Pair Selection for View Invariance in Siamese Network

Yu-Cheng Lin
UC Davis
ycjlin@ucdavis.edu

Wei-Chih Chen
UC Davis
wccchen@ucdavis.edu

Abstract

First-person cameras are getting prevalent for sport enthusiasts, but the amount is still quite few. At the same time, convolutional neural networks (CNNs) made a big step forward since the occurrence of Alexnet. To take advantage of the existing numerous third-person videos on Internet and the success of CNNs, the transfer learning with the help of the Siamese network is a promising way to raise the video classification accuracy of first-person videos, given limited first-person video training dataset.

To find the mapping between different views, we use the dataset that has synchronized videos in first-person view and third-person views. However, when the pair amount becomes larger, the training time takes longer. Therefore, in this work, we try to find the best strategy to do pair selection for view invariance training in Siamese network. From our experiment results, we show that pair selection should focus on hard negatives and clarify positive boundary by compensation. In addition, choosing an appropriate feature is very important to gain better results for view invariance training in Siamese network for multiple view videos.

1. Introduction

As the first-person cameras like GoPro become popular for some sports hobbyists and geeks, there would be more first-person videos in the near future. Most of the videos online, such as Youtube, are shot in third person views. However, the research of video action recognition on first-person videos is still in the beginning. Given that there are so many videos on the Internet, which are mostly third-person point-of-view, we are going to leverage the abundance of these third-person videos to train and recognize first-person videos.

The idea is inspired by using transfer learning [8][9] to do a target task better based on the previous experience of other source tasks. In order to achieve the transfer learning for first-person video classification, we want to find a feature space which has view invariance property—that is to have similar representations for first-person and third-person videos.

2. Related Work

Convolutional neural networks (CNNs) have got great improvement on visual recognition since AlexNet [1]. In addition to applying CNNs for image recognition, detection, segmentation and retrieval, some papers use CNNs to combine image information across videos [2][3].

On the other hand, there are still few papers talking about the relationships and recognition about the first-person and the third-person videos. The only paper that is kind of similar to our work is focusing on the recognition of each video for its action class and viewpoint [10]. The NUS First-person Interaction dataset for action recognition comprises 8 actions in 2 perspectives (first-person and third-person perspectives). There is no relationship mapping between the videos. Hence there are 16 classes in total. And all videos were shot at different instants of time as only one camera was used. On the contrary, our dataset is time-synchronized and shot with one egocentric GoPro camera and two third-person video cameras. We would like to use video data that is time-synchronized and leverage this relationship between the first-person and the third-person videos to find the best options to train a feature embedding with view invariance.

To learn an invariant mapping, Hadsell et al. proposed the Siamese network architecture using a contrastive loss function to learn the parameters that pull the similar neighbors together and push apart the dissimilar objects [6].

For the application of Siamese network, DeepFace [13] got a success in face verification by learning a deep feature representation. In advance, Lin et al. [11] used it for a more challenge task: matching street view and aerial view imagery.

3. Method

3.1. Architecture

To learn similar and dissimilar pair, we adopt Siamese network to learn these pairs. In Siamese network, it based on exist two neuron networks to fetch feature vectors of two input images, and try to close distance between two feature vectors if they are similar and to separate two vectors as far as possible if they are dissimilar. For training process, we use Alexnet to fetch feature vector and share weights of these two Alexnet shown in Figure 1. Since feature weights of these architectures are different from original Alenet, we train from scratch on full connected layers which are fc6 and fc7. To normalize variance mean of fc7, we put an extra fc7_mvn layer after fc7. After training process, we get any one of two fine-tined networks as base and extract features to visualize it by t-sne. We use t-sne results to analyze which kind of similar or dissimilar pair is more suitable that shown in section 4. After loss converging to certain small number, we stop training process for fine-tine. And this architecture is our purposed new network for video recognition.

