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.

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.

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.
- Problems with the data
  - Methods require large quantities
  - Unusual poses/movements

Introduction

- ClimbAlong at NorthTech ApS
  - Frame-idependent pose-detector for bouldering

Introduction

- ClimbAlong at NorthTech ApS
  - Frame-idependent pose-detector for bouldering
  - Proposition: Incorporate temporal information

Introduction

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

The Data

- 30 fully annotated videos of climbers with 25 keypoints, totalling to 10,293 frames, provided by ClimbAlong

The Data

- 30 fully annotated videos of climbers with 25 keypoints, totalling to 10,293 frames, provided by ClimbAlong
- Both static and quick movements.

The Data

- Very small dataset

The Data

- Very small dataset
- Instead, pretraing on related datasets and finetune of ClimbAlong

The Data

- The BRACE dataset:
  - Breakdancers

The Data

- The BRACE dataset:
  - Breakdancers
  - 1,352 video sequences / 334,538 frames with 17 keypoints.

The Data

- The BRACE dataset:
  - Breakdancers
  - 1,352 video sequences / 334,538 frames with 17 keypoints.
  - Compared to ClimbAlong:
    - Swaps between static and quick movements
    - Less frequent static movements
    - Quicker movements

The Data

- Penn Action dataset:
  - People performing various actions

The Data

- Penn Action dataset:
  - People performing various actions
  - 2,326 video sequences with 13 keypoints and binary visibility-flag.

The Data

- Penn Action dataset:
  - People performing various actions
  - 2,326 video sequences with 13 keypoints and binary visibility-flag.
  - Filtered down to 307 video sequences / 26,036 frames.

The Models

- Motivation for valg
- 3D Conv
- DeciWatch
- bi-ConvLSTM Model S
- bi-ConvLSTM Model C

The Models

- Generally, three approaches
  1. 3-dimensional convolutional layer
  2. Convolutional recurrent neural network
  3. Transformer

The Models

- Generally, three approaches
  1. 3-dimensional convolutional layer
  2. Convolutional recurrent neural network
  3. Transformer
- One architecture based on each approach

The Models

- 3DConv
  - 3-dimensional conv. layer + ReLU
  - Input/output: heatmaps

The Models

- 3DConv
  - 3-dimensional conv. layer + ReLU
  - Input/output: heatmaps
  - $K \in \mathbb{N}$ filters with $h, w \in \mathbb{N}$ height and width
  - $T \in \mathbb{N}$ frames

The Models

- bi-ConvLSTM Model S
  - Adoptation of Unipose by Artacho and Savakis
  - Bidirectional convolutional LSTM + conv. layers and ReLU
  - Processing directions summed together

The Models

- bi-ConvLSTM Model S
  - Adoptation of Unipose by Artacho and Savakis
  - Bidirectional convolutional LSTM + conv. layers and ReLU
  - Processing directions summed together
  - Input/output: heatmaps
  - $K \in \mathbb{N}$ filters with $h, w \in \mathbb{N}$ height and width
  - $T \in \mathbb{N}$ frames

The Models

- bi-ConvLSTM Model C
  - Similary to bi-ConvLSTM Model S
  - Problem: Prioritization of processing direction

The Models

- bi-ConvLSTM Model C
  - Similary to bi-ConvLSTM Model S
  - Problem: Prioritization of processing direction
  - Solution: Using convolution

The Models

- DeciWatch by Zeng *Et al*
  - Transformer-based
  - Only considers every *n*th frame
  - Processes keypoints

The Models

- DeciWatch by Zeng *Et al*
  - Transformer-based
  - Only considers every *n*th frame
  - Processes keypoints
  - Encoder: DenoiseNet
  - Decoder: RecoverNet

Pretraining

- Not training pose-detector

Pretraining

- Not training pose-detector
- Adding noise to input-data

Pretraining

Data preprocessing

- Discarding images

Pretraining

Data preprocessing

- Discarding images
- Simulating output of pose-detector on ClimbAlong dataset:
  1. Making input data consist of bboxes with side-length of 56 px

Pretraining

Data preprocessing

- Discarding images
- Simulating output of pose-detector on ClimbAlong dataset:
  1. Making input data consist of bboxes with side-length of 56 px
  2. 25 Heatmaps

Pretraining

Data preprocessing

- Discarding images
- Simulating output of pose-detector on ClimbAlong dataset:
  1. Making input data consist of bboxes with side-length of 56 px
  2. 25 Heatmaps
  3. Adding noise to input
     - 3.1 Shifting-scalar
     - 3.2 Gaussian filter standard deviation

Pretraining

- Data configuration
  - $s = 5$ frames
  - Splitting of data into non-overlapping and repeating subsets
  - Handling of missing keypoints

Pretraining

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

Pretraining

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

Pretraining

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

Pretraining

- Experiments
  - Sample Gaussian filter standard deviation from $\{1, 1.5, 2, 2.5, 3\}$.
  - Fixed standard deviation
  - Decreased frame rate
- Two different shifting-scalars $s \in \{1, 2\}$

Pretraining

- Experiments
  - Sample Gaussian filter standard deviation from $\{1, 1.5, 2, 2.5, 3\}$.
  - Fixed standard deviation
  - Decreased frame rate
- Two different shifting-scalars $s \in \{1, 2\}$
- DeciWatch - inspected source code:
  - Sampling every 5th frame

Pretraining

- Results

Finetuning

- Data-preprocessing
- Training Details
- Results
- Visualizations?
- Subconclusion

Finetuning

- Freezing pose-detector
  - Quicker fitting

Finetuning

- Freezing pose-detector
  - Quicker fitting
  - Greater understanding of results

Finetuning

- Data preprocessing
  - Making data more similar to pretraining
    - Setting negative values to 0
    - Normalizing summing up to same value as pretraining heatmaps

Finetuning

- Data preprocessing
  - Making data more similar to pretraining
    - Setting negative values to 0
    - Normalizing summing up to same value as pretraining heatmaps
  - Using predicted bbox to create heatmaps

Finetuning

- Data preprocessing
  - Making data more similar to pretraining
    - Setting negative values to 0
    - Normalizing summing up to same value as pretraining heatmaps
  - Using predicted bbox to create heatmaps
    - Handling of groundtruth keypoints out of bbox

Finetuning

- Data configuration
  - Careful splitting of dataset

Finetuning

- Results

Finetuning

- Additional experiments

Finetuning

- Additional test results

Discussion

Conclusion

Extras: Mistakes Were Made!