Forward Propagation, Backward Regression, and Pose Association for Hand Tracking in the Wild

Mingzhen Huang[1,2], Supreeth Narasimhaswamy[1], Saif Vazir[1,3], Haibin Ling[1], Minh Hoai[1,4]
[1]Stony Brook University, [2]University at Buffalo, [3]Tulip Interfaces, [4]VinAI Research

Abstract

We propose HandLer, a novel convolutional architecture that can jointly detect and track hands online in unconstrained videos. HandLer is based on Cascade-RCNN with additional three novel stages. The first stage is Forward Propagation, where the features from frame `t-1` are propagated to frame `t` based on previously detected hands and their estimated motion. The second stage is the Detection and Backward Regression, which uses outputs from the forward propagation to detect hands for frame `t` and their relative offset in frame `t-1`. The third stage uses an off-the-shelf human pose method to link any fragmented hand tracklets. We train the forward propagation and backward regression and detection stages end-to-end together with the other Cascade-RCNN components.

To train and evaluate HandLer, we also contribute YouTube-Hand, the first challenging large-scale dataset of unconstrained videos annotated with hand locations and their trajectories. Experiments on this dataset and other benchmarks show that HandLer outperforms the existing state-of-the-art tracking algorithms by a large margin. Code and data are available at https://vision.cs.stonybrook.edu/~mingzhen/handler/.

1. Introduction

Hand tracking is an important problem in various application scenarios, from gesture and activity recognition to contact tracing and skill evaluation. One approach for tracking hands is to consider them as parts of a human body and then perform hand tracking based on the tracked human pose. But pose detection and tracking can be unreliable by itself, especially for people that are partially occluded or outside the field of view of the camera. Another approach for hand tracking is to use off-the-shelf tracking methods. Unfortunately, single-object trackers are not appropriate for tracking multiple hands, while existing multiple-object trackers do not work well for hands even though they have shown impressive performance for tracking pedestrians and vehicles [2, 5, 19, 47, 49, 50, 62]. Hand tracking is difficult because hands are not ordinary objects, given the extreme articulation of hands and the frequent interaction of hands with other objects. In a short period of a few frames, the size, shape, location, and visibility of a hand can change dramatically and frequently. Many existing multiple-object trackers use the detection and assoication paradigm. However, hand detection would fail in the presence of motion blur and occlusion, while hand linking across time is difficult as the size, location, pose, and appearance of a hand can change drastically. Simultaneously, two different hand instances might look alike, so distinguishing them would be difficult even for a sophisticated re-identification module that has been trained specifically for hands.

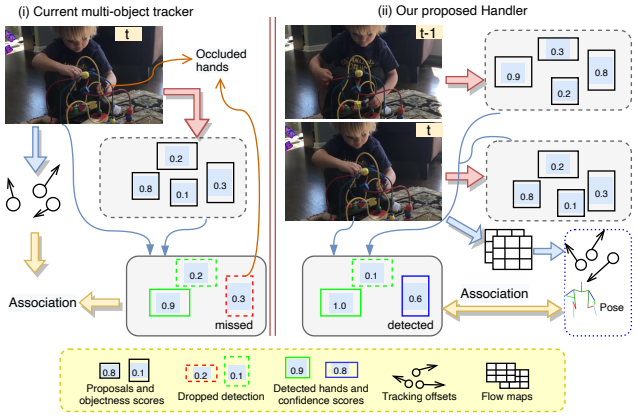


Figure 1. We develop an advanced detection and association algorithm for tracking multiple hands. Different with previous methods that estimate probability of a detected hand on the objectness scores at current frame only, we estimate the probability based on both the objectness scores at frames `t` and `t-1`, and associate hands across frame via both pose and tracking offsets.

In this work, we develop a novel convolutional architecture that can detect and track hands in unconstrained videos. We name the proposed architecture HandLer, which stands for *Hand Linker*. HandLer takes as input two consecutive video frames at times `t-1` and `t`, and output the detected hands in frame `t` as well as their corresponding locations in frame `t-1`. The processing pipeline consists of three stages. The first stage is the Forward Propagation, which propagates features from frame `t-1` to frame `t` based on the locations of previously detected hands and their estimated movements. The second stage is the Detection and

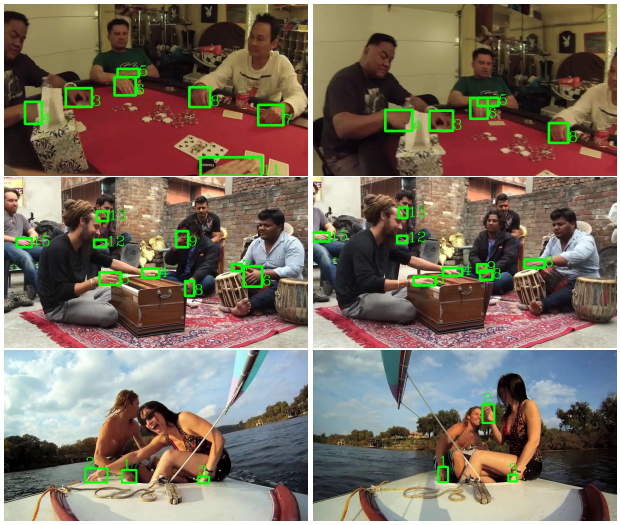


Figure 2. Representative image sequences from our dataset. The size, shape, location, appearance, and visibility of a hand can change drastically and frequently.

Backward Regression that uses outputs from the Forward Propagation to obtain the hand locations for frame `t` as well as their counterparts in frame `t-1`, and estimate their confidence conditioned on both the objectness scores at frames `t` and `t-1` as shown in Fig. 1. This allows us to link hand detections between two frames. Third, we establish correspondence between hand tracklets via pose association. This is to leverage the fact that hands are undetachable parts of a human body, so we can use pose to recover prematurely terminated hand tracklets.

Each stage of the proposed processing pipeline has its benefits. The propagation step and conditional confidence estimation step are useful for detecting blurry and occluded hands. The detection step is necessary to account for new hands in a video and avoid the potential drifting problem common in methods solely based on propagation. The regression step brings detections at two different times into a common reference frame for a more reliable linking. The high-level pose association step avoids frequent ID switch due to motion blur and occlusion.

