Temporal Smoothing in 2D Human Pose Estimation for Bouldering

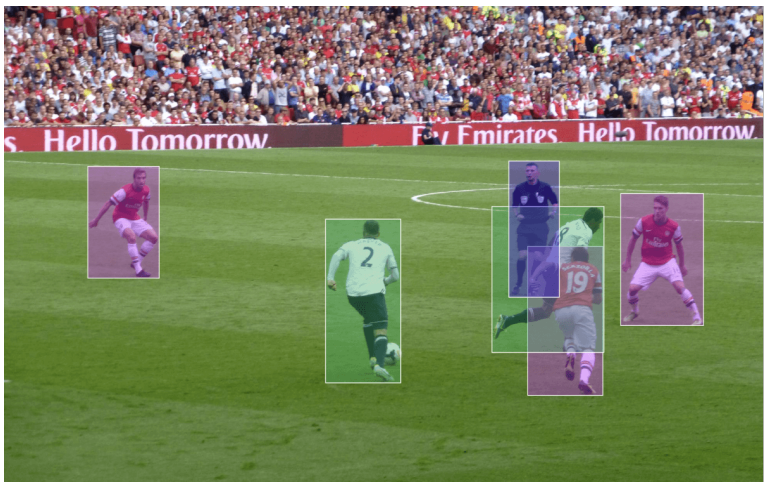
André Oskar Andersen
wpr684

Institution of Computer Science, University of Copenhagen

2023

Introduction

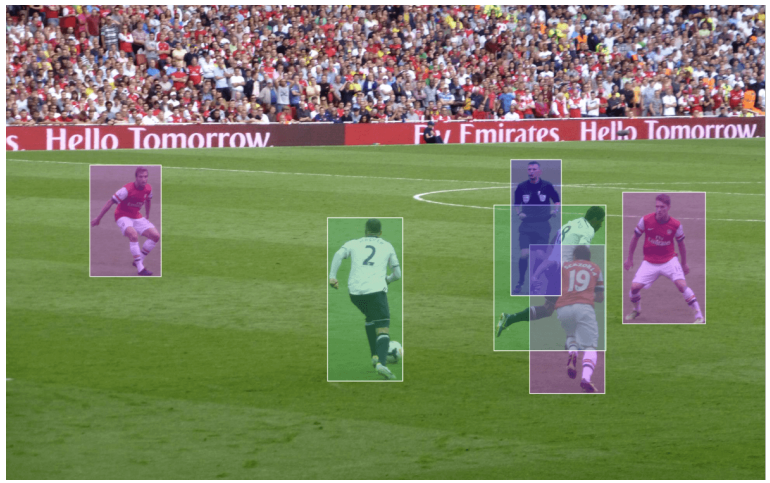
- Increased usage of video analysis in sports.



https://www.superannotate.com/blog/computer-vision-in-sports

Introduction

- Increased usage of video analysis in sports.
- Often requires the position of the players.
  - Already developed for popular sports.
  - Missing for the less popular sports.



https://www.superannotate.com/blog/computer-vision-in-sports

Introduction

- Increased usage of video analysis in sports.
- Often requires the position of the players.
- Problems with the data
  - Methods require large quantities
  - Unusual poses/movements

https://climbalong.com/lab

Introduction

- Increased usage of video analysis in sports.
- Often requires the position of the players.
- Problems with the data
- ClimbAlong at NorthTech ApS
  - Frame-idependent pose-detector for bouldering

https://climbalong.com/lab

Introduction

- Increased usage of video analysis in sports.
- Often requires the position of the players.
- Problems with the data
- ClimbAlong at NorthTech ApS
  - Frame-idependent pose-detector for bouldering
  - Proposition: Incorporate temporal information

https://climbalong.com/lab

Introduction

- Aim: extending the ClimbAlong pose-detector to use temporal information.

