Weakly Supervised Human-Object Interaction Detection in Video via Contrastive Spatiotemporal Regions
Supplementary Material

Shuang Li[1]* Yilun Du[1] Antonio Torralba[1] Josef Sivic[2] Bryan Russell[3]
[1]MIT [2]CIIRC CTU [3]Adobe
https://shuangli-project.github.io/weakly-supervised-human-object-detection-video

In this supplementary material, we first give the model architecture details and implementation details in Section 1. Then we provide the dataset collection details in Section 2. In Section 3, we show the dataset statistics.

1. Model architecture details and implementation details

We provide the model architecture details in Section 1.1 and implementation details in Section 1.2.

1.1. Feature learning

In Section 3.2 of the main paper, we introduce the object feature, contextual frame feature, and attended human/object features. In this section, we provide more details about the different types of features.

Verb-object query feature learning. To extract the feature of the verb-object query described in the main paper (Figure 2), we first map the input verb and object queries to embedded features $\hat{e}^v$ and $\hat{e}^o$, respectively, using the publicly available Google News Word2Vec model [6]. Next, we pass the embedded features through linear mappings W_v and W_o to obtain 128-dimensional vectors $e^v = W_v\hat{e}^v$ and $e^o = W_o\hat{e}^o$.

Human feature learning. To get the human region features f^h_t described in the main paper (Figure 2), we first extract candidate human location proposals in each video frame using the publicly available DensePose model [1], which returns a binary segmentation mask of humans in the scene and human bounding-box proposals. As shown in Figure 1 of this supplement, at time t, each human region proposal i has a bounding box $b^h_{t,i}$. We pass the segmentation mask to a convolutional network to generate human feature maps and then use ROI pooling over the human bounding box $b^h_{t,i}$ to generate human region features $f^h_{t,i}$. The convolutional network consists of a 7×7 spatial convolutional layer, followed by ReLU and max-pooling nonlinearities, followed by a 3×3 spatial convolutional layer.

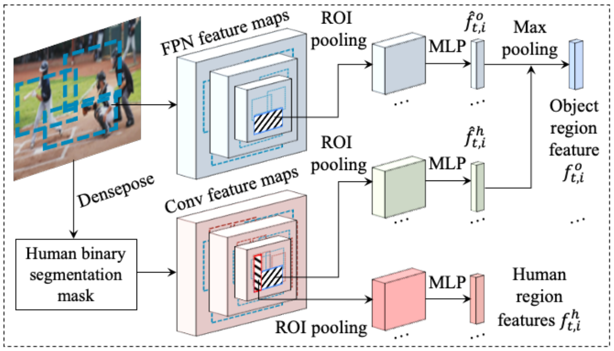

Figure 1: **Illustration of extracting human/object features.** We learn convolutional filters to encode the Densepose segmentation mask to intermediate features. We obtain the feature of each object region $f^o_{t,i}$ by combining its ROI pooled features from the FPN feature maps $\hat{f}^o_{t,i}$ and the human conv feature maps $\hat{f}^h_{t,i}$. The feature of each human region $f^h_{t,i}$ is the corresponding ROI pooled feature from the human conv feature maps. (Video credit: TheOnDeckCircle [10])

Contextual frame feature learning. We describe the contextual frame feature learning in the main paper Section 3.2. Here we illustrate the learning process of the contextual frame feature in Figure 2 of this supplement.