We also introduce a new dataset called YouTube-Hand for developing and evaluating hand tracking algorithms. YouTube-Hand contains 240 video sequences from diverse scene categories, including kitchens, mechanical workshops, and gyms. This dataset has 19,728 annotated frames with 864 unique hand instances. To the best of our knowledge, this is the first large-scale hand tracking dataset containing videos for unconstrained environments with multiple annotated hand trajectories for each video. Fig. 2 shows some representative images from our dataset. We will release this dataset and code for research usage.

2. Related Work

Many prior works only focused on hand detection in static images [6, 8, 9, 13–15, 22, 26–28, 32, 37, 38, 40, 52, 55], and works on hand tracking were developed for constrained settings such as laboratory environments and egocentric perspectives. Sridhar et al. [42] proposed a method to track hands captured using a depth camera. Zhang et al. [60] proposed a hand tracking solution that predicted a hand skeleton of a human from a single RGB camera for AR/VR applications. Wang and Popović [46] used a single camera to track a gloved hand with an imprinted pattern. Sharp et al. [39] provided a hand tracking and pose estimation system based on a single depth camera. Mueller et al. [24] developed a 3D hand tracking approach for monocular RGB videos using a kinematic 3D hand model. Sridhar et al. [43] proposed a method to track hands manipulating objects in RGB-D videos. However, none of these methods was developed for videos in the wild; they required special markers, depth information, ego-centric perspectives, or scenes with plain background.

Hands are objects, and we can consider Multiple-Object Tracking (MOT) methods. A popular MOT approach is tracking-by-detection, where an object detector first localizes objects, and then an association method constructs trajectories. Depending on the association method, we can categorize MOT methods as offline tracking or online tracking. Given a current frame `t`, offline methods [35, 56, 57] can use future frames and pose the association as a global optimization method. Meanwhile, most online methods [33, 54, 58, 63] are constrained to use frames up to frame `t` only. A typical way to associate detections over different frames is the Hungarian algorithm [25] with the affinity costs defined based on the overlaping criterion. Bewley et al. [4] proposed to predict bounding box movement with Kalman Filter and use Hungarian algorithm for linking those boxes into tracks. However, this approach does not work well for unconstrained videos since hands often move fast, interact, and cross each other. Moreover, the two-step approach of detecting hands first and then associating them can lead to suboptimal results since the two steps are not jointly optimized end-to-end.

There are existing methods to alleviate the disadvantages of the two-step tracking-by-detection paradigm. Bergmann et al. [2] developed a framework that uses object locations in the current frame to directly regress their corresponding locations in the next frames. However, this method only uses current frame object locations as region proposals for the next frame. This method does not work well for tracking hands since hand locations change drastically over frames. Zhou et al. [62] proposed a point-based framework for joint detection and tracking, representing each object by a single point and tracking such points. This method outputs an offset vector from the current object center to its center

in the previous frame for tracking. However, only using a point representation does not work well for hands, which are highly deformable.

There are methods that process multiple frames at the same time. Feichtenhofer et al. [10] introduced correlation features that represented object co-occurrences across time to generate two-frame tracklets. However, this method does not work well when an object undergoes heavy occlusions, which is often the case for hands. Peng et al. [30] extended [10] by adding an appearance-based identity attention and proposed an online method to link two-frame tracklets. Wu et al. [51] proposed to generate a re-ID embedding in each pixel and estimate objects movement offset from this embedding. This offset can be used to propagating feature and associating objects. However, those algorithms are appearance-based that does not work well for hands since the appearance of a hand can change drastically over time and different hand instances can have similar appearance. Instead of using correlation features or appearance-based approaches, our method directly estimates the relative offsets of hands in the previous frame given the hand locations in the current frame. As shown in our experiments, this makes our hand tracking system more robust to occlusions or motion blur and reduces identity switches with other hand instances.

3. Proposed Method

In this section, we describe our novel method for online tracking of multiple hands. We illustrate the proposed architecture in Fig. 3. Our method's core is a convolutional network that operates on a pair of two consecutive frames at a time. At time `t`, the input to the network is a pair of video frames at time `t-1` and `t`, and the output of the network are locations and confidence scores of the detected hands in frame `t` as well as their corresponding locations and confidence scores in frame `t-1`. We use the estimated locations of hands in time `t-1` to establish the association with the existing hand tracks, assuming that we have tracked hands in the video until time `t-1`.

Specifically, given two frames $\mathbf{I}_{t-1}$ and $\mathbf{I}_t$ at time `t-1` and `t`, we use a backbone network to obtain their features $\mathbf{X}_{t-1}, \mathbf{X}_t \in \mathbb{R}^{h\times w\times d}$. Here, $h \times w$ denotes the spatial size and d denotes the number of channels. We also use an off-the-shelf pose tracker [29] to obtain pose heatmaps $\mathbf{P}_{t-1}, \mathbf{P}_t \in \mathbb{R}^{h\times w\times 15}$ corresponding to 15 human joints. Let $\mathbf{H}_{t-1} \in \mathbb{R}^{h\times w}$ denote heatmap for hands that were detected in frame $\mathbf{I}_{t-1}$. We pass features $\mathbf{X}_{t-1}$ and $\mathbf{X}_t$, pose heatmaps $\mathbf{P}_{t-1}$ and $\mathbf{P}_t$, and hand heatmap $\mathbf{H}_{t-1}$ to the forward propagation stage.

3.1. Forward Propagation

Given features $\mathbf{X}_{t-1}$ and $\mathbf{X}_t$, pose heatmaps $\mathbf{P}_{t-1}$ and $\mathbf{P}_t$, and hand heatmap $\mathbf{H}_{t-1}$, the forward propagation stage estimates a flow map $\mathcal{F}_t \in \mathbb{R}^{h\times w\times 2}$ and uses this flow map to obtain temporally aggregated features $\mathbf{Z}_t \in \mathbb{R}^{h\times w\times d}$.

Flow Estimation. To estimate the flow map $\mathcal{F}^t$, we propose to use the Flow Estimation Network [12]. The inputs to this network are multi-scale features $\mathbf{X}_{t-1}$ and $\mathbf{X}_t$, and the output is the 2-channeled flow map $\mathcal{F}_t \in \mathbb{R}^{h\times w\times 2}$ denoting the motion between frames $\mathbf{I}_{t-1}$ and $\mathbf{I}_t$. The two channels in $\mathcal{F}_t$ corresponds to flows in the horizontal and vertical directions.