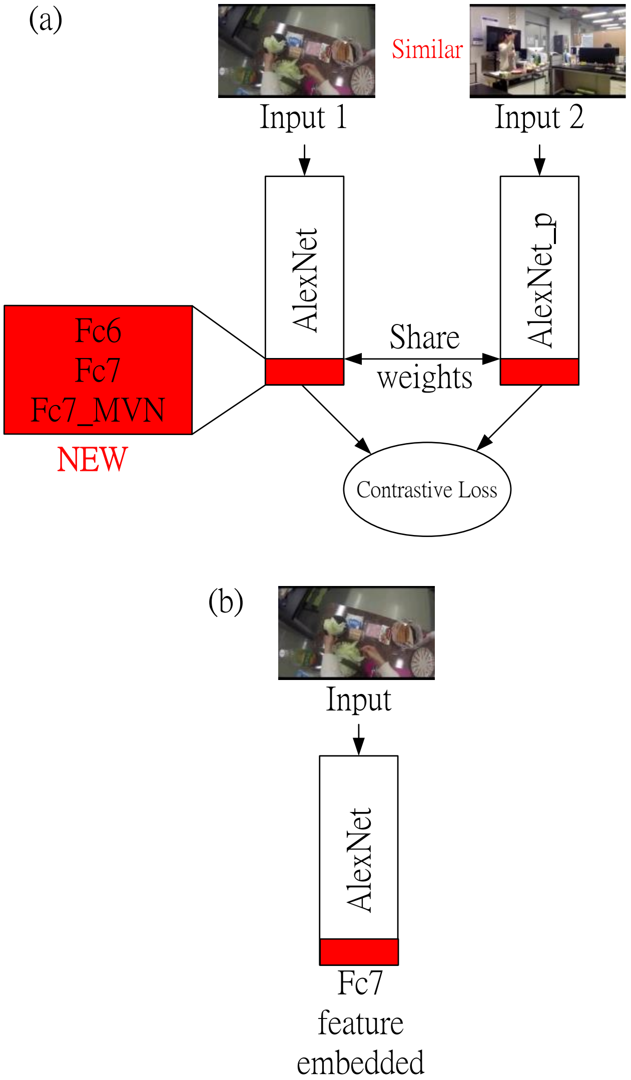


Figure 1. (a) Proposed architecture for training based on Siamese Network. (b) After the training process, we use the trained network to fetch the feature embedding of the fc7 layer.

3.2. Siamese Network

The use of the Siamese network is to learn an invariant mapping between a pair of images [6]. This invariant mapping is achieved by letting the similar pairs have small Euclidean Distance in the feature space [11]. The Contrastive Loss Function proposed in [6] takes this as a training target: not only pulling the similar pairs altogether, but also pushing apart the dissimilar pairs at least the margin distance m. The loss function is shown below:

$$L(x,y,l) = \frac{1}{2} l D^2 + \frac{1}{2}(1-l) max^2(0, m - \|D\|_1)$$

where $l \in \{0,1\}$ is the similarity label. The label l is equal to 1 for similar pairs; l is equal to 0 for dissimilar pairs. The margin distance m > 0 is used for the component of dissimilar pair loss calculation. $\|D\|_1 = \|f(x) - f(y)\|_1$ is the Euclidean Distance between the two input image pairs f(x) and f(y) in the feature space. The square of the max function makes it a differentiable function, which is more appropriate for back propagation in training.

3.3. Visualization

To evaluate the performance of the view invariance feature embedding, we need to reduce the high dimensional data to low dimensional data to do analysis on different options. T-Distributed Stochastic Neighbor Embedding (t-SNE) is a technique to visualize high dimensional data by dimension reduction. Therefore, we use t-SNE library [7] to visualize the feature representation for further discovering the effects of different pair selection options. Specifically, we use t-SNE to reduce the 4096-dimension normalized fc7 feature vectors to 2-dimension x-y coordinates.

4. Experiments

4.1. Dataset

This dataset comprises 8 different labels for videos and each scene shot in three different views which have two still cameras in different positions and one gropro. These videos in one scene are all time-synchronized. We got 191 video in total that shot in 63 different scenes and all of them are provided by a professor in Indiana University. In Table 1 shows how many videos shot in each category.

Since input of classifiers is an image, we use frames of video as input. But a video has more than four thousand frames. We sample 1/30 of total frames in a video for reducing training time. And within all of the dataset, we random sample three fourth of data as training set and one fourth of data as testing set.