Human-object interactions are temporal events and occur over a period of time. To utilize the temporal information from the whole video, we use a soft attention module [11] to learn a contextual feature representation x_t for each frame. Given a video frame I_t, we send the frame to Faster R-CNN [9] and extract the final layer of the FasterR-CNN feature pyramid network to obtain an intermediate feature map. We add an average pooling layer after the intermediate feature map and generate a feature vector as the frame feature descriptor $\hat{x}_t$. Then we send $\hat{x}_t$ to an embedding layer to generate a "query" feature vector x^{que}_t. We use the same method to extract the features of all frames in the input

[2]Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague

*Work done at Adobe Research during SL's summer internship

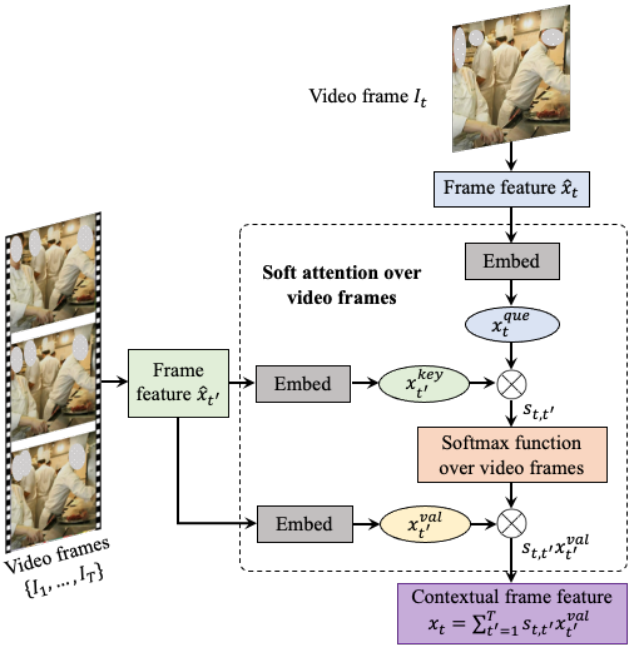
Video frame I_t
Frame feature $\hat{x}_t$
Embed
Soft attention over video frames
x_t^{que}
Frame feature $\hat{x}_{t'}$
Embed
$x_{t'}^{key}$
$s_{t,t'}$
Softmax function over video frames
Embed
$x_{t'}^{val}$
$s_{t,t'}x_{t'}^{val}$
Contextual frame feature $x_t = \sum_{t'=1}^{T} s_{t,t'}x_{t'}^{val}$
Video frames $\{I_1, \dots, I_T\}$

Figure 2: **Illustration of learning contextual frame feature.** Given a frame feature $\hat{x}_t$ obtained by passing this frame through a neural network, we send $\hat{x}_t$ to an embedding layer to generate a "query" feature vector x_t^{que}. For the feature of each frame in the same video, we use two different embedding layers to get "key" $x_{t'}^{key}$ and "value" $x_{t'}^{val}$ vectors. We compute the inner product of the "query" and "key" to get a similarity score $s_{t,t'}$ of the current frame and each frame in the same video. A softmax layer is then applied to the similarity scores to normalize the similarity of each frame to the current frame. The contextual frame feature is obtained by the weighted average over frame "value" features. (Video credit: The Best Gallery Craft [2])

video and represent them as $\{\hat{x}_1, \cdots, \hat{x}_T\}$. For the feature of each frame in the video, we use two different embedding layers to get "key" $x_{t'}^{key}$ and "value" $x_{t'}^{val}$ vectors. We compute the inner product of the "query" and "key" to get a similarity score $s_{t,t'} = (x_t^{que})^T x_{t'}^{key}$ of the current frame and each frame in the same video. A softmax layer is then applied to the similarity scores to normalize the similarity of each frame to the current frame. The contextual frame feature is obtained by the weighted average over frame "value" features $x_t = \sum_{t'=1}^{T} s_{t,t'} x_{t'}^{val}$.

Region attended human/object feature learning. To obtain the attended human and object features, Φ_t^h and Φ_t^o, used in the main paper Figure 2, we first compute an attention score for each human/object region and then aggregate the human/object features based on their attention scores. In Figure 3 of this supplement, we show the details of the region attention module used in the main paper (Figure 2). The region attention module computes attention scores for the human/object region proposals to measure their relative relevance to the given verb-object query. For each human region in frame I_t, we first concatenate its feature representation $f_{t,i}^h$ with the contextual frame feature x_t and the verb-object query feature and then pass them through a small network (consisting of two fully-connected layers with LeakyReLU as the activation function in the middle) to obtain a score. We apply the softmax function over the scores of all human regions in this frame and get the final human attention scores σ_t^h. Similarly, each object region has an object attention score σ_t^o after applying the softmax function over all object regions. The attention scores are used to aggregate human/object features using Equation 2 in the main paper.

