Embodied One-Shot Video Recognition: Learning from Actions of a Virtual Embodied Agent

Yuqian Fu[1*], Chengrong Wang[1*], Yanwei Fu[2]
Yu-Xiong Wang[3], Cong Bai[4], Xiangyang Xue[1], Yu-Gang Jiang[1#]
Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University[1]
School of Data Science, Fudan University[2] Robotics Institute, Carnegie Mellon University[3]
Zhejiang University of Technology[4]
{yqfu18, 19212010024, yanweifu, xyxue, ygj}@fudan.edu.cn, yuxiongw@cs.cmu.edu, congbai@zjut.edu.cn

ABSTRACT

One-shot learning aims to recognize novel target classes from few examples by transferring knowledge from source classes, under a general assumption that the source and target classes are semantically related but not exactly the same. Based on this assumption, recent work has focused on image-based one-shot learning, while little work has addressed video-based one-shot learning. One of the challenges lies in that it is difficult to maintain the disjoint-class assumption for videos, since video clips of target classes may potentially appear in the videos of source classes. To address this issue, we introduce a novel setting, termed as *embodied agents based one-shot learning*, which leverages synthetic videos produced in a virtual environment to understand realistic videos of target classes. In this setting, we further propose two types of learning tasks: embodied one-shot video *domain adaptation* and embodied one-shot video *transfer recognition*. These tasks serve as a testbed for evaluating video related one-shot learning tasks. In addition, we propose a general video *segment augmentation* method, which significantly facilitates a variety of one-shot learning tasks. Experimental results validate the soundness of our setting and learning tasks, and also show the effectiveness of our augmentation approach to video recognition in the small-sample size regime.

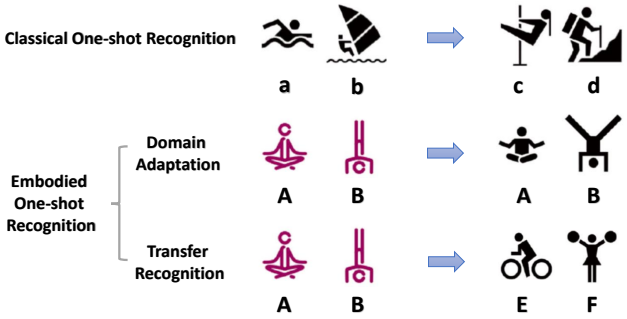

Figure 1: Comparison between the classical one-shot video recognition setting and our novel embodied one-shot recognition setting. Black symbols denote the real video data, purple symbols denote the virtual video data synthesized in our virtual embodied environment. The first row represents the classical one-shot setting. The second row represents the one-shot video domain adaptation task and the third row represents the one-shot video transfer recognition task. (best viewed in color)

CCS CONCEPTS

• **Computing methodologies** → **Activity recognition and understanding**; *Intelligent agents*; Transfer learning.

KEYWORDS

One-shot Learning; Video Action Recognition; Embodied Agents

ACM Reference Format:
Yuqian Fu[1*], Chengrong Wang[1*], Yanwei Fu[2] and Yu-Xiong Wang[3], Cong Bai[4], Xiangyang Xue[1], Yu-Gang Jiang[1#]. 2019. Embodied One-Shot Video Recognition: Learning from Actions of a Virtual Embodied Agent. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3351015

* indicates equal contributions, # indicates corresponding author.

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3351015

1 INTRODUCTION

Deep learning has achieved great success in many multimedia applications, including image object detection [17, 30, 41] and captioning [6, 50, 55]. Due to the powerful learning ability, deep architectures have been also extended to tackle tasks in more complex video domains, such as video classification [18, 21, 26, 47]. However, a large amount of manually labeled data is required to train these models and this may not be realistic in real-world multimedia applications. Therefore, one-shot learning [12, 15, 40, 44, 49, 52, 53], which aims to enable models to recognize a novel unseen concept with only one or few examples, has attracted increasing attention. In the widely used one-shot learning setting, we are given a source domain and a target domain, and any data in the target domain should not be contained in the source domain. All the labeled data in the source domain can be used to help train a model to recognize novel classes of the target domain.