The Data

ClimbAlong

- Fully annoated videos of climbers

https://climbalong.com/lab

The Data

ClimbAlong

- Fully annoated videos of climbers
- Problem: Very small dataset

https://climbalong.com/lab

The Data

ClimbAlong

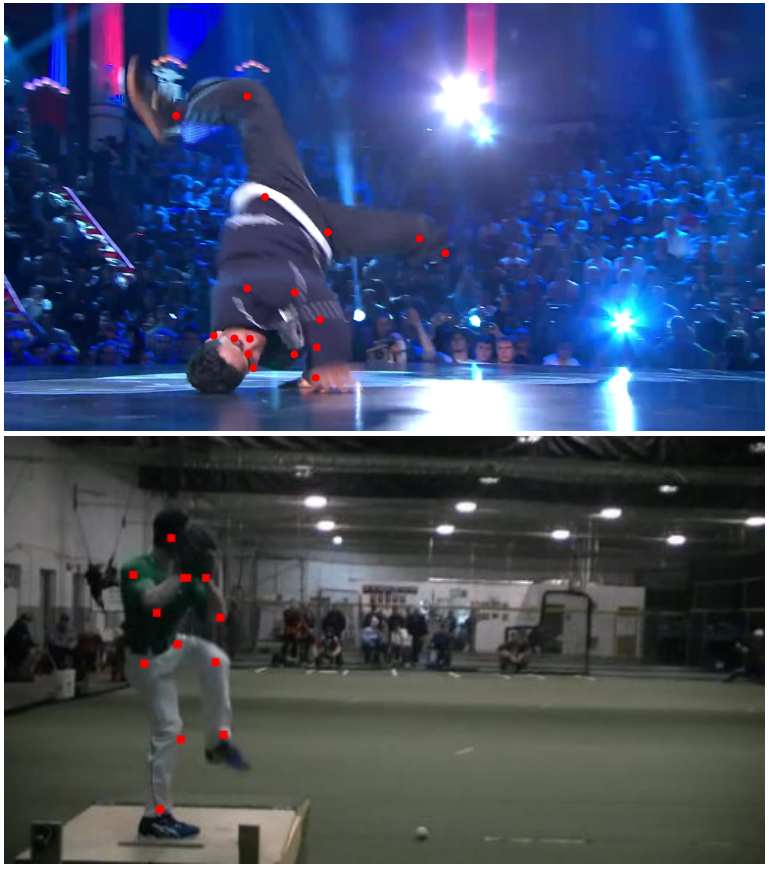
- Fully annoated videos of climbers
- Problem: Very small dataset
- Solution: pretrain on related datasets and finetune on ClimbAlong
  - BRACE
  - Penn Action

https://climbalong.com/lab

The Data

ClimbAlong

- Fully annoated videos of climbers
- Problem: Very small dataset
- Solution: pretrain on related datasets and finetune on ClimbAlong
  - BRACE
  - Penn Action