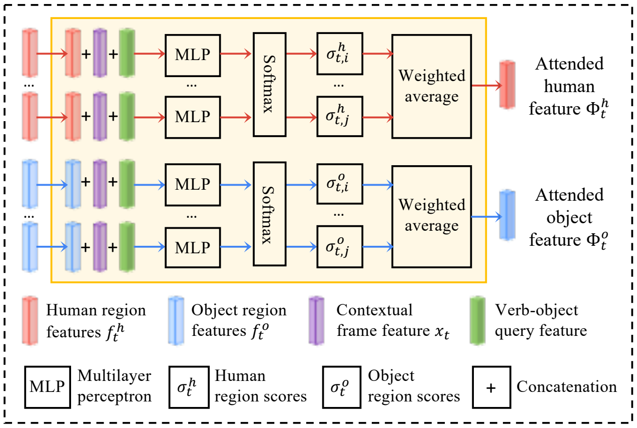
...
+
MLP
Softmax
$\sigma_{t,i}^h$
$\sigma_{t,j}^h$
Weighted average
Attended human feature Φ_t^h
$\sigma_{t,i}^o$
$\sigma_{t,j}^o$
Attended object feature Φ_t^o
Human region features f_t^h
Object region features f_t^o
Contextual frame feature x_t
Verb-object query feature
MLP Multilayer perceptron
σ_t^h Human region scores
σ_t^o Object region scores
+ Concatenation

Figure 3: **Illustration of the region attention module.** The region attention module computes attention scores for the human/object region proposals to measure their relative relevance to the given verb-object query. For each human region in frame I_t, we first concatenate its feature representation $f_{t,i}^h$ with the contextual frame feature x_t and the verb-object query feature and then pass them through an MLP to obtain a score. We apply the softmax function over the scores of all human regions in this frame and get the final human attention scores σ_t^h. Similarly, each object region has an object attention score σ_t^o after applying the softmax function over all object regions. The attention scores are used to aggregate human/object features as weights in a weighted average given by Eqn. 2 in the main paper.

1.2. Implementation details

Our model is initialized with a ResNext101 Faster R-CNN model, with the RPN pretrained on the COCO dataset from the Detectron library [3]. During training, we select 12 frames from each video and 512 object region proposals (after non-maximum suppression) as object candidate bounding boxes and 25 human bounding boxes for each frame. For the weakly supervised language-embedding alignment $\mathcal{L}_L$ loss (Equation 3 of the main paper), we compute the loss over 15 sampled negatives from $\mathcal{E}^v$ for the human term and 15 sampled negatives from $\mathcal{E}^o$ for the object term, during training.

For the self-supervised temporal contrastive loss $\mathcal{L}_T$ in each frame I_t, we compute the loss over 15 sampled negatives from the negative feature set $\mathcal{F}_t^o$. In practice, we find that the objects or humans of interest are not always present across all the frames in a video. Some video frames will only show part of the object/human or background. To make the proposed self-supervised temporal contrastive loss more robust to frames that do not contain the mentioned human-object interaction, we only use the temporal contrastive losses on 50% frames that have the lowest temporal contrastive losses in each video. The selected frames are more likely to contain the target human and objects.

We used the Adam optimizer [4] with a learning rate of 1e-4 and a learning rate of 1e-6 for the Faster R-CNN. We use a weight coefficient of $\alpha = 0.1$ for the temporal contrastive loss $\mathcal{L}_T$ in Equation 5 of the main paper.