Most of the existing work focuses on image-based one-shot learning [7, 11, 12, 15, 49, 51]. By contrast, videos consist of temporal sequences of frames, increasing the difficulty and complexity in

learning representations and one-shot classifiers. To address this issue, previous work has explored metric learning [16, 24, 33] and meta learning [58].

One-Shot Learning Setting Revisited. In the ideal one-shot learning setting, the target data of novel classes are supposed to be strictly disjoint from the source data of known base classes [49]. However, videos are more complex than images, and it is common that a video labeled as a certain action category contains some video clips from other actions. For example, the videos of the "shooting basketball" class are likely to contain the clips of the "running" class. Thus, if we take the classes of "shooting basketball" and "running" as the source and target domains, respectively, some videos of the "running" class may appear in the source data but with different class labels. This indicates that the source and target classes are *not disjoint* in such scenarios.

Violating the disjoint-class assumption may result in some undesired effects in one-shot video recognition. First, the feature representation may not be necessarily learned to generalize to videos of novel classes, since some videos have already existed in the source domain. Second, it is difficult to analyze and evaluate the transfer and generalization ability of proposed algorithms for one-shot learning tasks, since the improved performance might be contributed to that some target videos have been seen by the model.

To address these limitations when extending one-shot image recognition to video domains, we advocate learning from actions of *a virtual embodied agent*, which is inspired by recent work on embodied agents [1]. An embodied agent is an intelligent agent that interacts with the environment through its body. In our video recognition problem, we define the environment as scenarios where humans perform actions, and the goal of the agent is to mimic human actions as real as possible. This imitation process is loosely relevant to how humans recognize a novel action. To better understand one particular novel action, humans might play that action in the brain. Incorporating this ability to hallucinate video instances of new actions in a virtual world might help machine vision systems perform better one-shot learning. More importantly, by leveraging the purity of synthetic videos, we mitigate the aforementioned overlapping issue between source and target data.

Formally, we introduce a novel one-shot learning setting — *embodied agents based one-shot learning for video recognition*. In our setting, source data is the synthetic animations from the embodied environment, and we take real videos as the target data. Concretely, we propose two tasks — embodied one-shot video *domain adaptation* and embodied one-shot video *transfer recognition*. The action classes of source and target domains are the same in the former task, but different in the latter task.

A comparison between the typical one-shot setting and our new setting is shown in Figure 1. One merit of our setting is that it allows to synthesize massive virtual videos effectively and efficiently. The virtual world is mainly composed of an agent and an environment. The agent performs specific actions repeatedly but with different poses, changing background scenes, and various camera parameters. This simulation runs automatically and is implemented based on a popular game engine called Unreal Engine 4. We conduct a pilot study in the new setting and by running our simulator, we construct a new dataset, termed as *UnrealAction*, which contains 14 action classes and each class has 100 virtual videos.

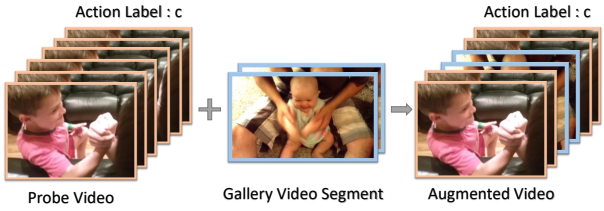

Figure 2: Illustration of our video segment augmentation method. Given a probe video with label c, we replace one segment in it with another gallery video segment to generate a new video whose label can still be regarded as c.

