Keypoint-Based Feature Amplification for Multiple Object Tracking

Erick Platero (1812570), Keyon Amirpanahi (1958721), Sindhuja Thogarrati (1857970)

Department of Computer Science
University of Houston

Spring 2023

Abstract: In this work, we present a fast and efficient feature amplification algorithm for the task of Multiple Object Tracking (MOT) that segments an image to amplify the foreground signal and dim the background noise. We apply this process on raw bounding box detections and evaluate the difference in performance when applied to DeepSORT, a current detection-and-tracking model. Our findings indicate poorer performance of DeepSORT on multiple metrics when this feature amplification is applied to the detections, possibly due to the loss of object features as a result of restrictive masking or the loss of spatial signals present in the background.

1. Introduction

Human detection is a field that has rapidly progressed in the last decade due to the remarkable performance of deep learning algorithms. An organic step from this domain is the task of human tracking. Human tracking, unlike human detection, captures spatial-temporal and object appearance cues to infer the identity of the detection. A subcategory of the field is one-shot learning where the constraint is that with one or few detections of a unique identity, the model can generalize the identity to future images of the person. A general approach to achieve this task is to use a two-step process: (i) detect all objects of interest and (ii) give identities to each object detection. The former uses a detection model while the latter uses an embedding network and a motion algorithm to associate identities. The embedding network is responsible for creating a vector that encodes the appearance cues of the detection. As far as we know, all two-step models directly feed the raw re-sized bounding box image to the embedding network [1,2]. The drawbacks of this approach is that cropped bounding box images have background noise which forces the embedding network to also learn to discriminate between noise and signal appearance cues on a low-dimensional representation of the image. To aid the network to be able to separate between noise and signal cues, feeding the cropped image to a segmentation model that can black out the background noise and retain the foreground noise will aid the embedding network to create a better representation of the detected object. However, segmentation models are infamously large models that preclude any real-time application. In this work, we propose a faster and simpler approach to perform segmentation on cropped images that harnesses pose models instead of segmentation models.

2. Previous Work

The MOT field has received much attention due to its applications and challenges. While tracking can be performed on any object, most of the progress has focused on pedestrian tracking. The reasons for this are three-fold: (i) pedestrian tracking has organic applications in surveillance, (ii) humans are dynamic and non-rigid objects, and (iii) data for this environment is widely available. In this area, there are two primary tracking paradigms: 1) tracking-by-detection (also known as two-step tracking) and 2) joint detection and tracking. The former relies on association schemes to assign trajectories to the bounding boxes given by the pre-trained detection model. The latter unifies the detection and tracking into a single end-to-end model. Of the two, tracking-by-detection is the most popular likely due to its simplicity and intuitive formulation. In this project, we will be focusing on the former approach. An early and popular formulation was made in [3] where bounding boxes were given trajectories by leveraging spatial-temporal motion with the Kalman Filter. Then, [4] added to the association scheme by introducing an embedding network that learns targets based on their appearance cues. From this, others have introduced new schemes at every component. For detection, [1] leverages the confidence of detections to make better trajectories; for motion, [5] introduces a scheme that leverages just the bounding boxes to make a prediction, and for association, [6] introduces an association scheme that does not depend on appearance cues. Still others have explored different areas: [7] introduces association schemes that leverage stereo data, [8] harnesses the fact that two objects are unable to occupy the same space to make detections, and others have made specific algorithms to handle target occlusions [9–12]. Each of these methods attempts to implement more sophisticated algorithms or exploit different assumptions to improve MOT performance. However, the common similarity between the methods that harness appearance cues is that they all work with the raw bounding box images to make decisions. In this work, we evaluate the effects of pre-processing the cropped image to amplify foreground signal and diminish background noise.

3. Methodology

3.1. Feature Amplification

The novel component we introduce is a Feature Amplification (FA) of objects being tracked using some predicted characteristics. We particularly focus on the task of human tracking, using DeepSORT as our base tracking model. Instead of directly feeding the cropped images to the re-identification network, we can pre-process the image to suppress the background noise and amplify the human detection component as shown in fig. 1. The final output (shown in rightmost image) shows how the background noise is suppressed to black while the signal is maintained.