The Models

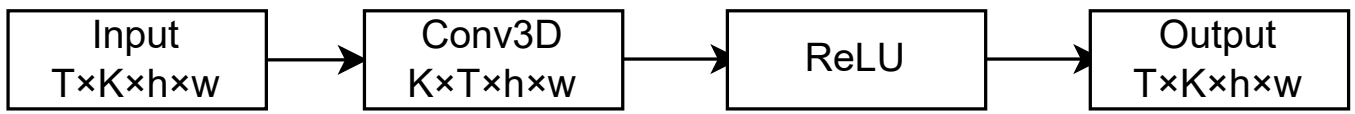
Generally, three approaches

1. Convolutional layer
2. Recurrent neural network
3. Transformer

The Models

3DConv

- 3-dimensional conv. layer + ReLU



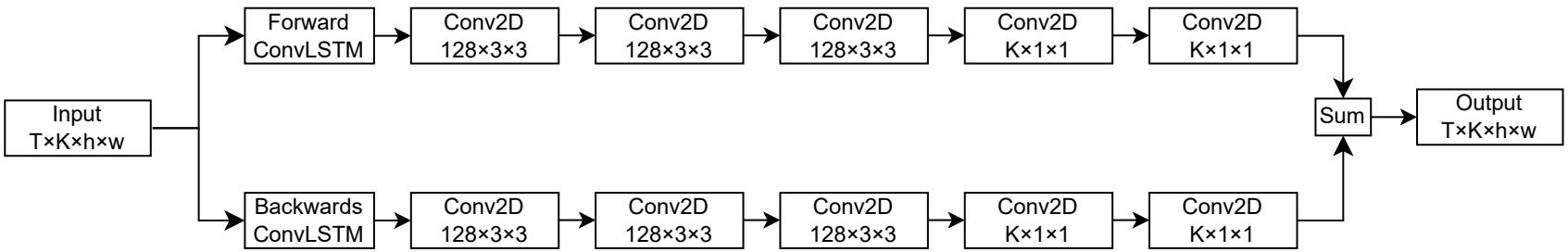
---

https://arxiv.org/pdf/2203.08713.pdf

The Models

bi-ConvLSTM Model S

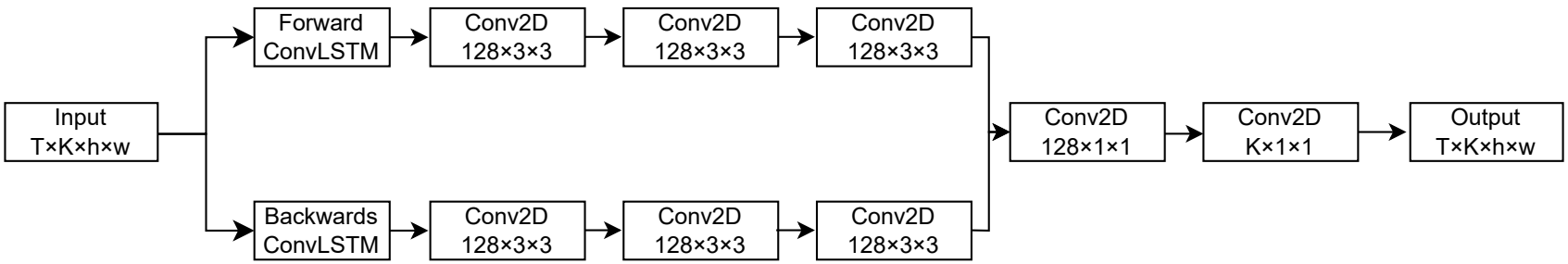
- Adaptation of Unipose-LSTM by Artacho and Savakis
- Bidirectional convolutional LSTM + conv. layers and ReLU
- Processing directions summed together



The Models

bi-ConvLSTM Model C

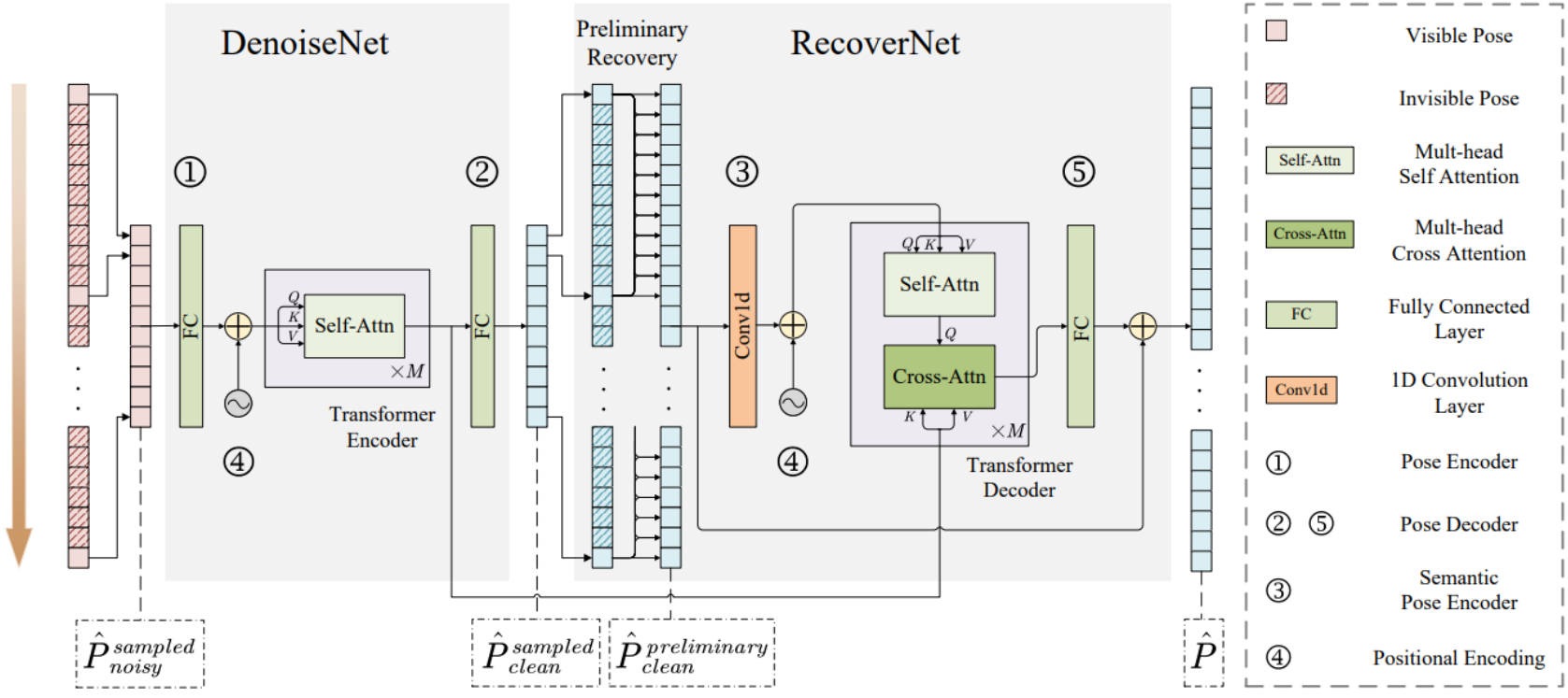
- Problem: Prioritization of processing direction
- Solution: Using convolution