Label	Class	Video number
0	Coffee	18
1	Kidnap	56
2	Milk	3
3	Office	59
4	Pizza	15
5	Presentation	18
6	Sandwich	13
7	Typing	9
	Total	**191**

Table 1. List all videos shot in different category for data set.

4.2. Setup

Since the computation speed of our Siamese network is around 6 million pairs per day, to save experiment time, we focus on the category “sandwich” to find the general principles of pair selection for Siamese network. The six input videos are sandwich1_gopro, sandwich1_mac1, sandwich2_gopro, sandwich2_mac1, sandwich3_gopro, and sandwich3_mac1. To reduce computation amount, we pick one frame out of 30 frames, which equals 1 frame/sec for our 30fps videos.

Similar pair definition. To achieve view invariance in Siamese network training, the videos of the same environment setting from different view cameras are conceptually set as similar (e.g., sandwich1_gopro and sandwich1_mac1 are similar).

However, even in the same set videos, all the frames in different timeline should not be set as similar. Therefore, we make all frames within ± 2 seconds in the different view videos as similar pairs. Since we pick 1 frame out of 30 frames every second, for a specific frame $F_g(t)$ in gopro, the corresponding frames in mac1: $F_m(t-2)$, $F_m(t-1)$, $F_m(t)$, $F_m(t+1)$, and $F_m(t+2)$ are all defined as similar, and vice versa. There are 3953 similar pairs in total.

Experimental options. There are four experimental options in our experiments. The options (A), (B), and (C) are for dissimilar pairs, and the option (D) is for similar pairs. The detailed descriptions are followed.

Option (A): same video, different view pair. For a set of multiple view videos, the frames which are within ± 2 seconds in the different view videos are assumed similar; the frames which are out of ± 10 seconds in the different view videos are assumed dissimilar. Since we pick one frame out of thirty frames every second, for the frame $F_g(t)$ in gopro, the frames in mac1 $F_m(t-10)$, $F_m(t-9)$, ..., $F_m(t)$,..., $F_m(t+9)$, $F_m(t+10)$ cannot be chosen as dissimilar pairs, and vice versa. We choose 25 times dissimilar pairs for each input frame. There are 39650 dissimilar pairs.

Option (B): different video pair. All video frames come from different video sets are assumed as dissimilar pairs. For example, all the frames in sandwich1_gopro and sandwich2_gopro or sandwich2_mac1 can be dissimilar pairs. There are totally 4*Combinational(n, 2) = 4*3 = 12 combinations for each input video. We choose 12 times dissimilar pairs for each input frame from all the combinations for each input video. There are 39936 dissimilar pairs.

Option (C): same video, same view, different time pair. In the very same video which is recorded in the same view, the frames which are shot after a long time period should be assumed as dissimilar pairs. The frames which are out of ± 10 seconds in the same video are assumed dissimilar. Since we pick one frame out of thirty frames every second, for the frame $F_g(t)$ in gopro, the frames in gopro $F_g(t-10)$, $F_g(t-9)$, ..., $F_g(t)$, ..., $F_g(t+9)$, $F_g(t+10)$ cannot be chosen as dissimilar pairs; for the frame $F_m(t)$ in mac1, the frames in mac1 $F_m(t-10)$, $F_m(t-9)$, ..., $F_m(t)$, ..., $F_m(t+9)$, $F_m(t+10)$ cannot be chosen as dissimilar pairs. We choose 20 times dissimilar pairs for each input frame. There are 31720 dissimilar pairs.

Experiment	Similar pairs	Dissimilar pairs	Total	Pair Ratio(sim:dis)
1. (A)(C)	3953	39650+31720 =71370	75323	1:18.05
2. (A)(C)(D)	3953+3160=7113	39650+31720 =71370	78483	1:10.03
3. (A)(B)(C)	3953	39650+39936+31720 =111306	115259	1:28.16
4. (A)(B)(C)(D)	3953+3160=7113	39650+39936+31720 =111306	118419	1:15.65

Table 2: Four experiment settings and their corresponding pair numbers.