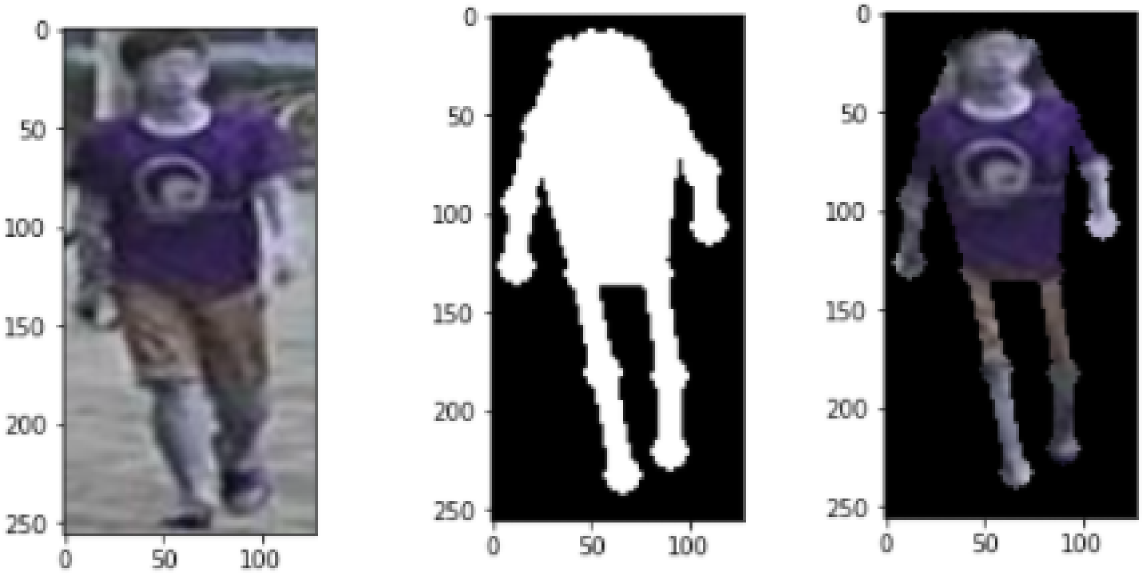


Fig. 1: Leftmost: image from market-1501; Middle: masked image; Rightmost: image after noise suppression.

The pose detection model mmpose is used to predict the keypoints of joints in the pose's skeleton. From there, our segmentation function generates a mask from those keypoints and applies it to the image, removing the background noise and amplifying the person being tracked. The rationale behind this is that if we can contrast the noise from the signal, then we will expect our embedding network to be able to create better compressed representations of the human-component. The way that we transform a body-pose prediction into masks is that we first apply a dilation operation to the skeleton of the joints and then we create a polygon that captures the torso and head of the person in the cropped image. This image pre-processing also serves as a form of semi-supervised learning for the re-identification network, as it creates a new decision boundary between the human and non-human components of the bounding boxes that the re-identification model is trained on. Furthermore, the pre-processing is also ultimately applied to the test data, so the incorporation of this pre-processing step is not merely transductive in nature. The implementation of our feature-amplifying segmentation step is described in algorithm 1 (code can be found in the `preprocess_image` function of `mmpose_preprocess.py`). Lastly, It is worthy to note that FA can be performed with pretrained segmentation models. However, image segmentation models are notoriously large models that preclude its application to any real-time setting. In this work, we find a middle-ground between MOT and MOT Segmentation models by performing FA using a pose model.

Algorithm 1 Feature Amplification of Tracked Object Using Predicted Keypoints

```
Input
    $[\boldsymbol{X}]_{w \times h}$    image tensor, cropped to bounding box of object
    $\boldsymbol{k}$    vector of predicted keypoints
    $\boldsymbol{s}$    vector of keypoint-keypoint pairs that defines object's skeleton
Output
    $\boldsymbol{X}_M$    feature-amplified image tensor in which background of object is reduced

$\boldsymbol{M} \leftarrow [0]_{w \times h}$    ▷ Initialize mask
for $(x, y) \in \boldsymbol{k}$ do
    $\boldsymbol{M}_{xy} \leftarrow 1$
end for
for $(k_1, k_2) \in \boldsymbol{s}$ do
    DRAWLINE($\boldsymbol{M}$,$k_1$,$k_2$)
end for
DILATE($\boldsymbol{M}$)    ▷ Dilate drawn skeleton in mask using kernel to cover more of object
DRAWCONTOURS($\boldsymbol{M}$)    ▷ Fill interior of skeleton in mask to capture torso and head
$\boldsymbol{X}_M \leftarrow \boldsymbol{X} \odot \boldsymbol{M}$    ▷ Hadamard product
return $\boldsymbol{X}_M$
```

3.2. Self-Training

As the pre-trained mmpose model that would be used for this segmentation was trained on the Common Objects in Context (COCO) dataset, which is not used for object tracking tasks, we felt it would be further necessary to self-train mmpose on the Multiple Object Tracking 17 (MOT17) dataset to better adapt to those tasks. Self-training, in particular, is necessary because the MOT17 dataset does not include ground truth for the keypoints of people being tracked. As a result, this data is considered unlabeled for the purposes of training mmpose in this new context. Through self-training, a set of unlabeled data can be given labels from the predictions of a pre-trained model, and this newly labeled data can be used to train the model. We have X_L as the labeled dataset that the pose-model is trained under and X_U which is the set of cropped images given by a detection network on the training dataset. We create subset $X'_U \subseteq X_U$ which contains all cropped images in the training dataset in which the pre-trained pose-model inferred with high confidence.