2. Dataset collection details

We extract the videos from the Moments in Time dataset [7]. The Moments in Time dataset has 800k videos with associated metadata, such as title sentences and tags. Moreover, each video has a manually provided action label, such as "drinking" and "pushing". We leverage the action labels to help find labels for the human-object interactions from the metadata associated with the videos. We achieve this goal by initially filtering videos to contain the action label in the title sentence or metadata. However, some videos do not have verbs corresponding to human-object interactions, such as "storming" and "erupting", so we manually discard videos that do not correspond to human actions. We then used the Stanford NLP parser [5] to find videos containing noun phrases after the action label in the title or metadata, and use the resulting noun phrase as the object label. Finally, we remove videos with non-English metadata and manually filter out bad parsing results. After filtering, we obtained approximately 14,000 videos. We manually filtered out bad examples, such as videos having low frame resolution, wrong language labels, or blurry humans and objects. We finally obtained 6,594 videos in total.

We semi-automatically analyzed the natural language descriptions that accompanied the videos. We do not define a fixed list of HOIs a priori but instead use action-object pairs that appear with a certain frequency in the language captions. By considering more videos with accompanying descriptions, the vocabulary naturally increases.

We collect human and object bounding box annotations using Amazon Mechanical Turk for the test and unseen datasets. We ask each worker to annotate the specific human and object bounding boxes participating in the given human-object interaction label. For each video frame, we collect bounding box annotations from 3 different workers. We average the annotations from each worker to obtain the object bounding box annotations. We assume that there can be multiple people interacting with the given object in a video frame. To obtain the accurate number of humans in the input video frame, we want to cluster the human bounding boxes collected from different workers. The close human bounding boxes are more likely to describe the same human. By counting the number of clusters, we can estimate the number of humans in the input video frame. To do this, we ran an affinity propagation clustering algorithm [8] on all labelled human bounding boxes across multiple workers. We select the clusters which have more than two annotations and average all the annotations within each cluster as the bounding box annotation of that person. We further manually examine the annotated bounding boxes and discard low-quality annotations.

3. Dataset statistics

Our focus is on video-based, human-centric HOI detection without exhaustively annotating the spatial location of objects in a video at training which is time consuming given the large number of frames in a video. Our dataset consists of 244 different object classes and 99 different action classes. There are 756 verb-object classes in total with diverse human-object interactions.

All the videos are extracted from the Moments in Time dataset [7], which contains short trimmed videos. We semi-automatically analyzed the natural language descriptions that accompanied the videos. By considering more videos with accompanying descriptions, the vocabulary can naturally increase.

We present the dataset statistics in Figure 4, Figure 5, and Figure 6 of this supplement. Figure 4 shows the distribution of objects. We show the top 50 most frequent object classes. Figure 5 shows the distribution of the top 50 most frequent action classes. Figure 6 shows the distribution of the top 50 most frequent verb-object classes.