The Models

DeciWatch by Zeng *Et al*.

- Transformer-based
- Only considers every *n*th frame
- Encoder: DenoiseNet
- Decoder: RecoverNet



Pretraining

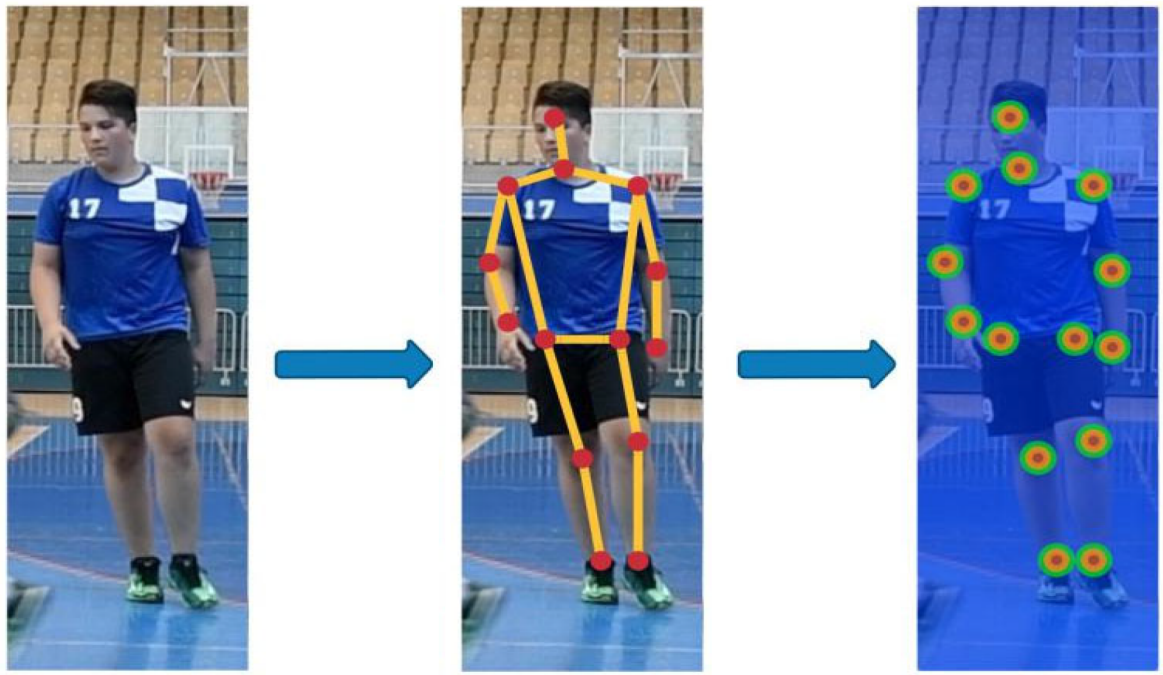
Procedure

- Not training pose-detector
    1. Different input images

Pretraining

Procedure

- Not training pose-detector
  1. Different input images
- Simulating pose-detector output by adding noise
  1. Gaussian filter standard deviation
  2. Shifting keypoints



https://www.mdpi.com/2313-433X/8/11/308

Pretraining

Finding optimal setting of models

- Three experiments
  1. Sample Gaussian filter standard deviation from $\{1, 1.5, 2, 2.5, 3\}$.