Our process for self-training a new pose-detection model is as follows. Using an mmpose model pre-trained on the COCO dataset, we generate keypoint predictions for the images of identities present in the MOT17 dataset. The confidence values of these predictions are averaged over each skeleton, and we observe the distribution of these averages as in fig. 2. In order to select a confidence threshold over which predicted keypoints would be used as ground truth labels for training a new model, we observed the segmented images at each quartile of this distribution as in fig. 3. In our case, the segmentation at the median appeared to show accurately identified keypoints of a human identity, so the median confidence value of 0.75 was chosen as our threshold. A new pose-detection model was then trained on the images from the MOT17 dataset on which the pre-trained model predicted keypoints with confidence greater than the threshold, using those keypoint predictions as the ground truth.

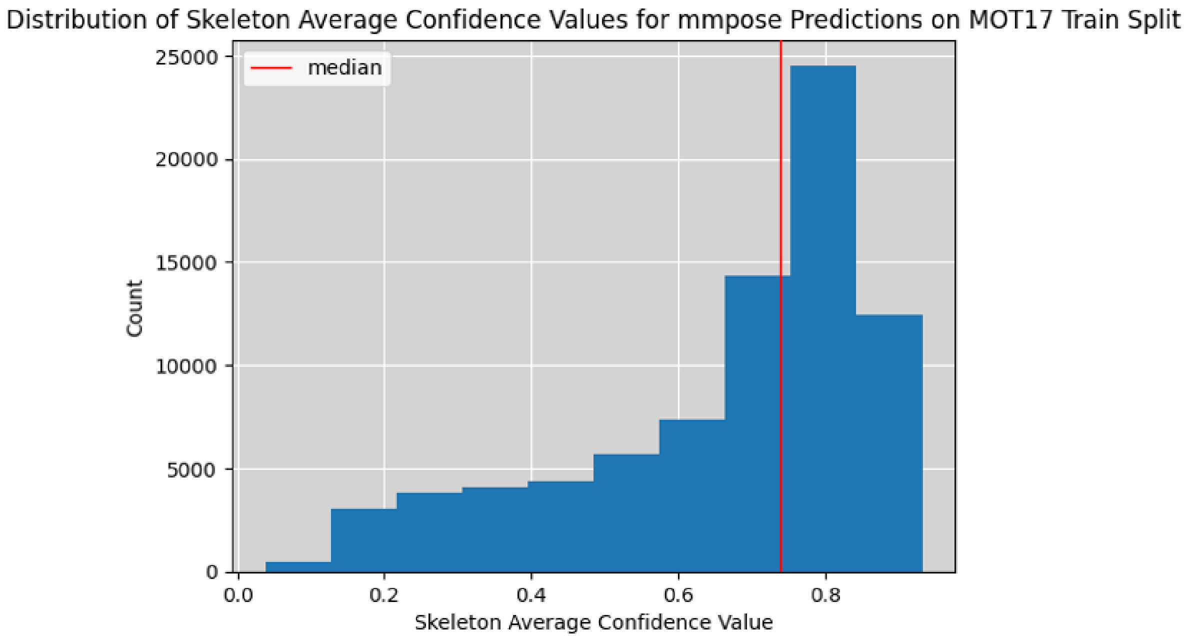


Fig. 2: Distribution of Average Skeleton Confidence Values From Pre-trained Pose-Detection Model

3.3. *FA Embedding Network*

The Embedding network of DeepSORT is responsible for transforming the cropped images into a vector embedding representation. From this representation, the pairwise distances of all tracked object is calculated and then, if the distances are above a set threshold, then the model identifies two identities as being different (else they are the same). The entire motivation of the FA technique is that if we suppress the noise of these cropped images, then the network will be able to create better embedding representation of the object, thereby performing better decisions as to whether two identities are the same or not. We form the hypothesis that the network performance may improve if we perform FA without any additional training to the embedding network. However, we also evaluate whether

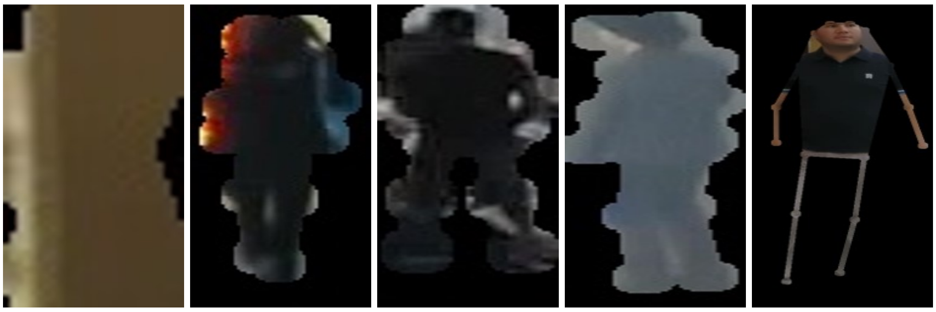


Fig. 3: Segmented Images at Specific Points of Keypoint Prediction Confidence Distribution (from left to right: Minimum, 1st Quartile, Median, 3rd Quartile, Maximum)

training this embedding network with FA images using triplet loss will also provide any gains in performance. In fig. 4, we show a visual representation of how triplet loss images appear when and when not using FA.

