1 Discussion

The following section discusses our obtained results and our configuration choices, as well as suggests some possible future work.

1.1 Results

1.2 Data Configurations

1.3 Model Configurations

1.4 Future Work

============== OLD ==============

2 Discussion

The following section discusses our approach, including the our choice of configuration of the models, the data and the training parameters, as well as suggests some future work.

2.1 Data Configuration

2.1.1 Datasets

- Vanishing gradient for unipose - den fungerer bedre på range [0, 255] end på range [0, 1]
- Handling af data uden for pred bbox ved CA-data
- Valg af shifting-scalar:
  - Jeg kunne have approximeret den fra mask rcnn
  - Jeg kunne have gjort den afhængig af det enkelte led
- Pretrain: jeg shifter gt med op til 6 std, men jeg udglatter højst med 3, så jeg dækker egentlig ikke den rigtige placering af gt
-

For the data we used three datasets. For the pretraining stage we used the BRACE datase and the Penn Action dataset, whereas for the finetuning stage we used the ClimbAlong dataset. For our choice of pretraining datasets we see a couple of suboptimal choices.

BRACE and Penn Action can be very semantic different from each other. For instance, in the BRACE dataset the person in the video sequence is very often upside down and performs some very quick movements, whereas the people in the Penn Action dataset is usually in some more usual poses an do not make movements that are as quick as the ones from BRACE. We tried to fix this by only including video sequences from the Penn Action dataset that had more poses where the person was not just standing straight up, however, even by doing so, the two datasets are still sematically different. Further, the video sequences from the two datasets are often filmed from two very different camera angles, resulting in the same keypoints being placed at different locations. If the two datasets are very different this will have an effect on the fitting of the models.

We also see some problems in the pretraining datasets individually. For the BRACE dataset, it is clearly documented, that the video sequences is filmed using 30 frames per second. For

Penn Action, on the other hand, a similar information is not stated anywhere and the video sequences comes as individual frames without the duration of the video sequences noted, hence why cannot either compute the frame rate of these video sequences. This could cause some problems, as a two windows of frames, one from each dataset, could span two different durations and thus capture two different duration of context. For the BRACE we also see an issue, as the dataset has not been fully manually annotated, but instead annotated using a pose estimator and only certain incorrectly predicted poses were manually annotated, where incorrectly predicted poses were detected by a machine learning model. However, if this machine learning model for detecting incorrectly predicted poses delivers suboptimal results, then the annotations would also be suboptimal and thus contain some noise, which is very difficult for our models to predict.

2.1.2 Data Preprocessing

For the preprocessing of the datasets, we split each video sequence into multiple windows of five frames. One could have argued, that we this was incorrect of us to do and that we instead should have used the whole video at once. However, we see a couple of problems with this. First off, this would require a lot of memory, as we have to store all of these frames at once. Secondly, for models like the 3-dimensional convolutional layer which has a parameter based on the length of the input sequence, this would be a problem, as the video sequences have different lengths and the only solution would thus be to either pad the shorter video sequences or trim the longer video sequences, such that all video sequences have the same length. Generally, we believe that our choice of cutting the video sequences into multiple windows was correct. However, one might argue, that these windows should consist of more than five frames, as this is way too little information for the models to learn the context of each frame, such that they accurately can perform temporal smoothing. However, we also disagree. We have clearly seen in section **??** and **??**, how our models massively outperforms identity function, indicating that five frames is enough context for the models. Further, as stated by Artacho *Et al.*, the performance of the models plateaues after using more than five frames [1]. They did however use both different models and datasets than we did, so we could potentially have a different experience than what they did if we used more than five frames.

For the preprocessing of the pretraining dataset, the goal was to preprocess the data, such that it simulated the output of the already developed pose-estimator. This was done through multiple steps, where we used multiple values. For instance, we expanded the bounding-boxed created from the keypoints by 10% in each direction, we used a standard deviation in the range $\{1, 1.5, 2, 2.5, 3\}$ for the Gaussian filter on the input data, and we shifted the input by sampling from a normal distribution with mean $\mu_{in} = 1$ and standard deviation 3 or 6, depending on the experiment. These values were all some values that we came up with. We could instead have picked these values in a more sophisticated manner such that the data would be more similar to the ClimbAlong data. This could for instance have been done by approximating them from the ClimbAlong dataset. We could further have made this standard deviation keypoint-dependent, such that it would have modelled whether how much movement each keypoint makes. However, we do not believe this to be much beneficial, as (1) we generally already receive some great results which are difficult to beat, and (2) for the experiments where we increased the added noise, we did not see an improvement on of the keypoints that carry a lot of movement, for instance the pinky, hence why moveement does not require a more shifted keypoint.

For the last part of the preprocessing of the pretraining dataset, we split the data into a training, validation and test set, consisting of 60%, 20% and 20% of the data respectively. This was done by making sure that none of the windows were repeated and that none of the windows were overlapping between the three datasets. By doing so we ensured that the evaluation did carried minimal bias, as the models had not seen any of the frames in the validation and test set during its training. However, different frames of the same video sequence could appear across the three sets, which could maybe carry some bias, as the same person would then appear in the three sets. However, we do not believe this to happen in our case, as the pretraining dataset does not include the actual frames but instead only include the annotations.

For the finetuning data we used the output of the already developed pose-estimator as the input to our models. Of course, by further fitting the already developed pose-estimator in connection with our temporal inclusive models we could have received some better results than what we did, however, we decided not to do so and instead to freeze the already developed pose-estimator, as this would decrease the training time of our models, as well as we can easily see the difference it makes to incorporate the temporal information in the pose estimation, as this was the major goal of the project.

For the finetuning dataset we also used a window consisting of $k = 5$ frames and a value of $c = 255$ in the heatmaps as a representation the ground truth placement of the keypoints. We simply did this, as it was the most similar to how we did it in the pretraining stage and thus the model would have to learn less than in the case that we used another setting.

2.2 Model Configuration

- Model parametrer

2.3 Training Configuration

- Skulle have brugt en længere patience til early-stopping
-
- Kør eksperimenter flere gange for at fjerne randomness i resultater.

2.4 Future Work

- Transformer uden skipping af frames
- Dataaugmentation til CA
- Træn uden PA
- Vision transformer