We train this Flow Estimation Network end-to-end along with other components of the network as follows. Given a pair of hands that have the same ID in frames `t-1` and `t`, we obtain two binary masks $\mathbf{M}_{t-1}, \mathbf{M}_t \in \mathbb{R}^{h\times w}$ corresponding to two frames. These masks are the ground-truth binary segmentation maps for the hand in frames `t-1` and `t`, respectively. We then use a bilinear warping function $\mathcal{W}$ proposed by [65] to estimate a binary segmentation map for the hand at time `t`: $\mathbf{M}'_t = \mathcal{W}(\mathbf{M}_{t-1}, \mathcal{F}_t)$. We then define a loss for hand motion as the MSE loss between estimated $\mathbf{M}'_t$ and the groundtruth $\mathbf{M}_t$: $L_{hmo} := \text{MSE}(\mathbf{M}'_t, \mathbf{M}_t)$.

Similarly, we also use the pose heatmap pair $(\mathbf{P}^{t-1}, \mathbf{P}^t)$ to define a loss for pose motion. We first obtain an estimated pose at time `t`: $\mathbf{P}'_t = \mathcal{W}(\mathbf{P}_{t-1}, \mathcal{F}_t)$. We then define a loss for poss motion as the MSE loss between estimated $\mathbf{P}'_t$ and the groundtruth $\mathbf{P}_t$: $L_{pmo} := \text{MSE}(\mathbf{P}'_t, \mathbf{P}_t)$.

Temporal Feature Aggregation. The output $\mathcal{F}_t$ from the Flow Estimation Network is used to aggregate features from time `t-1` to features from time `t`. Specifically, we propagate features $\mathbf{X}_{t-1}$ to features $\mathbf{X}_t$ to obtain $\mathbf{Z}_t$:

$$\mathbf{Z}_t = [1+\mathcal{W}(\mathbf{H}_{t-1}, \mathcal{F}_t)]\odot\mathbf{X}_t + \mathcal{W}(\mathbf{H}_{t-1}\odot\mathbf{X}_{t-1}, \mathcal{F}_t) \quad (1)$$

In the above equation, $\odot$ is the Hadamard product, $\mathcal{F}_t$ is the estimated flow map from frame `t-1` to frame `t`, and $\mathcal{W}$ is the bilinear warping function.

3.2. Hand detection and backward regression

The second important component of our architecture is the hand detection and backward regression module. The input to this module is the propagated feature map $\mathbf{Z}_t$ along with estimated flow map $\mathcal{F}^t$. First, a CenterNet [61] will be used to obtain a dense set of hand proposals at every pixel. Second, for each proposal we compute: (1) the bounding box of the hand at frame `t`, (2) the probability of this bounding box being a hand, (3) the relative offset bounding box of this hand at frame `t-1`, and (4) the the confidence of detected box and offset box belong to same hand identity.

Tracking-based detection. We observe that for some blurry and occluded hands, our model would yield relative lower confidence scores even though those hands are clearly visible in previous frames. Detections with low confidence scores might be dropped, leading to false negatives. To address this problem, we formulate the detection probability at frame `t` to be conditioned on both the objectness

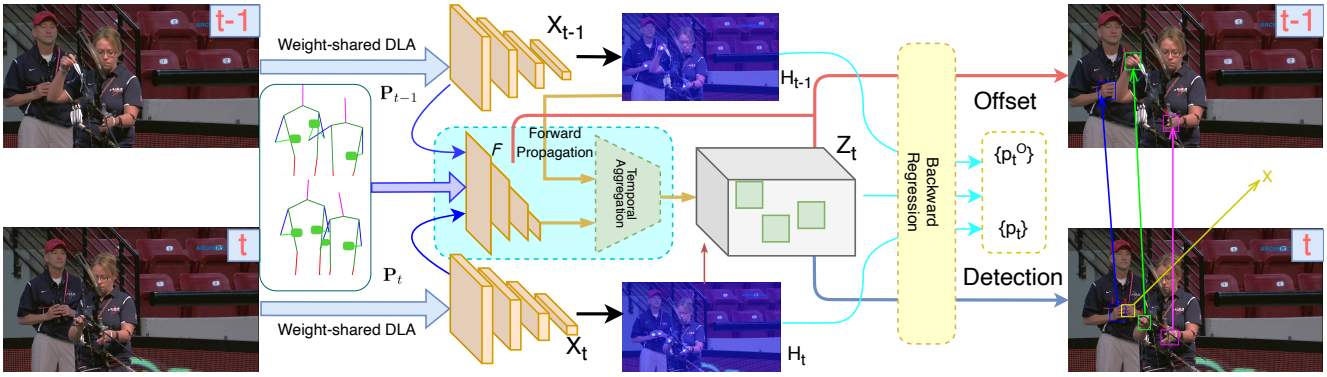


Figure 3. **Processing pipeline of HandLer.** Given input video frames at time t−1 and time t, we first extract their DLA features $\mathbf{X}_{t-1}$ and $\mathbf{X}_t$. We estimate the flow map O from frame t−1 to frame t, and also obtain a heatmap $\mathbf{H}_t$ in frame t from CenterNet [61]. Along with heatmap $\mathbf{H}_{t-1}$, we aggregate feature as described in Eq. (1) to obtain a feature map $\mathbf{Z}_t$. We then extract RoI features from $\mathbf{Z}_t$ and detect hands in frame t and also estimate their corresponding offset and probability in the frame t−1 with the backward regression.

scores at frames t and t−1 (shown in Fig. 1). Consider a proposal C_k^t at time t and position k and its corresponding detection $\mathcal{D}_k$ for the anchor box at location k, we use $P(\mathcal{D}_k) = P(\mathcal{D}_k{=}hand)$ to denote the probability that $\mathcal{D}_k$ is a hand, and $P(C_k^t) = P(C_k^t = object)$ to denote the objectness probability for the proposal C_k^t. The detection likelihood is formulated as:

$$P(\mathcal{D}_k) = P(\mathcal{D}_k|C_k^t)P(C_k^t) \quad (2)$$

$$= P(\mathcal{D}_k|C_k^t)\sum_j^N P(C_k^t|C_j^{t-1})P(C_j^{t-1}). \quad (3)$$

We further assume that $P(C_k^t|C_j^{t-1}) = 0$ if there is no motion from j to k, and $P(C_k^t|C_j^{t-1}) = P(C_k^t)$ otherwise. Thus the detection likelihood becomes:

$$P(\mathcal{D}_k) = P(\mathcal{D}_k|C_k^t)\sum_{j\in\mathcal{F}_k^t} P(C_k^t)P(C_j^{t-1}), \quad (4)$$

where $\mathcal{F}_k^t$ denotes the set of pixel locations that have motion vectors pointing to k in the optical flow map $\mathcal{F}^t$.