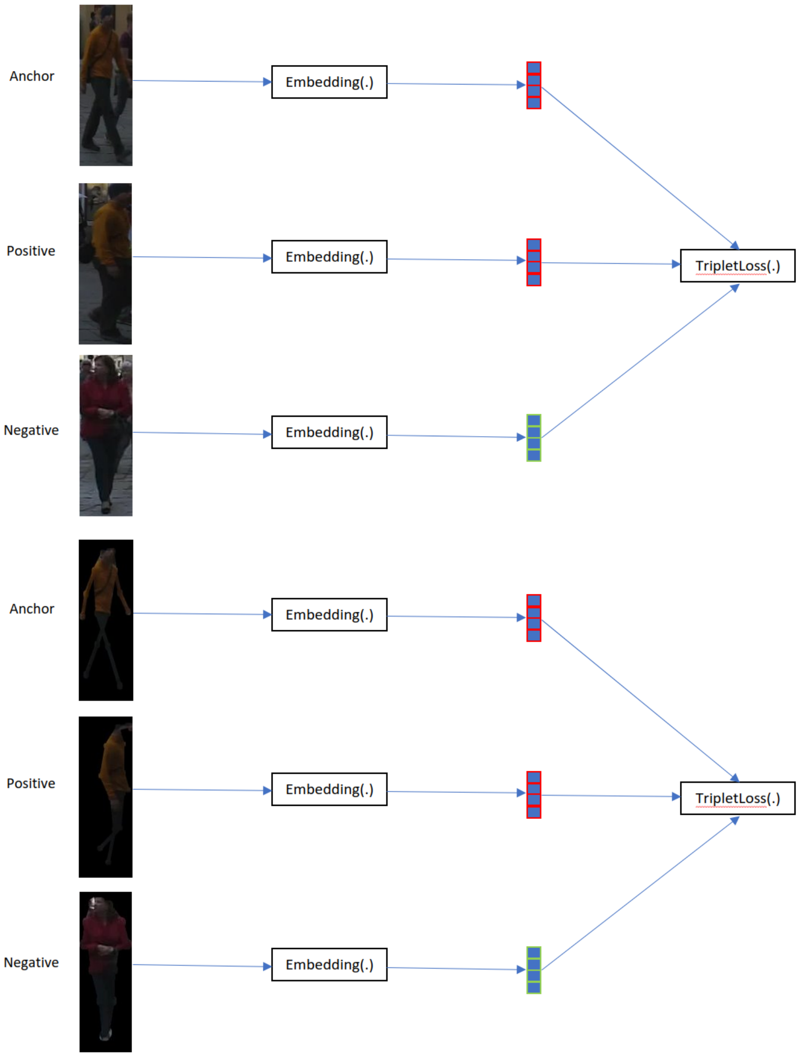


Fig. 4: Triplet Loss training without (top) and with (bottom) Feature Amplification on MOT17 training data

4. Experiments

4.1. Training and Testing Datasets

To perform our experiments, we evaluated all our models on the MOT17 dataset. This dataset is convenient because it provides bounding box detections for all videos from three different models: DPM, FRCNN, and SDP. Since we are interested in the improvement of the network embedding, it is convenient to have a set of pre-defined bounding box detections. For the evaluation of the pre-trained Deep SORT, we will use the latter half of all the videos as our evaluation dataset in which we will calculate the MOT metrics. Since the MOT17 does not have the ground truth of test data available to the public to maintain academic integrity, the pre-trained Deep SORT is trained using the former half of the training dataset. Example images from the datasets used can be found in appendix A.

4.1.1. MOT17

The MOT dataset [13] is widely used to evaluate the performance of computer vision algorithms on multiple object tracking tasks such as object detection, short term tracking, object flow etc. This dataset comprises of high resolution videos with multiple pedestrians in different scenarios. It consists of 4 training and 4 test videos. Each video is captured from different viewpoint with pedestrians walking in complex scenes such as shopping malls and crowded streets. The ground truth annotations provided for each video is by frame with position, size of each pedestrian bounding box, unique ID and visibility over time. The MOT17 dataset is a popular benchmark for state-of-the-art multiple object tracking algorithms.