Pretraining

Finding optimal setting of models

- Three experiments
  1. Sample Gaussian filter standard deviation from $\{1, 1.5, 2, 2.5, 3\}$.
  2. Fixed standard deviation

Pretraining

Finding optimal setting of models

- Three experiments
  1. Sample Gaussian filter standard deviation from $\{1, 1.5, 2, 2.5, 3\}$.
  2. Fixed standard deviation
  3. Decreased frame rate

Pretraining

Finding optimal setting of models

- Three experiments
  1. Sample Gaussian filter standard deviation from $\{1, 1.5, 2, 2.5, 3\}$.
  2. Fixed standard deviation
  3. Decreased frame rate
- Shifting by sampling from $\mathcal{N}(\mu = 0, \sigma = 3k)$ for $k \in \{1, 2\}$

Finetuning

- Using all of the developed models with pose-detector
- Freezing pose-detector
  1. Quicker fitting
  2. Greater understanding of results

Finetuning

- Test results

Discussion

Results:

- Translation vs translation + scaling

Discussion

Results:

- Translation vs translation + scaling
- Halfing frame rate

Discussion

Results:

- Translation vs translation + scaling
- Halfing frame rate
- Easiest vs most difficult joints

Discussion

Results:

- Translation vs translation + scaling
- Halfing frame rate
- Easiest vs most difficult joints
- Experiment 1 vs experiment 2

Discussion

Results:

- Translation vs translation + scaling
- Halfing frame rate
- Easiest vs most difficult joints
- Experiment 1 vs experiment 2
- Effects of pretraining

Discussion

Results:

- Translation vs translation + scaling
- Halfing frame rate
- Easiest vs most difficult joints
- Experiment 1 vs experiment 2
- Effects of pretraining
- Worst performing keypoints

Discussion

All models performed better during finetuning than pretraining

1. More data

Discussion

All models performed better during finetuning than pretraining

1. More data
2. Semantically different videos in pretraining

Discussion

All models performed better during finetuning than pretraining

1. More data
2. Semantically different videos in pretraining
3. Noise in BRACE annotations

Discussion

All models performed better during finetuning than pretraining

1. More data
2. Semantically different videos in pretraining
3. Noise in BRACE annotations
4. Frame rate in Penn Action

Discussion

All models performed better during finetuning than pretraining

1. More data
2. Semantically different videos in pretraining
3. Noise in BRACE annotations
4. Frame rate in Penn Action
5. Performace of identity function

Discussion

Which model is the best?

- Greatest testing accuracy: DeciWatch 1.1/1.2

Discussion

Which model is the best?

- Greatest testing accuracy: DeciWatch 1.1/1.2
- Greatest rough estimation: bi-ConvLSTM Model C 1.1

Discussion

Which model is the best?

- Greatest testing accuracy: DeciWatch 1.1/1.2
- Greatest rough estimation: bi-ConvLSTM Model C 1.1
- Speed and memory: 3DConv

Discussion

General reflections

- Pretraining
  - Should have estimated parameters of data

Discussion

General reflections

- Pretraining
  - Should have estimated parameters of data
  - Overlapping video sequences

Discussion

General reflections

- Pretraining
  - Should have estimated parameters of data
  - Overlapping video sequences
- Finetuning
  - Groundtruth outside of bbox

Discussion

Future work

1. DeciWatch with all frames

Discussion

Future work

1. DeciWatch with all frames
2. DeciWatch with vision transformer

Discussion

Future work

1. DeciWatch with all frames
2. DeciWatch with vision transformer
3. Avoid overfitting

Discussion

Future work

1. DeciWatch with all frames
2. DeciWatch with vision transformer
3. Avoid overfitting
4. Multiple retraining

Conclusion

Succesfully developed and tested the incorporation of temporal smoothing for pose estimation

Extras: Mistakes Were Made!

Misimplemented evaluation-function