Supplementary Material for In-Home Daily-Life Captioning Using Radio Signals

Lijie Fan*, Tianhong Li*, Yuan Yuan, and Dina Katabi

MIT CSAIL

This supplementary material provides a more detailed description of our implementation details, data collection system, and some visualization of the results in the paper. We also provide a demo video to demonstrate the model and results of our proposed RF-Diary.

Appendix A: Implementation Details

Network Architecture: For the RF skeleton generator, we use a pre-trained spatio-temporal skeleton generation model in [5]. For the skeleton feature encoder, we use an HCN network [2] pre-trained on the skeleton modality of NTU RGB-D dataset. For the video feature encoder, we use I3D network [1] pre-trained on the Charades dataset as initialization. For the RNN language generator, we use a network architecture similar to a state-of-the-art video captioning model [4]. The hidden state size is 512 for both encoder and decoder LSTMs, and the word embedding dimension is also 512. The language generator is pre-trained with the I3D network on the Charades dataset. We implement the discriminators D_m, D_n following Least Square Generative Adversarial Network [3]. D_m is with three 3D convolution layers (kernel size=3) followed by 2 fully-connected layers, and D_n is with two 1D convolution layers on time (kernel size=3) followed by 2 fully-connected layers.

Training details: At each iteration, the data loader samples an RF clip, its paired 3 videos with different viewpoints, and a video from the Charades dataset. We first use a pre-trained RF skeleton generator [5] to extract 3D human skeletons from RF signals. After that, the extracted skeletons are associated and smoothed across time using a hidden Markov model. Then the skeleton sequence first passes the HCN, and then combines with the features from corresponding floor-maps to generate $\mathbf{u}^P$.

We forward a batch of the four videos to the I3D feature extraction network and perform a discriminator on the extracted video features to give the discriminator loss. The L_2 loss is added between features from the 3 paired videos and the skeleton sequence. Since the L_2 loss has a small magnitude, we give it a 500 weight.

We then combine the features ($\mathbf{u}^P, \mathbf{v}_n^P, \mathbf{v}_n^U$) from all RGB videos and RF into one batch and forward them to the language model to get the caption loss. The loss is then combined with the feature alignment losses and backpropagated through the network. We update the discriminator 1 step for every 5 steps of the model.

To train our model, we use Adam optimizer with learning rate 4e-4, weight decay 5e-4, batch size 2 for each GPU. We train our model with 300 epochs in total, with a learning rate drop at 100 and 200 epochs (each epoch corresponds to one pass over RCD dataset).

* Indicates equal contribution. Correspondence to Tianhong Li <tianhong@mit.edu>.

All experiments are performed on 8 Nvidia Titan X Pascal GPUs. During testing, the RF-based model can run in real-time with only one GPU. All our experimental results are averaging over 5 runs.

Appendix B: Data Collection

Floormap: We use a laser distance meter like the one in the figure aside. It costs about $15 and is widely available. We use it to measure the locations and size of the objects. We always set the object's longer side to be the length and shorter side to be the width. The rotation of the object is defined as the angle between the long side of the object and the X-axis of the coordinate system.

RF: we use a radio device to collect RF signals (orange box in Figure 1). The raw RF signal from one antenna array is a complex matrix s of shape 14x128, where 14 is the number of antennas (N) and 128 is the number of evenly spaced channels between 5.4 GHz and 7.2 GHz (T). The raw RF signal is then processed using standard FMCW and antenna array equations to generate vertical and horizontal heatmaps, where the value at each pixel is given by:

$$P(x,y) = \sum_{n=1}^{N} \sum_{t=1}^{T} s_{n,t} e^{j \cdot 2\pi \frac{d_n(x,y)}{\lambda_t}},$$

where d_n denotes the round trip distance from the transmit antenna to the point at (x, y), and back to the k-th receive antenna. The transmission power of our device is less than one millie Watt, similar to a commercial WiFi router.

Camera System: We design a wireless camera system to collect multi-view videos across the home to provide ground truth caption (blue boxes in Figure 1). We use 12 Raspberry Pi 3 single-board computers to control each camera and a laptop to control the Pis. The remote Raspberry Pis are synchronized with the camera system controller. Our radio and cameras are synchronized using network time protocol (NTP), whose synchronization error is typically less than 10ms. The original video is of size 1232x1640 and is resized and center-cropped to 224x224 to feed into the I3D network.

Appendix C: Feature Alignment Details

To align the features from video and RF, we need to make them consistent in both the temporal dimension and the feature dimension. Since the frame-rate of heatmaps from RF is 30 FPS, while the frame-rate of the video frame is 15 FPS. We add one down-sample layer in RF skeleton generator so the extracted skeletons are in 15 FPS. We further modified the HCN architecture to have similar padding and pooling strategy as the I3D network so the extracted features from both modalities have the same size in the temporal dimension.