4.1.2. ReID

The Re-Identification (ReID) dataset includes two components - image data and metadata information. Re-identification is a computer vision task to identify a person across multiple non-overlapping camera viewpoints. Every image is of different resolution, size and format. There is a possibility of performing some kind of pre-processing on those images. The metadata information consists of the unique identity of the person in the image and the image location or any other information that help identify the person in the image. This metadata is used to train and evaluate ReID models that learn feature representation that can be used to match and identify individuals across different camera viewpoints. To self-train the mmpose model with only the re-embedded images, we need to extract each bounding box in the MOT17 training set, splitting it into 80-20 train-validation partition by setting a visualization threshold and validation split percentage.

4.1.3. COCO

The Common Objects in Context [14] dataset is a popular computer vision dataset for object detection, segmentation and captioning tasks. It contains over 330,000 images with more than 2.5 million object instances labeled across 80 different object categories. The dataset format includes two components - image data and annotation data. Every image is in high-resolution JPEG format with unique image IDs. The annotation consists of information about objects in the image. All annotations are stored in a JSON file. Each annotation record includes Image ID, Object ID, Category ID, Bounding box coordinates of the object and segmentation mask of the object. It includes a pre-defined set of training, validation and testing images. In addition to the bounding boxes, we have identified the pose or body orientation of the individual, location and visibility; by taking into consideration the keypoints of joints in an individual, whether the individual is in crowd and if the person is visible in the given instance and a confidence score is assigned.

4.1.4. Feature Amplified ReID

This technique is used to enhance the performance of ReID algorithms. It involves augmenting ReID dataset with additional features to help distinguish between similar individuals with more certainty and improve the performance of the model. Based on the COCO keypoints identified, the image is reconstructed. This results in the area of the reconstructed image lesser than the original and highlighting the keypoint features that represent a person using mmpose and masks the original images with the filled contour of the skeleton obtained from mmpose.

4.2. Evaluations

To evaluate the performance of our method with the standard DeepSORT implementation, we tested seven different configurations of the pose model and the embedding network. The configurations specifically differ on how the model was trained and for how many epochs. For the pose model, this means whether it was trained solely on COCO or additionally through MOT17 via self-learning. For the embedding, this means whether it was trained under the ReID dataset or the FA ReID dataset. Before we analyze the performance as compared to DeepSORT, we first want to select the best combination between the pretrained pose model on COCO dataset with either the embedding network trained under ReID or FA ReID datasets. Similarly, we also want to select the best combination between self-learned pose model with either the embedding network trained under ReID or FA ReID datasets. For all of the evaluations, we use a set of metrics known as CLEAR MOT metrics [15]. The MOT metrics can be difficult to understand and require background reading. As such, we describe the most important metrics pertaining to this project: Multiple Object Tracking Accuracy (MOTA): combination of false positives and negatives; ID switches (IDs): target switching from a true positive ID to another identity; FragMentations (FM): number of times an ID goes from tracked to no ID); Mostly Tracked (MT): number of object that were tracked for 80% or higher of their lifetime; Mostly Lost (ML): number of object that were tracked for 20% or less of their lifetime, and Partially Tracked (PT): $1 - MT - ML$. [16] describes these and the rest of the metrics in detail.

4.2.1. Pose Evaluations

In this section, we compare the performance of the pose model trained under the COCO dataset with different training schemes on the embedding network. There are three configurations: (i) pose with reid network trained on ReID (pose-reid) for six epochs, (ii) pose with reid network trained on FA ReID for 18 epochs (pose-fareid18), and (iii) configuration (ii) but embedding is trained for 6 epochs (pose-fareid6). The DeepSORT embedding network was trained for six epochs and as such, we also trained the FA embedding network with six epochs. We also trained the FA embedding with eighteen epochs (twice the original amount) to evaluate whether performance would improve. Table 1 shows the performance of each configuration where the ranking from best to worst can be read from the top row to the bottom row. These results surprised us because we theorized that training the embedding network with FA would give better results than if we did not. This is because we would be explicitly teaching the network to ignore the noise background and only focus on the signal. Apart from this, results of FA embedding did improve when we extended training to 18 epochs instead of six.