Option (D): similar pair compensation. In option (C), we set the frames which are out of ±10 seconds in the same video as dissimilar pairs. However, these frames have mostly the same scenes. To avoid overemphasize the dissimilarity of these frames, we compensate this by adding all frames within ± 2 seconds in the same view video as similar pairs. Since we pick one frame out of thirty frames every second, for the frame $F_g(t)$ in gopro, the frames in gopro $F_g(t-2)$, $F_g(t-1)$, $F_g(t)$, $F_g(t+1)$, $F_g(t+2)$ are all taken as similar; for the frame $F_m(t)$ in mac1, the frames in mac1 $F_m(t-2)$, $F_m(t-1)$, $F_m(t)$, $F_m(t+1)$, $F_m(t+2)$ are all taken as similar. For the implementation, we just pick the similar pairs (F(t-2), F(t)), (F(t-1), F(t)), (F(t+1), F(t)), and (F(t+2), F(t)), for all t%2==0. There are 3160 similar pairs.

We set up four experiments to identify how to choose pairs efficiently for view invariance training in Siamese network. All four experiments have the common 3953 similar pairs and the total pair numbers are decided by their respective pair selection option settings. The detailed settings and pair numbers are shown in Table 2.

4.3. Results

The experiment results are shown in Figure 3, Figure 4 and Figure 5, which represent Sandwich1 video, Sandwich2 video and Sandwich 3 video respectively. To do a fair comparison between these four experiments, we try to make the epochs in the same value range in spite of their different pair sizes. Therefore, we choose the t-sne visualization results of experiment 1, 2, 3, and 4 with iterations equal to 27,500, 30,000, 40,000, and 40,000 correspondingly, which is equivalent to 93.46, 97.86, 88.84, and 86.47 epochs. To see how the trained embedding performs for the view mapping, we test the trained network with the input training data for Sandwich1, Sandwich2, and Sandwich3 videos. The dot of "o" means the input data frame belongs to the first-person videos, which is taken from gopro; The dot of "+" means the input data frame belongs to the third-person videos, which is taken from the still cameras, mac1 or mac2. The color encoding is done according to the time sequence of the frames in a video. The starting frames are marked as the blue color; the ending frames are marked as the red color. The ideal result is that the same colored dots from the first-person and third-person videos can overlap together, as shown in Figure 2. From the experimental results we can see that it is kind of hard to achieve. Hence, our judge principle is that the more similar the dot distribution of different view frames, the better is the performance.

From the experimental results shown in Figure 3, since the gopro dots are stretched longer and more consistent with the mac1 dots, the experiment two setting gets the best result for Sandwich 1 videos; in Figure 4 the experiment two setting also gets the best result for Sandwich 2 videos; in Figure 5 of Sandwich 3 videos, both the experiment one and two settings get the better results than the other settings.

In short, from these four experiments, the experiment two setting is overall better. Second, the option (B) makes distinct dots grouping together, instead of stretching apart. Third, similar pair compensation is crucial. This suggests that pair selection should focus on hard negatives and clarify positive boundary by compensation.

The training loss over iteration is shown in Figure 6. We can see that the loss does not decrease very smoothly, which means that the learning process does not work efficiently. There are two ways to improve this: finding a better feature extractor and refining the training architecture and hyper-parameters.

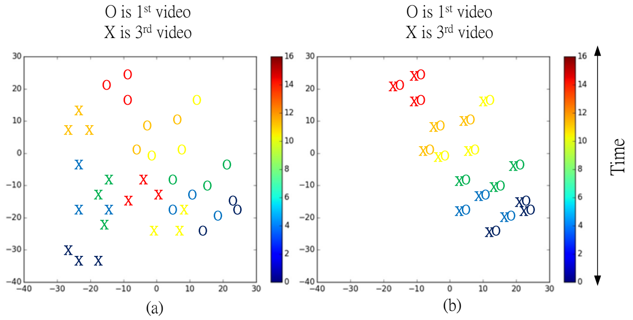


Figure 2. The ideal result is that the same colored dots from the first-person and third-person videos can overlap together. (a) is feature without mapping (b) is ideally feature with mapping.