3.3. Hand-track continuation and initialization

We now describe how a newly detected hand is linked with an existing hand track or used to initialize a new hand track. Consider a particular detected hand $\mathcal{D}$ obtained by running the detection module with the input being the two frames at t−1 and t. Frame t detection $\mathcal{D}_t$ is represented by a quadruple: $\mathcal{D}_t = (B_t, p_t, B_t^O, p_t^O)$, where B_t is the hand location in frames t, B_t^O is its corresponding offset location in frame t−1, p_t is the corresponding detection confidence and p_t^O is the confidence of B_t and B_t^O belong to same hand identity. Note that we only keep a detection where p_t is greater than a detection threshold θ_{det}.

We then use the Hungarian algorithm [25] to match a detection $\mathcal{D}_t^i$ and also other detections with the set of existing hand tracks. This is a joint optimization process, where the best set of one-to-one correspondences is determined. If $\mathcal{D}_t^i$ is matched with an existing hand track, we will use it to continue the track. Otherwise, we will either initialize a new hand track for B_t^i if the detection score p is higher than a threshold θ_{new}, or discard this detection. Note that θ_{new} should be higher than θ_{det} to avoid propagating false positives. In our experiments, we set $\theta_{det} = 0.6$, and $\theta_{new} = 0.9$.

The Hungarian matching process is done as follows. The inputs to this process are: (1) a set of detected hands, represented by the set of bounding boxes $\{\mathcal{D}_t\}$ in frame t, and (2) a set of active hand tracks, represented by a set of last bounding boxes of the tracks $\{\mathcal{T}_{t-1}\}$. Note that the last bounding box of $\mathcal{T}_{t-1}$ might not be at frame $t-1$. Following previous MOT methods, we only remove a hand track from the set of active hand tracks if this hand track is not matched to any new detection for more than σ frames. Given the two set of bounding boxes $\{B^O\}$ and $\{\mathcal{T}_{t-1}\}$, we obtain an affinity matrix $\mathcal{M}$ as $\mathcal{M}^{ij} = (\alpha + p_t^O)IoU(B_t^{Oi}, \mathcal{T}_{t-1}^j)$, and use the Hungarian algorithm to find the best set of one-to-one correspondences to maximize the total sum of the affinity. In our experiments, we set $\alpha = 0.1$ and $\sigma = 50$.

3.4. Pose association

Since hands are undetachable body parts of a human, we propose to use tracking result to guide our model for hand motion estimation and tracking. Specifically, we consider the state-of-the-art open source pose tracking algorithm LightTrack [29] and observe that it has a lower recall than our hand tracker, but most detected poses are generally accurate. We therefore propose to use LightTrack [29] to help estimate motion flows (described in Sec. 3.1) and link a newly detected hand to an existing hand track.

Also, recall that a newly detected hand is represented by a quadruple $\mathcal{D} = (B, p, B^O, p^O)$. In most cases, this detection will be used to continue a hand track as described in Sec. 3.3. In some cases, we will discard $\mathcal{D}$ if p is low, and we will create a new track if either p^O is low or there is no matching hand track for B^O. But these actions can lead to

a false negative or a false identity switch, so we propose to address these problems with pose tracking as follows. First, given a set of detected hands and a set of wrist locations of detected poses, we run the Hungarian algorithm to find the optimal matching, where the matching cost for a hand and a wrist is based on their distance. Second, we discard detections that have low p values and no matching wrists. Third, we use the procedure described in Sec. 3.3 to link some detected hands with existing hand tracks. For a detected hand $\mathcal{D}$ that has not been linked to any hand track, we will link it with a hand track $\mathcal{T}$ if: (1) $\mathcal{D}$ is linked with the right/left wrist of a pose $\mathcal{P}_t$ in frame $\mathtt{t}$; (2) $\mathcal{T}$ is linked with the right/left wrist of a pose $\mathcal{P}_{t-1}$ in frame $\mathtt{t-1}$; and either (3a) $\mathcal{P}_t$ and $\mathcal{P}_{t-1}$ are linked via pose tracking, or (3b) the left/right wrist of $\mathcal{P}_t$ is linked with another hand $\mathcal{D}'$ that is linked with the hand track $\mathcal{T}'$, which in turn is linked with the left/right wrist of $\mathcal{P}_{t-1}$.

3.5. Loss function

To train this hand detection and regression module, we optimize the combined loss function: $\mathcal{L} = \mathcal{L}_{hmo} + \mathcal{L}_{pmo} + \mathcal{L}_{RPN} + \mathcal{L}_{class} + \mathcal{L}_{reg} + \mathcal{L}^O_{class} + \mathcal{L}^O_{reg}$. Here, $\mathcal{L}_{RPN}$ is the loss for the region proposal network, $\mathcal{L}_{hmo}$ and $\mathcal{L}_{pmo}$ are flow map loss, and the other terms are for classification of bounding box or offset regression.

4. YouTube-Hand Dataset

We aim to develop a tracker that can track hands in unconstrained scenes, which may contain many people interacting with each other and the other surrounding objects. For training and evaluation, we needed a dataset of diverse conditions, but such a dataset did not exist. We therefore compiled a new dataset containing unconstrained videos and annotated them with hand locations and trajectories.

Dataset source. We name our dataset YouTube-Hand because the majority of the videos (200 out of 240) were collected from YouTube. Specifically, we scraped 200 videos from 10 scenarios, namely, casinos, concerts, cooking, dancing, driving, gyms, kids playing, mechanical workshops, sanitizing, and sports. To have a diverse dataset, we collected different videos from different YouTube uploaders. We manually verified the collected videos to ensure that they were unconstrained and diverse in terms of lighting conditions, camera perspectives, skin tones, and ages. We did not collect videos that have copyright marks. Altogether, we downloaded 200 videos from YouTube, with 20 videos for each scenario. Additionally, we selected 40 videos from the PoseTracks dataset and annotated them. The videos have spatial resolutions from 640×480 to 1920×1080 and frame rate from 24 to 30 fps.