Table 1: Pose MOT Metrics

Name	HOTA↑	IDF1↑	IDP↑	IDR↑	MOTA↑	MOTP↑	MT↑	FM↓	FN↓	FP↓	IDs↓	ML↓	PT↓
pose210-reid6	**0.458**	**0.534**	**0.81**	**0.399**	**0.469**	0.149	**209**	**3469**	82447	286	**3140**	362	**446**
pose210-fareid18	0.412	0.462	0.70	0.344	.453	**0.150**	202	3754	82456	295	5730	**358**	457
pose210-fareid6	0.435	0.510	0.773	0.380	0.440	0.149	205	3683	**82445**	**284**	7752	365	447

4.2.2. SL Pose Evaluations

In the previous section, we demonstrated different configurations for the pose model trained under the COCO dataset. In this section, we will evaluate the performance of using a Self-Learn (SL) pose model with different training schemes of epochs on the embedding network. Namely, we evaluate: SL pose with twelve epoch with ReID trained embedding network and for the rest, SL pose with 106 epochs trained with embedding network for six epochs and FA embedding trained for one and six epochs. The reason we trained SL pose for 12 epochs was to avoid any potential overfit with the data. From here, since COCO dataset contains about 80,172 images and because the pose model was trained for 210 epochs, we decided to take the ratio and multiply it by the number of MOT17 training images with was trained for 210 epochs. This process is shown in Listing 1.

Listing 1: SL pose epoch calculation

```
nTrainMOTImgs = 20086
nCOCOImgs = 80172
poseCOCOTrainedEpochs = 210
poseCOCORatio = poseCOCOTrainedEpochs / nCOCOImgs
slposeMOT17TrainedEpochs = floor(nTrainMOT17Imgs*poseCOCORatio) + 2
```

Table 2 demonstrates the performance of all configurations where the top row is the best performance and the bottom is the worst. Like the previous section, we see that the best performance was attained when we trained the embedding network on the ReID dataset (instead of it's FA counterpart). However, unlike the previous table, a close runner-up is the configuration where SL pose is trained on one-hundred and six epochs with a pretrained FA embeding network for just one epoch. It is interesting to note how the SL pose performance begins to downgrade once we train for for further than twelve epochs. Perhaps since we are performing self-learning from the COCO dataset, this does not need nearly as much training as was done with COCO. Perhaps the same could be said about the FA embedding network (as training for one epoch outperforms training for six). However, the most interesting highlight is that this table follows a similar pattern to our previous section: SL performs best when the embedding network is trained under the same configuration as DeepSORT. This further adds evidence that instead of attaining gains from training the FA embedding network, we see a dwindling of performance.

Table 2: SL Pose Metrics

Name	HOTA↑	IDF1↑	IDP↑	IDR↑	MOTA↑	MOTP↑	MT↑	FM↓	FN↓	FP↓	IDs↓	ML↓	PT↓
SLpose12-reid6	0.387	0.422	0.64	0.315	**0.452**	0.149	207	3558	**82447**	**286**	**5922**	**359**	451
SLpose106-fareid1	**0.393**	**0.432**	**0.655**	**0.322**	0.450	0.148	**208**	**3323**	82455	294	6114	365	**444**
SLpose106-fareid6	0.371	0.404	0.613	0.302	0.444	0.149	206	3566	82453	292	7070	360	451
SLpose106-reid6	0.353	0.377	0.572	0.282	0.442	**0.150**	202	3614	82458	297	7500	364	451

4.2.3. Comparing DeepSORT

In this section, we will be comparing the performance of DeepSORT against the best pose and SL pose configurations from the previous two sections. Table 3 demonstrates the performances where the top row is the best performance and the bottom row is the worst performance. In almost every metric, DeepSORT significantly outperforms either of the DeepFASORT implementations. This adds evidence that our current FA implementation actually downgrades performance of the base DeepSORT. There are two primary reasons why we think this is happening. The first is that our FA implementation is not perfect. As we see in the rightmost image of fig. 3, the FA implementation blacks out most of the signal of the legs and arms of the person. Figure 5 represents the dimensions of the image in red. From this we see that there is a large cluster of images with their dimensions similar to the one that produced that maximum pose prediction confidence. This is important because this means that there is a large cluster of high resolution images where our FA implementation may do more damage by blacking out the important features of the person. A possible solution for this is to create an adaptive mask where skeleton connection widths are determined by the resolution of the image. In this sense, we would want to increase the width of the skeleton to amplify most of the true signals. Second (and perhaps the most interesting), is that background noise may actually help the model make better distinguishing decisions. This is because if the model compares the embeddings of two people who are spatially in significantly different areas of each other but their appearance may be similar, then the model may analyze the context of the background to determine whether these two could potentially be around the same vicinity. This is even more so because the MOT17 dataset contains videos or sequential images. This means that between two consecutive frames, there may not be much change in either the background or the signal of the person itself. This knowledge will be instilled when training the embedding network with the ReID dataset because we do not expect an identity to have radically different background noises. Thus, the model may not be learning the appearance cues of the people as much as we think. Because of this, even if we perform FA with a segmentation model to attain better retention of the noise, the performance might still downgrade when compared to the original DeepSORT.