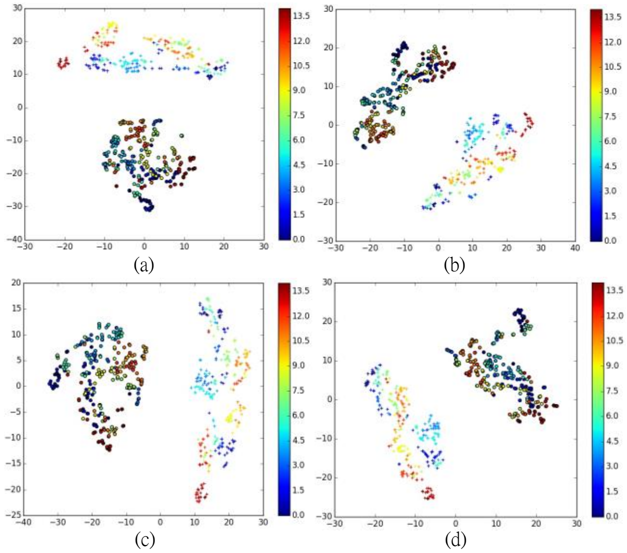
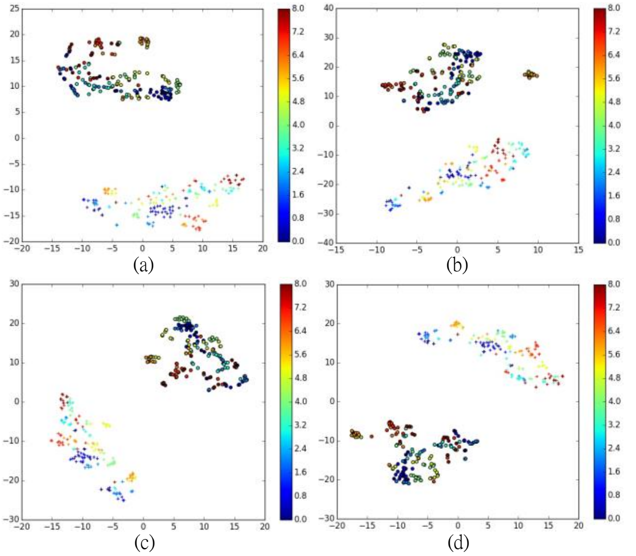


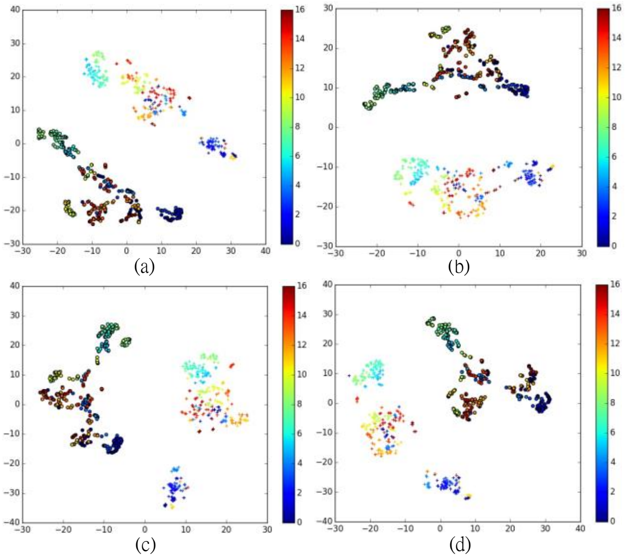
Figure 3. In sandwich1 video, (a), (b), (c) and (d) are t-sne visualization results of experiment 1, 2, 3, and 4 with iterations equal to 27,500, 30,000, 40,000, and 40,000

correspondingly, which is equivalent to 93.46, 97.86, 88.84, and 86.47 epochs. Legends: “o”: gopro, “+”: mac1 and Color: Blue: start frames, Red: end frames


(a) (b)

(c) (d)

Figure 4. In sandwich2 video, (a), (b), (c) and (d) are t-sne visualization results of experiment 1, 2, 3, and 4.