Annotation. For each collected video, we extracted frames using the original frame rate of the video and annotated every fifteenth frame. We annotated only those hand instances whose visible areas' axis-parallel bounding box had more than 100 pixels and whose trajectory appear for more than 50 frames. Our dataset was annotated by three annotators and subsequently verified by two people.

	Total	Data source split		Train/test split	
		YouTube	PoseTrack	Train	Test
#Videos	240	200	40	150	90
#Frames	232K	227K	5K	166K	65K
#Anno. hands	60K	41K	19K	30K	30K
#Trajectories	864	666	198	519	345

Table 1. **Statistics of the proposed YouTube-Hand dataset.**

Dataset	Scene/camera Constraints	Has Video	#Hand Trajs.	Maximum #trajs/video
EgoHands [1]	Google glasses		0	n/a
Handseg [20]	Color gloves		0	n/a
NYUHands [45]	Hands keypoints		0	n/a
ColorHandPose [64]	3D hands keypoints		0	n/a
HandNet [48]	Fingertips		0	n/a
GANeratedHands [24]	Synthetic		0	n/a
Oxford-Hand [22]	Unconstrained		0	n/a
TV-Hand [26]	Unconstrained		0	n/a
COCO-Hand [26]	Unconstrained		0	n/a
Contact-Hand [27]	Unconstrained		0	n/a
100DOH [38]	Unconstrained	✓	0	n/a
GTEA [16]	Ego-centric	✓	0	n/a
WorkingHands [40]	Down-facing cam.	✓	0	n/a
BSL [31]	TV show, segmentation	✓	2	2
SynthHands [23]	Ego-centric	✓	1	1
ICP-PSO [34]	Hand keypoints	✓	6	1
EpicKitchen [7]	Ego-centric, auto-label	✓	1400	2
VIVA [36]	Vehicle-mounted	✓	45	4
YouTube-Hand	Unconstrained	✓	864	15

Table 2. **Comparing YouTube-Hand with other hand datasets.**

Figure 4. Existing hand datasets are very different from ours. This shows some representative images from: VIVA [36] (top left), EpicKitchen [7] (top right), BSL [31] (bottom left) and SynthHands [23] (bottom right).

Train/test split. We split our data into disjoint training and testing sets. The training set contains 150 videos, randomly selected from the 200 YouTube videos. The remaining 90 videos are used for testing.

Statistics and comparison with other hand datasets. Table 1 shows the statistics of our dataset. Table 2 compares our dataset with other existing hand datasets; most of them

are for hand detection only, either having no video data or hand trajectories. Some datasets contain hand trajectories, but they only have videos for constrained camera settings, such as ego-centric or in-vehicle mounted cameras. Fig. 4 shows some images from these datasets, which are much more constrained than our dataset as shown in Fig. 2.

5. Experiments

In this section, we compare our method with various generic object-tracking methods and hand tracking algorithms. We also perform ablation studies, report qualitative results, and discuss failure cases.

5.1. Implementation details and evaluation metrics

Architecture Details. We implemented HandLer using Detectron2 [53]. Specifically, we built upon a Cascade-RCNN with a DLA-34 [11] backbone with a Bi-directional Feature Pyramid Network (Bi-FPN) [44]. The network can be trained end-to-end and the inference speed is 5Hz.

Training Details. The core of HandLer is a network that takes as input two frames and outputs the linked detections across these frames. The input to the network is not necessarily a pair of consecutive frames at neither training nor testing time. To handle a wide range of video frame rates and hand movements, including low frame rate videos and fast moving hands, we actually sampled training video frames $(\mathtt{t}', \mathtt{t})$, with varying distance between $\mathtt{t}$ and $\mathtt{t}'$. Specifically for each $\mathtt{t}$, we used $\mathtt{t}' = \mathtt{t} - 15k$, for $1 \leq k \leq 5$, because the training videos are annotated every fifteenth frame.

We pre-trained HandLer using static images from the TV-Hand [26] and COCO-Hand [26] datasets by using the same static image as both frames $\mathtt{t}-1$ and $\mathtt{t}$. This was to utilize the larger datasets of annotated hands. Subsequently, we fine-tuned the network on the proposed YouTube-Hand dataset. For fine-tuning, we optimized the training loss for 12K iterations using SGD, with an initial learning rate of 0.0005 and a batch size of 48. We reduced the learning rate by a factor of 10 after 8K iterations.

Evaluation metrics. We used the standard multiple-object-tracking evaluation metrics [3, 18, 21]: the identification F1 score (IDF1), the percentage of mostly tracked trajectories (MT), mostly lost trajectories (ML), false positives (FP), false negatives (FN), identity switches (IDs), multiple object tracking precision (MOTP), multiple object tracking accuracy (**MOTA**) and higher order tracking accuracy (**HOTA**). Among these evaluation metrics, MOTA is considered the most important metric to quantify the overall detection and tracking performance. MOTA is defined as: $\text{MOTA} := 1 - \frac{\sum_t(\text{FN}_t+\text{FP}_t+\text{IDs}_t)}{\sum_t \text{GT}_t}$, where FN_t, FP_t, IDs_t, and GT_t are the number of false negatives, false positives, identity switches, and number of true hands, respectively, for the frame t.

We found that none of the commonly used MOT metrics measures the recovery ability; they do not quantify how well a tracker can right the wrong ID switch by reconnecting a new hand tracklet with a prematurely terminated one. In particular, while the ID switches (IDs) metric measures the fragmentation of a ground truth trajectory; it would apply the same penalty to any identity switch, no matter whether the tracker switches to a new-and-wrong ID or an old-but-correct ID. For example, the sequences of trajectory IDs $(a \rightarrow b \rightarrow c)$ and $(a \rightarrow b \rightarrow a)$ would have the same performance in current metrics, but the later is more desirable. Thus, we introduce a new metric called Longest-Tracklet-Ratio (**LTR**). For a particular ground truth trajectory that is matched to multiple predicted tracklets with different IDs, LTR is defined as the ratio between the length of the longest predicted tracklet and the length of the entire trajectory. We will use the average LTR on all trajectories of a test set as the new performance metric.