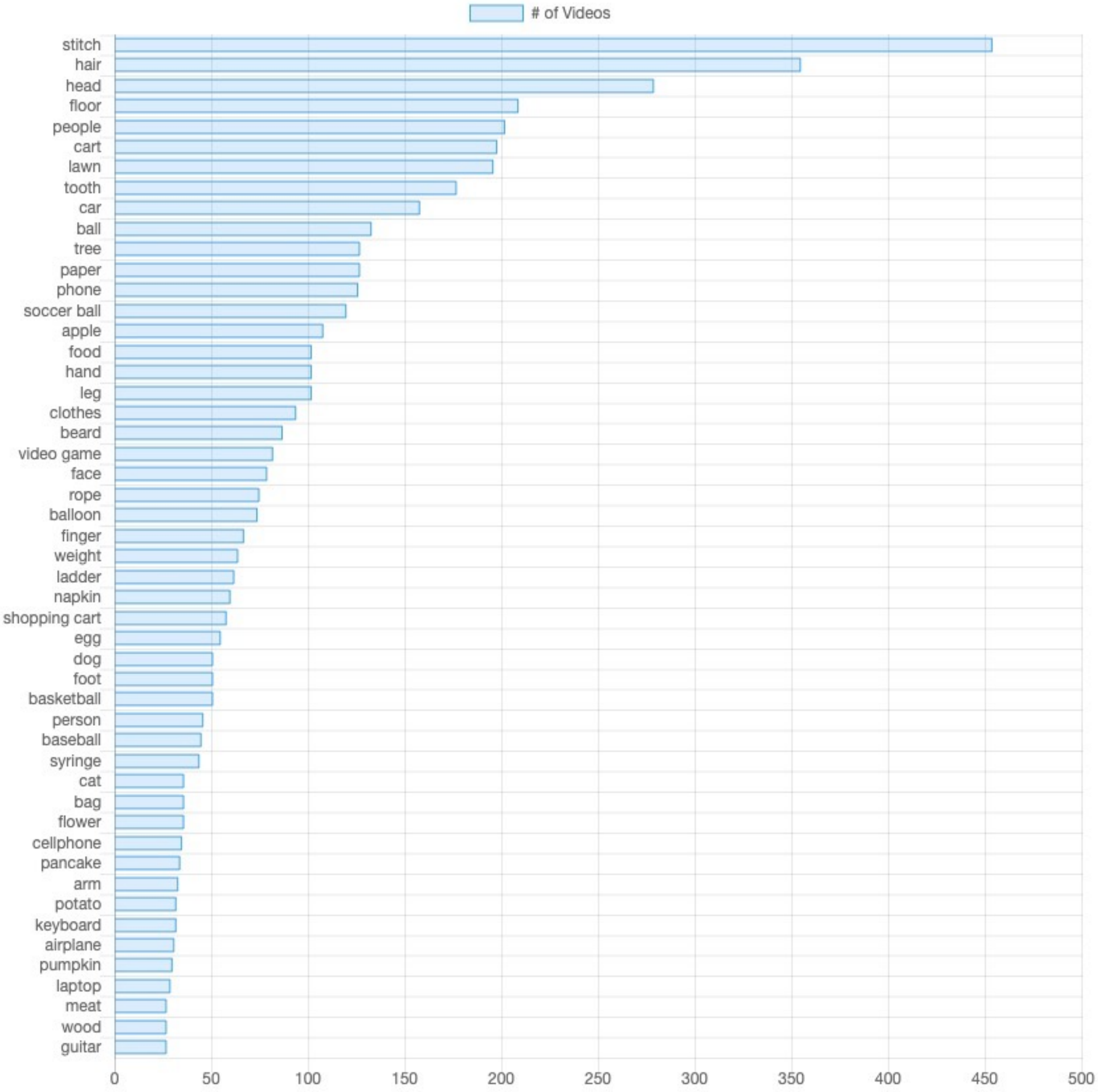

Figure 4: **Distribution of objects in our dataset.** Our dataset consists of 244 different object classes, where for brevity we only show the top 50 in the diagram above.