(a) (b)

(c) (d)

Figure 5. In sandwich3 video, (a), (b), (c) and (d) are t-sne visualization results of experiment 1, 2, 3, and 4.

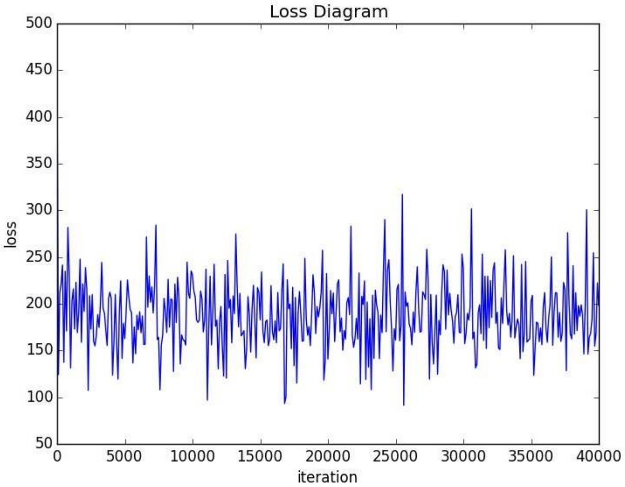


Figure 6. The training loss over iteration.

4.4. Data Analysis

We try to analyze what data we used to learn in Siamese network. Figure 7 is sandwich 1 video shot in three different views which are time synchronized. We chose meaningful content frames to represent this scene and marked 1 to 12 to stand for time sequence. Total length of each video is 328 seconds and it was separated into three main contents, such as making sandwich, preparing to eat, and eating sandwich.

By our definition of similar pairs, (a)-5 is similar with (b)-5 and other 4 frames within ± 2 seconds which should be almost the same as (b)-5 in general. And for dissimilar pairs in option (A), (a)-5 is dissimilar with frame before (b)-4 and frame after (b)-6 which are out of ± 10 seconds of video (b). However, these dissimilar frames are looks pretty similar as (b)-5 which defined as similar. This is a contradiction definition. Besides that, other dissimilar definitions will also get some similar contradiction results. This explains why our loss value does not monotonic decrease over iterations.

5. Conclusion

Feature is the key. From our four experiments, we find a better strategy to select pair options for view invariance training in Siamese network. However, since the image features extracted by the Alexnet pre-trained model is not good enough, the training loss can not be reduced monotonously. We think the main problem of Alexnet model is that it cannot extract the time-domain information. Hence our future work would be taking another feature extractor, such as C3D [12], to take advantage of the salient motion information in the video context and doing the same experiments again to see if we could get better view invariance mapping in Siamese network.