5.2. Main Results

Table 3 compares the performance of our hand trackers with other state-of-the-art MOT tracking methods. **TraDes**, **CenterTrack** and **FairMOT** were end-to-end trainable MOT methods, which were trained to detect and track hands jointly, but they performed relatively poor on hands, perhaps because they were geared towards less deformable and articulated classes such as pedestrians and vehicles.

We also implemented several tracking-by-detection methods, where the detection results were provided by HandCNN [26], which is the state-of-the-art hand detection method. **LightTrack** used pose tracklets to linking hands. We first used LightTrack to detect and track human body joints then associated HandCNN detected hand to a person based on the distances between the predicted wrist keypoint and the center of the detected hand bounding box. **CenterTrack*** was a method where the detection component of CenterTrack was replaced by HandCNN. **MPNTrack** was an offline tracking method, in which a Message Passing Network (MPN) was used for HandCNN detection association. For all methods, we first pre-trained using static images from TV-Hand and COCO-Hand datasets to improve the hand detection performance and then fine-tuned them using the training set of YouTube-Hand.

Based on the those metrics, HandLer outperforms the others by a wide margin. Fig. 5 shows some representative results and failure cases by HandLer.

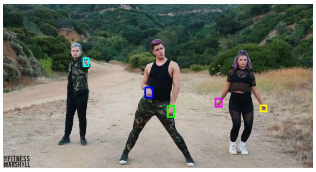
5.3. Ablation Studies

We now present our experiments to study the effectiveness of different components of the proposed architecture.

Effectiveness of HandLer. To study the importance of the

Methods	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDs↓	MOTP↑	MOTA↑	LTR↑	HOTA↑
LightTrack [29] (Pose)	53.4	101	70	6240	12816	1955	74.5	30.8	48.4	48.5
FairMot [59]	41.4	96	57	**2065**	12753	3448	76.8	39.9	31.3	39.0
MPNTrack [5] (Offline)	49.0	156	66	5918	11263	1039	77.0	40.0	44.3	40.7
CenterTrack[62]	37.2	113	62	2279	12379	3362	76.5	40.7	27.3	39.0
CenterTrack*[62]	57.8	137	43	3208	10317	1647	79.0	50.0	37.5	49.1
SORT [4]	48.3	101	72	2295	12960	1475	76.7	44.9	47.6	46.1
TraDeS [51]	53.6	168	43	3271	9102	1982	76.4	52.7	44.4	46.4
HandLer (proposed)	**70.9**	**218**	**23**	2412	**5986**	**712**	**79.9**	**70.0**	**64.3**	**59.4**

Table 3. **Hand tracking performance on the test set of YouTube-Hand.** In terms of MOTA, the most indicative MOT metric, HandLer outperforms other methods by a large margin. In each column, the best result is highlighted in **bold**, and the second best result is underlined.



(a) Tracking results by HandLer. This visualizes hand tracking results across two frames. Hands that belong to the same trajectory are visualized with the same color.

(b) Hand detection and backward regression results. The left and right images correspond to frames `t-1` and `t`, respectively. The detected hand in frame `t` and its corresponding location obtained using backward regression in frame `t-1` are shown in magenta color. The detected hands in frame `t-1` are visualized in blue and green.

(c) Comparing HandCNN and HandLer. HandCNN fails to detect blurry and occluded hands. Benefit from our temporal feature aggregation and tracking-based detection, HandLer can detect those hands.

(d) Failure cases from HandLer. The left image shows a case where a hand is not detected due to heavy occlusions, and the second images shows a case where other skin areas are mistaken for hands.

Figure 5. **Qualitative results on YouTube-Hand dataset.**

proposed forward propagation for hand tracking, we trained a model where there was no forward propagation. Similarly, we trained a model where there was no backward regression to frame `t-1`. In this case, we linked hand detections using the the Hungarian algorithm with the hand bounding boxes in frame `t`. Finally, we trained and tested the model without pose. The results are shown in Table 4. We use **HandLer** to refer to our full model, and **HandLer-NP** is HandLer without the pose. As can be seen, all those three components are important component of HandLer.

	FP↓	FN↓	IDs↓	MOTA↑	LTR↑
HandLer	2412	5986	712	70.0	64.3
HandLer w/o forward	3107	6432	761	66.1	62.1
HandLer w/o backward	2838	6195	1488	65.4	58.4
HandLer-NP	2875	6169	1256	66.1	59.0
HandLer-NP w/o forward	3076	6821	1203	63.4	56.4
HandLer-NP w/o backward	2301	6965	1536	64.4	52.2

Table 4. **Effectiveness of each components of HandLer.**

Robustness to low frame rates. We studied how the tracking performance changed as the frame rate of a video dropped. For this purpose, we ran HandLer on every K-th frame for various values of K. Specifically, we used $K = 1, 3, 5, 15$, which corresponded to 30, 10, 6, and 2 frames per seconds (fps). The results are shown in Table 5. As can be seen, the MOTA of HandLer did not decrease much when the fps was reduced from 30 to 6. Compared to SORT [4] (with HandLer detection), another tracking method described in Sec. 5.2, this level of MOTA reduction was relatively small. This demonstrates the robustness of our linking algorithm across different time gaps.

Tracking	SORT				HandLer			
Stride	FP↓	FN↓	IDs↓	MOTA↑	FP↓	FN↓	IDs↓	MOTA↑
$K = 1$	2446	36297	1902	64.9	2412	5986	712	70.0
$K = 3$	1177	2977	2903	59.2	1099	3525	1284	65.8
$K = 5$	915	2297	3301	55.6	906	2861	1468	64.3
$K = 15$	664	1636	3569	51.2	651	2077	1759	62.7

Table 5. **Performance of tracking algorithms as the frame rate of videos decreases**. K is the stride of the tracking algorithm.

5.4. Hand detection

HandLer can also be used for hand detection, as long as the input is a video. To study the effectiveness HandLer for detecting hands especially blurry and occluded hands, we sample a subset of YouTube-Hand that only contains blurry and occluded hands to test the effectiveness of HandLer for detecting such hands. Here we use the hand keypoints estimation method proposed in [41] to detect hand keypoints within every ground truth hand box. We claim

that the hand is blurry or occludes if [41] cannot detect all hand keypoints. Along with YouTube-Hand and VIVA dataset, we evaluated the performance of various hand detection methods on those three datasets using the VOC average precision metric. Since hand keypoints estimation [17**?**] cannot detect hand well for in-the-wild videos, we compared with HandCNN [26], the state-of-the-art hand detection method and summarize the results of these experiments in Table 6. Moreover, using HandLer as a detector (without linker) would also boost the tracking performance of other tracking methods, as can be seen in Table 7 for the YouTube-Hand and VIVA datasets. Note that here we only report the performances of the methods that support tracking with external detections.