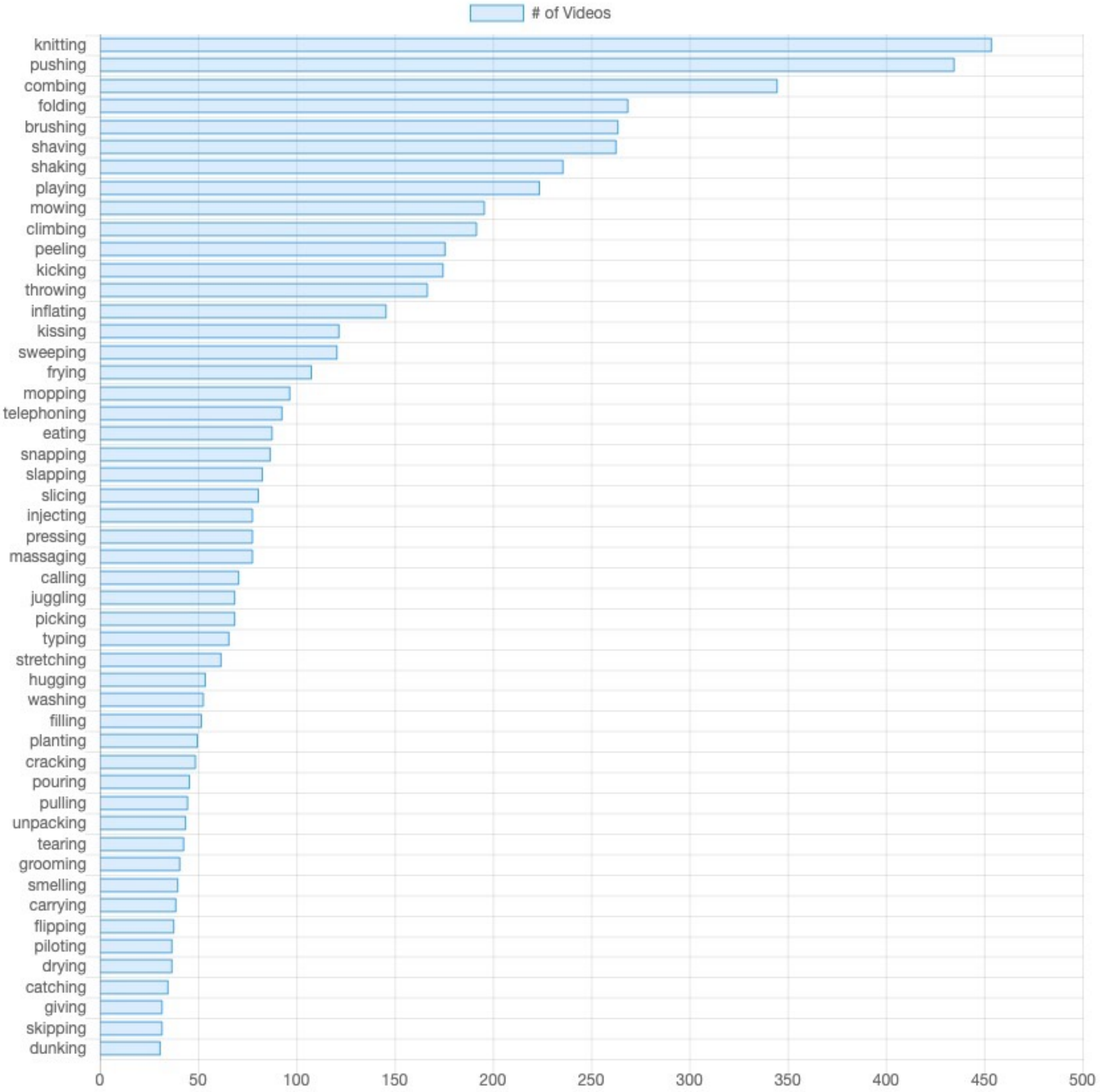

Figure 5: **Distribution of actions in our dataset.** Our dataset consists of 99 different action classes, where for brevity we only show the top 50 in the diagram above.