Table 3: DeepSORT and DeepFASORT Metrics

Name	HOTA↑	IDF1↑	IDP↑	IDR↑	MOTA↑	MOTP↑	MT↑	FM↓	FN↓	FP↓	IDs↓	ML↓	PT↓
DeepSORT	**0.502**	**0.608**	**0.922**	**0.453**	**0.481**	0.148	**209**	**3350**	**82444**	**283**	**1199**	364	**44**
pose210-reid6	0.458	0.534	0.81	0.399	0.469	**0.149**	**209**	3469	82447	286	3140	362	446
SLpose12-reid6	0.387	0.422	0.64	0.315	0.452	**0.149**	207	3558	82447	286	5922	**359**	451

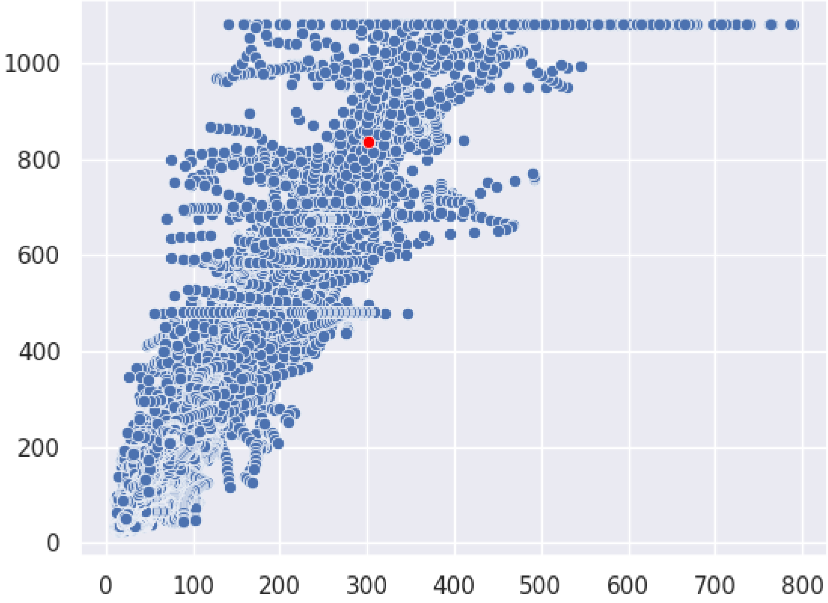


Fig. 5: MOT17 image dimensions with x as width and y as height. The red dot highlights the dimension of the image with the maximum confidence prediction of the pose model.