	Dataset		
Method	YouTube-Hand	Blur&Occ Split	VIVA [36]
HandCNN [26]	72.4	62.8(13.1% ↓)	89.2
HandLer	**84.1**	**76.7**(8.8% ↓)	**95.3**

Table 6. **Hand detection performance**. The colored number is the percentage of performance dropped on blurry and occluded hand split comparing on the full set of YouTube-Hand dataset. Comparing with HandCNN, which runs with around 2fps, our method achieves both efficiency and effectiveness.

	IDF1↑	IDs↓	MOTA↑	LTR↑
SORT[4]	60.6(+12.3)	1902(+427)	64.9(+20.0)	53.6 (+15.1)
MPNTrack[5]	61.1(+12.1)	1288(+249)	65.2(+25.2)	57.3(+13.0)
CenterTrack[62]	61.3(+24.1)	2167(-1195)	62.7(+22.0)	51.1(+23.8)
LightTrack[29]	71.0(+17.6)	1635(-320)	61.7(+30.9)	65.7(+17.3)

Table 7. **Using HandLer as a detector with other MOT methods on YouTube-Hand dataset.** The colored number indicts performance improvement or descent comparing with Table 3.

5.5. Other datasets & tasks

We also evaluate the tracking and detection performance of HandLer on other datasets: VIVA, BSL. Note that all methods below use HandLer detection and associating detected hands with their own linker.

The VIVA dataset [36] contains frames sampled from 20 videos captured by ego-centric cameras. It was collected to develop an algorithm to detect the hands of a driver and a passenger. We used 11 videos for training and the remaining 9 for evaluation. The results are shown in Table 8.

	IDF1↑	FP↓	FN↓	IDs↓	MOTA↑
CenterTrack[62]	45.6	341	1287	79	68.7
SORT[4]	44.1	517	884	93	72.6
MPNTrack[5]	46.2	793	545	**46**	74.7
HandLer	**62.0**	**272**	**367**	58	**87.2**

Table 8. **Comparing different methods on VIVA dataset**

The British Sign Language (BSL) dataset [31] contains 6000 frames from BBC TV shows, 296 of them have been annotated with hand segmentation. All methods reported in Table 9 were trained on YouTube-Hand training set and then tested on BSL dataset.

	IDF1↑	FP↓	FN↓	IDs↓	MOTA↑
CenterTrack[62]	**22.2**	65	128	177	45.9
MPNTrack[5]	11.6	144	**76**	64	58.8
SORT[4]	13.6	92	82	71	63.2
HandLer	20.5	**60**	89	**39**	**72.5**

Table 9. **Tracking performance on the BSL dataset.**

Pose tracking. Since hands are attached to the wrists, one might wonder if we can track the human pose and the wrists instead. We hypothesize that pose tracking is a difficult problem by itself, its performance is not better than hand tracking performance. To validate this hypothesis, we perform experiment on the PoseTrack Split of Youtube-Hands. Pose tracking tracks the wrist points, but comparing point tracking results with bounding box tracking results is not trivial because MOTA computation are done differently. For a fair comparison, we consider two transformation: (1) Box2Point: represent a bounding box by its center; (2) Point2Box: match a wrist point to a detected hand by HandLer as explained in Sec. 5.2. Table 10 compares the performance of HandLer and LightTrack after making these transformations.

	Box2Point	Point2Box
LightTrack [29]	60.7	49.2
HandLer	**69.6**	**61.2**

Table 10. **Comparing with pose tracking algorithm (LightTrack) on the PoseTrack split**. The evaluation metric is MOTA. Pose tracking is difficult problem by itself, and it does not perform as well as HandLer.

6. Conclusion and Potential Negative Impacts

We introduced HandLer, a novel convolutional architecture to detect and track hands in unconstrained videos. We also collected and annotated a large-scale challenging hand tracking dataset, called YouTube-Hand. This dataset contains videos of hands in unconstrained environments, and it can be used to develop and evaluate hand tracking systems.

Hand tracking is important in various application scenarios, but there is potential for abuse of the technology to invade privacy. We will release our implementation for research purposes, but the deployment of this technology need appropriate controls to limit harmful or malicious uses.

Acknowledgements. This work started when the first author was at Stony Brook University. The work was later supported partially by DARPA PTG HR00112220001 award. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

[1] Sven Bambach, Stefan Lee, David J. Crandall, and Chen Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *Proceedings of the International Conference on Computer Vision*, 2015. 5

[2] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *Proceedings of the International Conference on Computer Vision*, 2019. 1, 2

[3] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008. 6

[4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *Proceedings of the IEEE International Conference on Image Processing*. IEEE, 2016. 2, 7, 8

[5] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 7, 8

[6] Patrick Buehler, Mark Everingham, Daniel P Huttenlocher, and Andrew Zisserman. Long term arm and hand tracking for continuous sign language tv broadcasts. In *Proceedings of the British Machine Vision Conference*, 2008. 2

[7] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. *CoRR*, abs/2006.13256, 2020. 5

[8] Xiaoming Deng, Yinda Zhang, Shuo Yang, Ping Tan, Liang Chang, Ye Yuan, and Hongan Wang. Joint hand detection and rotation estimation using cnn. *IEEE Transactions on Image Processing*, 2018. 2

[9] Eng-Jon Ong and R. Bowden. A boosted classifier tree for hand shape detection. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2004. 2

[10] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. 2017. 3

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 6

[12] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[13] Leonid Karlinsky, Michael Dinerstein, Daniel Harari, and Shimon Ullman. The chains model for detecting parts by their context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 2

[14] Mathias Kolsch and Matthew Turk. Robust hand detection. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2004.