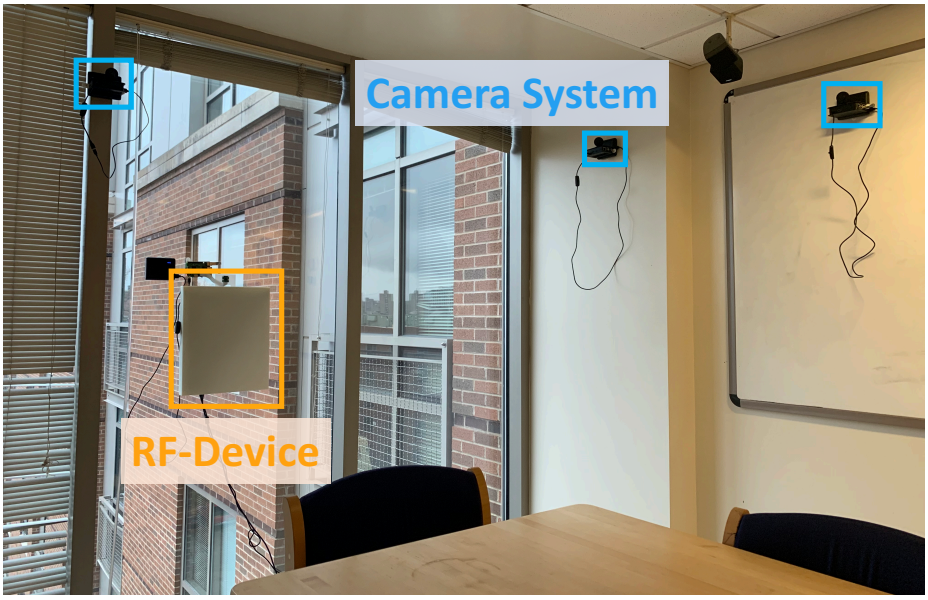

Fig. 1. The figure shows our data collection system. Orange box is our radio device to collect RF signals. Blue boxes show the camera system to collect video frames across the home.

Since we use a person-centric representation for the floormaps, we will get a different floormap representation at each different time step. The person-centric floormap representations are generated using the measured floormap and the intermediate skeletons, which are in 15 FPS. We further downsample the temporal dimension of floormaps using bilinear interpolation so that the features from floormaps and skeletons can be combined to get features **u**. Feature dimension of **u** should be consistent with the video feature **v** generated by I3D (i.e. 1024), so we require the embedding layer for **u** with the same output size of 1024. In this way, both the temporal dimension and the feature dimension is consistent between **u** and **v** so that we can perform feature alignment over them.

Appendix D: Feature Alignment Visualization

To further illustrate the effectiveness of our feature alignment framework, we perform visualization on the feature spaces we learned. In our feature alignment framework, there are in total three feature spaces: features extracted from RF+floormap on RCD, features extracted from videos on RCD, and features extracted from videos on the Charades dataset. We visualize these three distributions using t-SNE to see the difference between them.

As shown in Figure 2, without feature alignment (left), the two feature distributions extracted from the video-captioning network are close to each other while far away from the feature distribution of the RF-Diary. After applying the L_2-norm (middle), the features extracted from RF signals are aligned with the features extracted from the video on RCD, while the features extracted from Charades dataset are decoupled with those

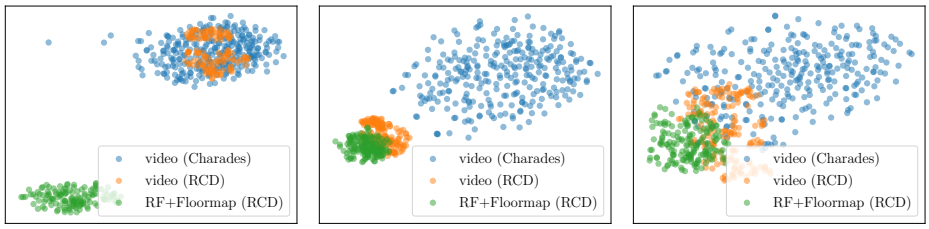

Fig. 2. Distribution of features extracted by RF-Diary and video-caption model. Left: without feature alignment. Middle: with L_2-norm, Right: with L_2-norm and discriminator.

from RCD because only the features extracted from RCD are affected by the L_2-norm. In the third figure, we show that this decoupling can be solved by adding a discriminator between the two feature distributions from paired and unpaired videos.

References

1. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
2. Li, C., Zhong, Q., Xie, D., Pu, S.: Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. pp. 786–792 (2018)
3. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2794–2802 (2017)
4. Wang, X., Chen, W., Wu, J., Wang, Y.F., Yang Wang, W.: Video captioning via hierarchical reinforcement learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4213–4222 (2018)
5. Zhao, M., Tian, Y., Zhao, H., Alsheikh, M.A., Li, T., Hristov, R., Kabelac, Z., Katabi, D., Torralba, A.: Rf-based 3d skeletons. In: Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. pp. 267–281. ACM (2018)