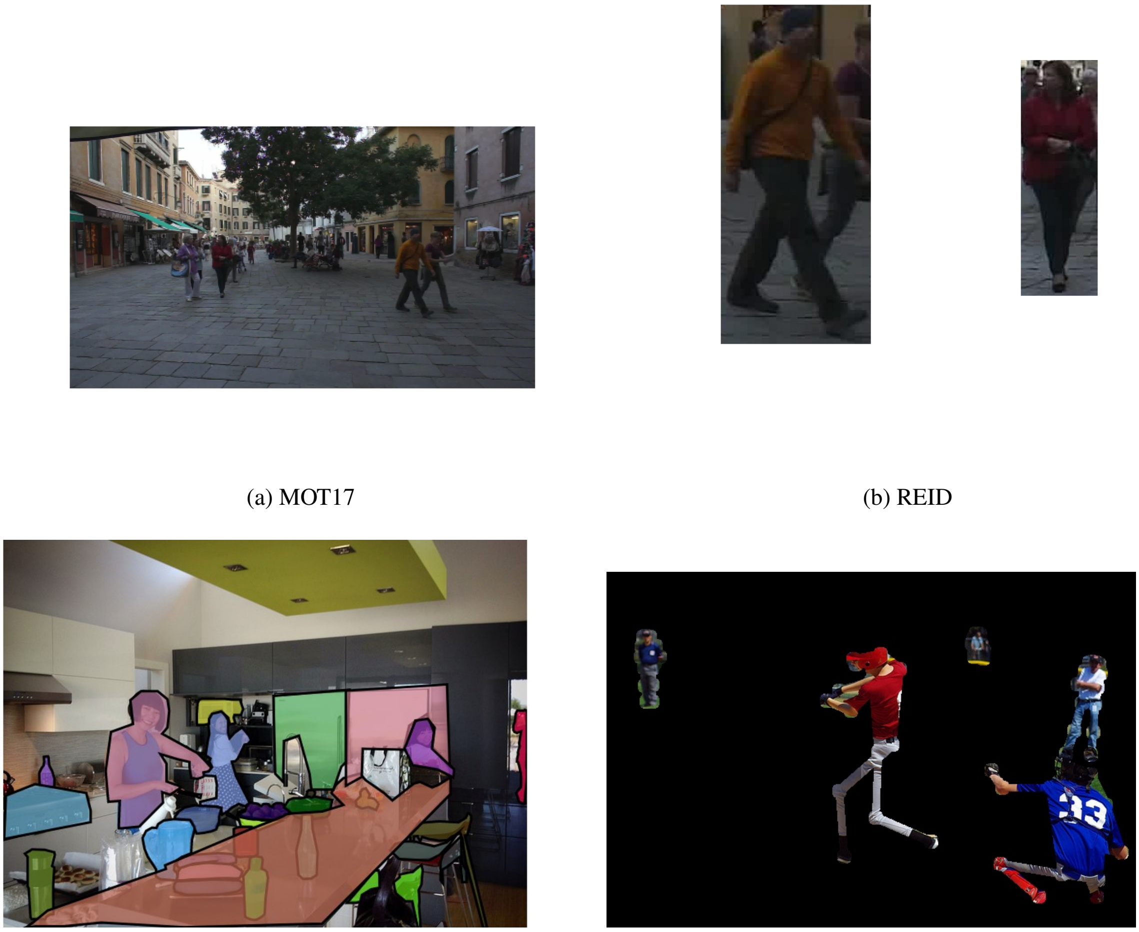
5. Conclusion

In this project, we set out to evaluate whether a Keypoint-based FA implementation would improve DeepSORT. Through the different model configurations, we found that our current implementation of FA is actually worsening the performance of DeepSORT. This may be not necessarily because of our implementation of FA, but rather more so that the FA, by blacking the background noise, is actually taking away spatial cues that DeepSORT is harnessing to make decisions. In other words, the background noise may serve as spatial signal. This was not a conclusion that we were expecting. We assumed that the embedding network attempts to muffle the background noise and retain only the signal to make decisions. If this was true, then DeepFASORT may indeed provide better performance than its original implementation. However, as this seems to be the actual case, because the model may harness contextual background noise, we are actually taking away important signals to make better distinguishing decisions.

References

1. Y. Zhang, P. Sun, Y. Jiang, D. Yu, Z. Yuan, P. Luo, W. Liu, and X. Wang, “Bytetrack: Multi-object tracking by associating every detection box,” arXiv preprint arXiv:2110.06864 (2021).
2. N. Wojke and A. Bewley, “Deep cosine metric learning for person re-identification,” in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV),* (IEEE, 2018), pp. 748–756.
3. A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in *2016 IEEE International Conference on Image Processing (ICIP),* (2016), pp. 3464–3468.
4. N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *2017 IEEE International Conference on Image Processing (ICIP),* (IEEE, 2017), pp. 3645–3649.
5. P. Bergmann, T. Meinhardt, and L. Leal-Taixe, “Tracking without bells and whistles,” in *Proceedings of the IEEE international conference on computer vision,* (2019), pp. 941–951.
6. Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, “Strongsort: Make deepsort great again,” IEEE Transactions on Multimed. (2023).
7. D. Mitzel and B. Leibe, “Real-time multi-person tracking with detector assisted structure propagation,” in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops),* (2011), pp. 974–981.
8. A. Milan, K. Schindler, and S. Roth, “Detection- and trajectory-level exclusion in multiple object tracking,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition,* (2013), pp. 3682–3689.
9. L. Zhang, Y. Li, and R. Nevatia, “Global data association for multi-object tracking using network flows,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition,* (2008), pp. 1–8.
10. W. Hu, X. Li, W. Luo, X. Zhang, S. Maybank, and Z. Zhang, “Single and multiple object tracking using log-euclidean riemannian subspace and block-division appearance model,” IEEE Transactions on Pattern Analysis Mach. Intell. **34**, 2420–2440 (2012).
11. D. Mitzel, E. Horbert, A. Ess, and B. Leibe, “Multi-person tracking with sparse detection and continuous segmentation,” in *Computer Vision – ECCV 2010,* K. Daniilidis, P. Maragos, and N. Paragios, eds. (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010), pp. 397–410.
12. A. Andriyenko, S. Roth, and K. Schindler, “An analytical formulation of global occlusion reasoning for multi-target tracking,” in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops),* (2011), pp. 1839–1846.
13. A. Milan, L. Leal-Taixé, I. D. Reid, S. Roth, and K. Schindler, “MOT16: A benchmark for multi-object tracking,” CoRR **abs/1603.00831** (2016).
14. T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Doll’a r, and C. L. Zitnick, “Microsoft COCO: common objects in context,” CoRR **abs/1405.0312** (2014).
15. K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The clear mot metrics,” EURASIP J. on Image Video Process. (2008).
16. W. Luo, X. Zhao, and T. Kim, “Multiple object tracking: A review,” CoRR **abs/1409.7618** (2014).

A. Dataset Examples



(a) MOT17

(b) REID

(c) COCO

(d) FA-ReID

Fig. 6: Example Images from Datasets Used