[15] Pavan. M. Kumar, Andrew Zisserman, and Philip. H. S. Torr. Efficient discriminative learning of parts-based models. In *Proceedings of the International Conference on Computer Vision*, 2009. 2

[16] Yin Li, Zhefan Ye, and James M Rehg. Delving into egocentric actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 5

[17] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019. 8

[18] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 129(2): 548–578, 2021. 6

[19] Wenhan Luo, Junliang Xing, A. Milan, X. Zhang, Wei Liu, Xiaowei Zhao, and T. Kim. Multiple object tracking: A literature review. *arXiv: Computer Vision and Pattern Recognition*, 2014. 1

[20] Sri Raghu Malireddi, Franziska Mueller, Markus Oberweger, Abhishake Kumar Bojja, Vincent Lepetit, Christian Theobalt, and Andrea Tagliasacchi. Handseg: A dataset for hand segmentation from depth images. *ArXiv*, abs/1711.05944, 2017. 5

[21] Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. 2016. 6

[22] Arpit Mittal, Andrew Zisserman, and P. H. S. Torr. Hand detection using multiple proposals. In *Proceedings of the British Machine Vision Conference*, 2011. 2, 5

[23] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017. 5

[24] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 5

[25] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, 5(1):32–38, March 1957. 2, 4

[26] Supreeth Narasimhaswamy, Zhengwei Wei, Yang Wang, Justin Zhang, and Minh Hoai. Contextual attention for hand detection in the wild. In *Proceedings of the International Conference on Computer Vision*, 2019. 2, 5, 6, 8

[27] Supreeth Narasimhaswamy, Trung Nguyen, and Minh Hoai. Detecting hands and recognizing physical contact in the wild. In *Advances in Neural Information Processing Systems*, 2020. 5

[28] Supreeth Narasimhaswamy, Thanh Nguyen, Mingzhen Huang, and Minh Hoai. Whose hands are these? hand detection and hand-body association in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2

[29] Guanghan Ning and Heng Huang. Lighttrack: A generic

framework for online top-down human pose tracking. In *Proceedings of CVPR Workshop on Towards Human-Centric Image/Video Synthesis and the 4th Look Into Person Challenge*, 2020. 3, 4, 7, 8

[30] Jinlong Peng, Changan Wang, Fangbin Wan, Yang Wu, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *Proceedings of the European Conference on Computer Vision*, 2020. 3

[31] Tomas Pfister, James Charles, Mark Everingham, and Andrew Zisserman. Automatic and efficient long term arm and hand tracking for continuous sign language tv broadcasts. In *Proceedings of the British Machine Vision Conference*, 2012. 5, 8

[32] Pramod Kumar Pisharady, Prahlad Vadakkepat, and Ai Poh Loh. Attention based detection and recognition of hand postures against complex backgrounds. *International Journal of Computer Vision*, 2013. 2

[33] Horst Possegger, Thomas Mauthner, Peter M. Roth, and Horst Bischof. Occlusion geodesics for online multi-object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 2

[34] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 5

[35] Z. Qin and C. R. Shelton. Improving multi-target tracking via social grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 2

[36] Akshay Rangesh, Eshed Ohn-Bar, Mohan M Trivedi, et al. Driver hand localization and grasp analysis: A vision-based real-time approach. In *International Conference on Intelligent Transportation Systems*, 2016. 5, 8

[37] Kankana Roy, Aparna Mohanty, and Rajiv Ranjan Sahay. Deep learning based hand detection in cluttered environment using skin segmentation. In *International Conference on Computer Vision Workshops*, 2017. 2

[38] Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2, 5

[39] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, Daniel Freedman, Eyal Krupka, Andrew Fitzgibbon, Shahram Izadi, and Pushmeet Kohli. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of ACM Conference on Human Factors in Computing Systems*, 2015. 2

[40] Roy Shilkrot, Supreeth Narasimhaswamy, Saif Vazir, and Minh Hoai. WorkingHands: A hand-tool assembly dataset for image segmentation and activity mining. In *Proceedings of the British Machine Vision Conference*, 2019. 2, 5

[41] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 7, 8

[42] Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. Fast and robust hand tracking using detection-guided optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2015. 2

[43] Srinath Sridhar, Franziska Mueller, Michael Zollhoefer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *Proceedings of European Conference on Computer Vision*, 2016. 2

[44] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning*, 2019. 6

[45] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 2014. 5

[46] Robert Y. Wang and Jovan Popović. Real-time hand-tracking with a color glove. *ACM Transactions on Graphics*, 28(3), 2009. 2

[47] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. *arXiv preprint arXiv:1909.12605*, 2019. 1

[48] Aaron Wetzler, Ron Slossberg, and Ron Kimmel. Rule of thumb: Deep derotation for improved fingertip detection. In *Proceedings of the British Machine Vision Conference*, 2015. 5

[49] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *Proceedings of the IEEE Workshop on Applications of Computer Vision*, 2018. 1

[50] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *Proceedings of the IEEE International Conference on Image Processing*, 2017. 1

[51] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 3, 7

[52] Ying Wu, Qiong Liu, and Thomas S. Huang. An adaptive self-organizing color segmentation algorithm with application to robust real-time human hand localization. In *Proceedings of the Asian Conference on Computer Vision*, 2000. 2

[53] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 6

[54] Y. Xiang, A. Alahi, and S. Savarese. Learning to track: Online multi-object tracking by decision making. In *Proceedings of the International Conference on Computer Vision*, 2015. 2

[55] Xiaojin Zhu, Jie Yang, and A. Waibel. Segmenting hands of arbitrary color. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2000. 2

[56] B. Yang, C. Huang, and R. Nevatia. Learning affinities and dependencies for multi-target tracking using a crf model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 2

[57] Yichen Wei, Jian Sun, Xiaoou Tang, and Heung-Yeung Shum. Interactive offline tracking for color objects. In *Proceedings of the International Conference on Computer Vision*, 2007. 2

[58] Ju Hong Yoon, Chang-Ryeol Lee, Ming-Hsuan Yang, and Kuk-Jin Yoon. Online multi-object tracking via structural constraint event aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[59] Yifu Zhan, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. A simple baseline for multi-object tracking. *arXiv preprint arXiv:2004.01888*, 2020. 7

[60] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020. 2

[61] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 3, 4

[62] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *arXiv preprint arXiv:2004.01177*, 2020. 1, 2, 7, 8

[63] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, and Ming-Hsuan Yang. Online multi-object tracking with dual matching attention networks. In *Proceedings of the European Conference on Computer Vision*, 2018. 2

[64] Xiaojin Zhu, Jie Yang, and Alex Waibel. Segmenting hands of arbitrary color. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2000. 5

[65] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3