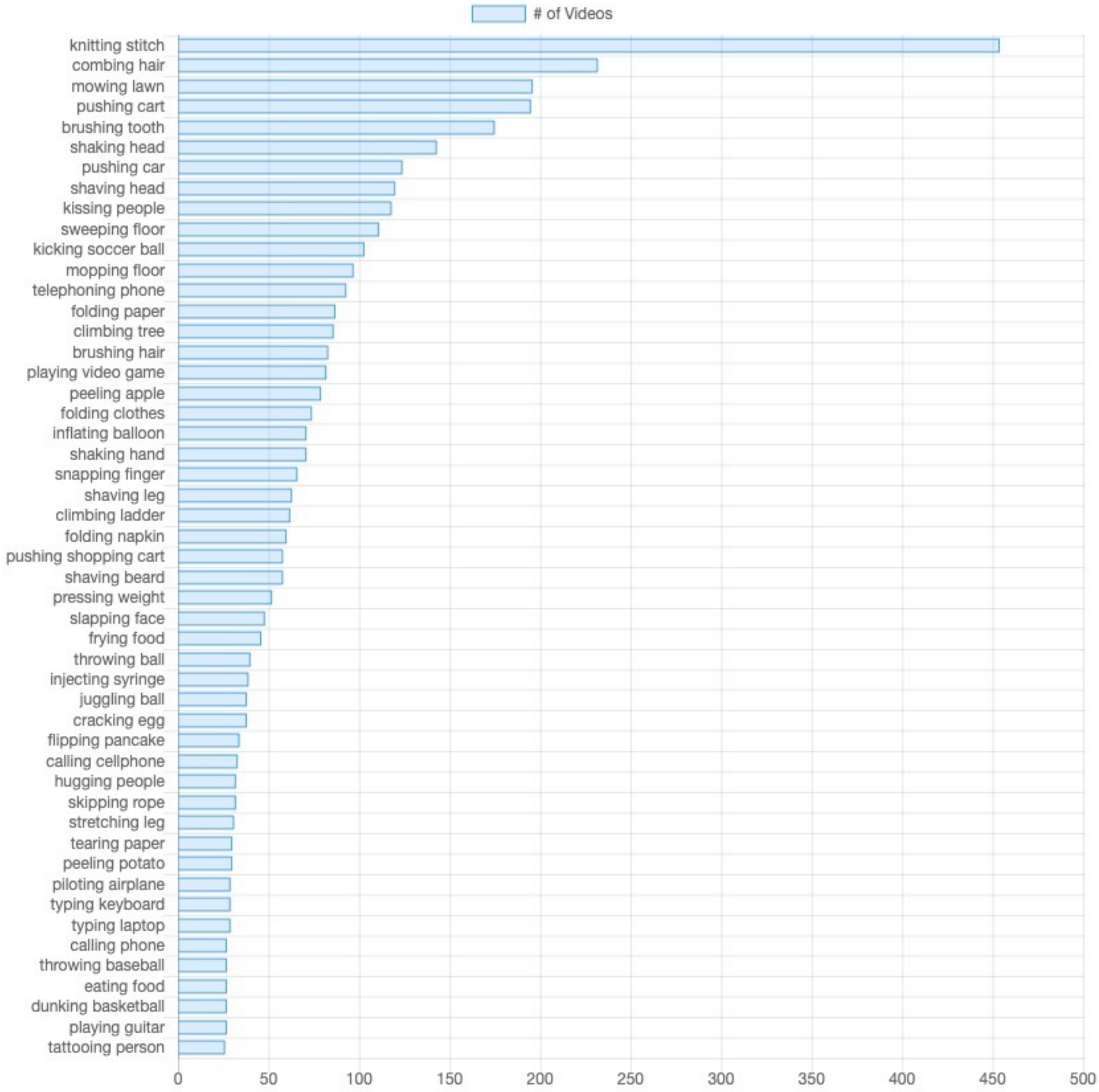

Figure 6: **Distribution of verb-object classes in our dataset.** Our dataset consists of 756 different verb-object classes, where for brevity we only show the top 50 in the diagram above.

References

[1] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018. 1

[2] The Best Gallery Craft. $https://www.youtube.com/watch?v = vo07h1vpi54$. 2018. 2

[3] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. https://github.com/facebookresearch/detectron, 2018. 2

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 3

[5] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014. 3

[6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv:1301.3781*, 2013. 1

[7] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Yan Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 3

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 3

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1

[10] TheOnDeckCircle. $https://www.youtube.com/watch?v = kgqlty_u6ok$. 2013. 1

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 1