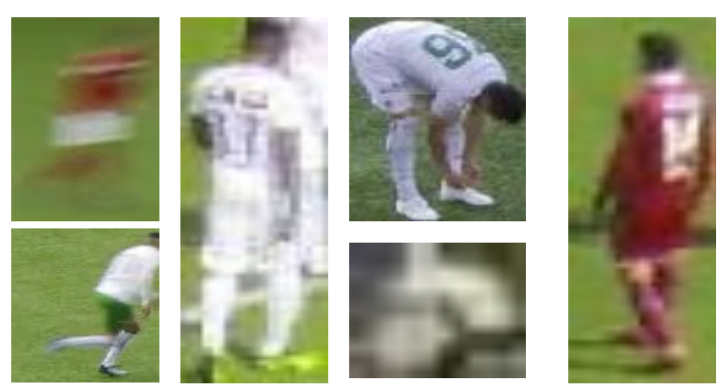
Tzu-Jo Hsu, Chia-Pei Chu, Oliver Wang, Xinyu Mei, Era Parihar
University of Michigan EECS 545 Machine Learning

Introduction/Motivation

Jersey number recognition is crucial for enhancing sports analytics, aiding in accurate player tracking and identification. Challenges such as player occlusion, multiple players, and blurry images complicate this task.

Our approach utilizes a two-step method:
- **Using a detector to find jersey numbers**
- **A recognizer to accurately identify these numbers**

Dataset



The training and testing dataset consists of 1427 and 1211 short videos of soccer players annotated with their corresponding numbers.

All Jersey numbers range from 0-99 and players with non visible numbers are classified as -1.

Results

- Fine-tuning DBNet++ with SVTR-small raised accuracy to 73.4%, and **weighted rules further increased it to 76.9%**.
- Fine-tuning with SVTR-base reached 71.5% accuracy, improving to 75.6% with weighted rules. FCENet paired with SVTR-small yielded 71.9% accuracy, rising to **79.8% with weighted rules**.
- Ongoing tests with FCENet suggest fine-tuning this detector may improve performance on our task.

These outcomes affirm that fine-tuning and weighted rules improve the performance of our model.

Detector	Recognizer	Accuracy (raw)	Accuracy (with weighted rules)
DBNet++	SAR	0/300, not run full test set	-
fine-tuned DBNet++	SVTR-small	889/1211 (0.734)	932/1211 (0.769)
fine-tuned DBNet++	SVTR-base	866/1211 (0.715)	916/1211 (0.756)
fine-tuned FCENet	SVTR-small	871/1211 (0.719)	966/1211 (0.798) rule_v1
fine-tuned FCENet	SVTR-small	871/1211 (0.719)	**1005/1211 (0.830) rule_v2**

Methodology

Baseline and Annotation

We used out-of-the-box pretrained detectors and recognizers as our baseline, but none of the examples were correctly identified. Noticing that many out-of-box recognizers worked decently with handmade crop samples, we **hand-annotated 1288 samples** with bounding boxes and their labels to improve the detector.

Detector	Precision- 10 epochs
fine-tuned DBNet++	0.9091
fine-tuned FCENet	0.9605

Fine-tuning and Experiments

We applied **image augmentation** techniques to enhance the performance of fine-tuning:
- ColorJitter: randomly change the brightness, contrast and saturation
- Random crop/resize/padding/rotation to the annotated data.

After the fine-tuning, our methods are able to achieve around **70~75% accuracy on the full test dataset**.

We noticed that the fine-tuning on recognizers did not improve the overall performance. Thus we then analyzed mis-classified samples and identified several edge cases.

Research and Solution

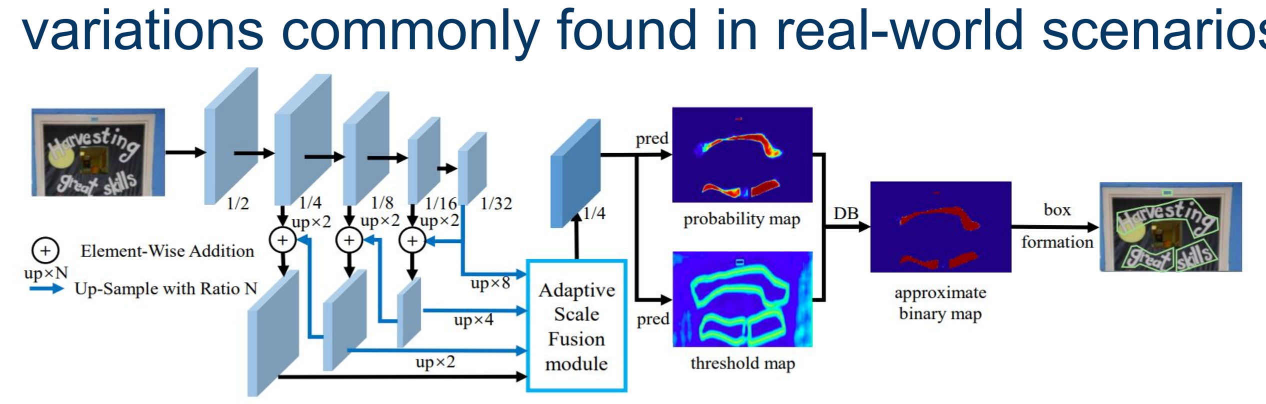
To address the high variety in detected numbers, we incorporated **rule-based methods**:
- Increasing weighting for continuous digits & double digits
- Applying detection thresholds to enhance the consistency of the final prediction.

With these rules and fine-tuning, accuracy reached ~83%.

We also **bootstrapped a larger fine-tuning dataset of 43824 samples** using our fine-tuned detector and recognizer. Samples were included in a larger pool only if the model outputs matched the ground truth.

Detectors and Recognizers

DBNet++: a real-time scene text detection method with Differentiable Binarization and **A**daptive **S**cale **F**usion.
- It features an ASF module that adaptively fuses multi-scale features, ensuring robustness to the scale variations commonly found in real-world scenarios.



FCENet: a novel approach for detecting text with arbitrary shapes using Fourier Contour Embedding.
- FCEnet predicts text contours in the Fourier domain and reconstructs them in the spatial domain using Inverse Fourier Transformation for accurate shape approximation.

SVTR: a three-stage network with progressively decreasing height.
- In each stage, a series of mixing blocks are carried out and followed by a merging or combining operation.
- At last, the recognition is conducted by a linear prediction.
- The latest recognizer provided by mmOCR; SVTR-small gave better results than other recognizers in experiments

Failure Cases

Cases	Examples	Possible Solution
Similar numbers	5 detected as 6; 6 as 8; 9 as 3, etc.	Improve the recognizer through fine-tuning or mixture of votes
Detections on incorrect players	Opposing player steps into frame and their number is detected	Color based filtering (K-means approach below)
Partial occlusion/blur	Player with number 11 turned to the side may be detected as 1	Improve the detector quality and use a higher detection threshold
Out-of-distribution orientations	6 appears as 9 with bent-over player	Finetune with rotation augmentations or filter out these detections

Applying K-Means to Identify the Main Player