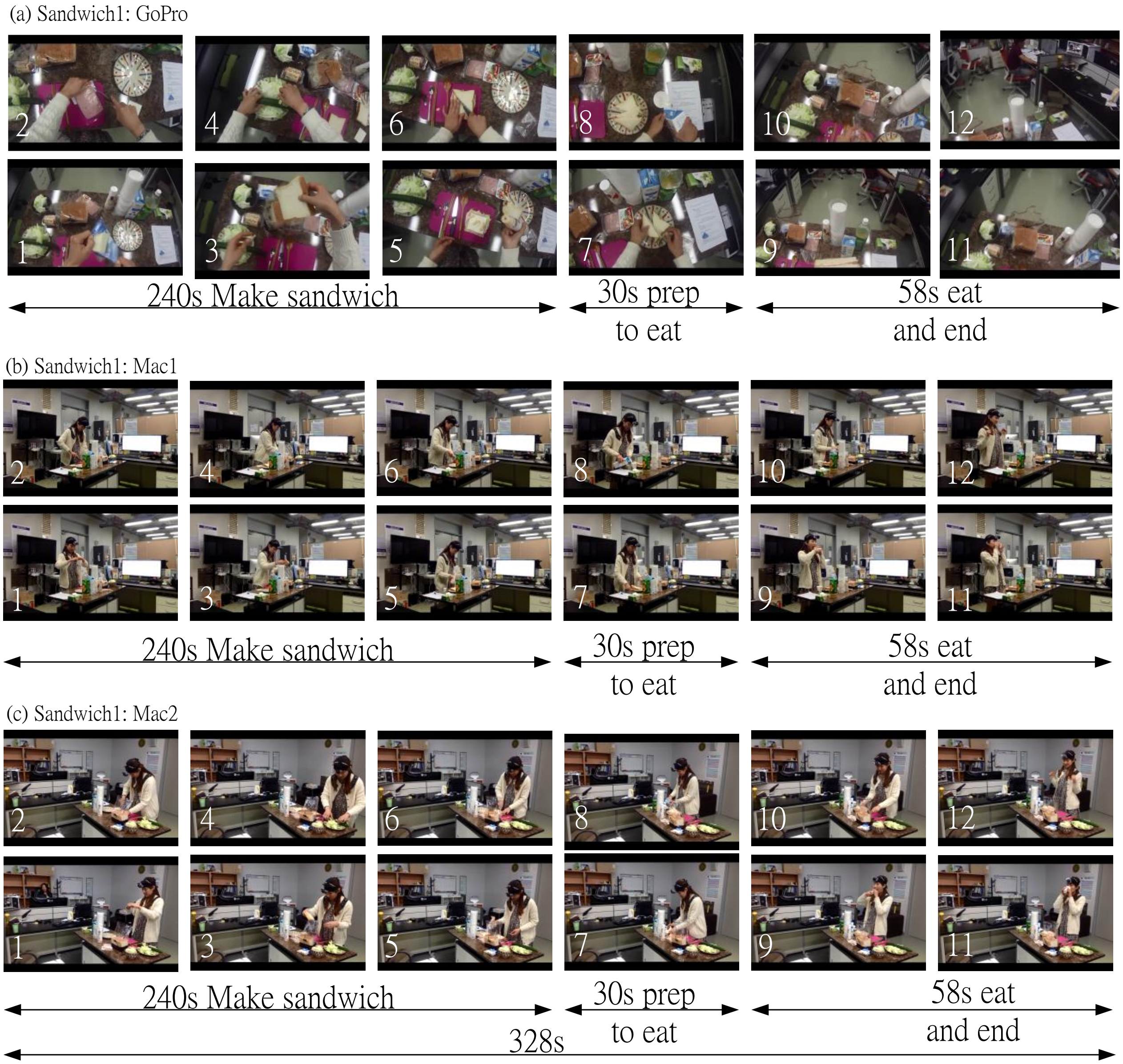


Figure 7. Three videos shot for sandwich 1 in three different views. We chose meaningful content frames to represent this scene and marked 1 to 12 to stand for time sequence. (a), (b) and (c) represents for gopro, mac1 and mac2 video respectively.

References

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[2] Ng, J. Y. H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. arXiv preprint arXiv:1503.08909.

[3] Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In Advances in Neural Information Processing Systems (pp. 568-576).

[4] Narayan, S., Kankanhalli, M. S., & Ramakrishnan, K. R. (2014, June). Action and Interaction Recognition in First-person videos. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on (pp. 526-532). IEEE.

[5] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... & Darrell, T. (2014, November). Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the ACM International Conference on Multimedia (pp. 675-678). ACM.

[6] Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In Computer vision and pattern recognition, 2006 IEEE computer society conference on (Vol. 2, pp. 1735-1742). IEEE.

[7] Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9(2579-2605), 85.

[8] Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. Knowledge and Data Engineering, IEEE Transactions on, 22(10), 1345-1359.

[9] Torrey, L., & Shavlik, J. (2009). Transfer learning. Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques, 1, 242.

[10] Narayan, S., Kankanhalli, M. S., & Ramakrishnan, K. R. (2014, June). Action and Interaction Recognition in First-person videos. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on* (pp. 526-532). IEEE.

[11] Lin, T. Y., Cui, Y., Belongie, S., Hays, J., & Tech, C. (2015). Learning Deep Representations for Ground-to-Aerial Geolocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5007-5015).

[12] Tran, Du, et al. "Learning spatiotemporal features with 3d convolutional networks." *ArXiv e-prints* (2015).

[13] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In CVPR, 2014. 1, 3, 5