In addition to the new setting and benchmark, we introduce a novel video *segment augmentation* method that leverages the virtual videos for one-shot video learning. Inspired by the subliminal advertising experiment [22, 34] in advertising industry, we augment videos by replacing some short clips. This introduces some small turbulence in learning to extract video features. In psychological science, it is also known as Subliminal Perception [39]. Specifically, we first collect some videos from the source domain as gallery videos, and then we divide these gallery videos into consecutive video segments. Given a labeled probe video, we calculate the similarity between probe video segments and gallery video segments. As demonstrated in Figure 2, by replacing one segment in a probe video with the corresponding gallery video segment, we generate a new video of the same label as the original probe video. In this way, we are able to augment video instances on a large scale. One-shot video recognition tasks are thus conducted over these augmented videos. Extensive experiments on the UnrealAction and MiniKinetics datasets validate our new setting and learning tasks, and show the effectiveness of our augmentation approach in one-shot learning tasks.

Contributions. We summarize our contributions as follows. 1) For the first time, the task of embodied agents based one-shot video learning is proposed. We introduce a novel learning setting — the Unreal environment, with a set of action scripts, virtual avatars, and the corresponding evaluation protocol. The *UnrealAction*[1] dataset is publicly available. 2) We propose two novel tasks in this setting — embodied one-shot video domain adaption and embodied one-shot video transfer recognition, as an extension of one-shot learning to video domains. 3) We further propose a novel video segment augmentation method to address one-shot video learning. Extensive experiments show the soundness of our learning setting and the effectiveness of our learning algorithm.

2 RELATED WORK

One-Shot Learning. Previous work mainly focuses on metric-learning, meta-learning, and generative models for one-shot learning in image domains. Flagship techniques in metric-learning methods include Deep Siamese Network [25], ProtoNet [44], and Matching Net [49]. Meta-learning methods [15, 40, 52–54] train a meta-learner to optimize the parameters of recognition models. And

[1] http://www.sdspeople.fudan.edu.cn/fuyanwei/dataset/UnrealAction/

generative models are complementary to these discriminative approaches [12, 27, 32, 51]. By contrast, we address one-shot learning in video domains, a more challenging and under-explored task. The most relevant work is Compound Memory Network (CMN) [58]. While CMN improves the network architecture for one-shot video recognition, our method focuses on data augmentation.

Video Representation Learning. The core of video action recognition is video representation learning. Conventional approaches focus on hand-crafted representations [23, 28, 29]. Most of them detect spatio-temporal interest points and then describe these points with local representation. More recently, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have shown high performance in video representation learning. Among them, we can briefly group the model-based methods into two categories: image-based (mainly rely on 2D ConvNets) [9, 14, 43, 57] and video-based (mainly rely on 3D ConvNets) [2, 20, 37, 48]. And they are not mutually exclusive. Some image-based two-stream networks apply 3D ConvNets to fuse the spatial and flow streams [13]; some video-based networks use 2D convolutional layers to reduce the size of models [2, 37].

Although they have achieved very high accuracy on existing video recognition benchmarks, these methods are still not able to recognize a novel class with only one or few examples. In this paper, we mainly address one-shot learning tasks by enabling our model to quickly learn new concepts. In addition, data augmentation, such as random sampling, random cropping, flipping, rotation, and semantic augmentation [8], has been used in learning video representation. We propose a frame-level video segment augmentation method, which replaces one segment in a probe video with the most semantically similar gallery video segment. The generated video thus maintains the semantic information and temporal consistency of the original probe video.

Learning from Virtual Data. Many attempts have been made to train a network using data rendered from 3D models, such as GTAV [42], flying chairs [10], UnrealCV [36], and DeepDrive [38]. We build a virtual world which is designed for generating videos of human actions with various camera viewpoints, 3D models of characters, agent poses, and environment. Virtual video action recognition datasets are relatively rare and this is the first virtual action recognition dataset to the best of our knowledge.

Embodied Agents. The concept of embodied agents is widely used. The work of navigation learning [1] is relatively similar to ours. They establish a benchmark which consists of scenarios from environment datasets, such as SUNCG [45] and Matterport3D [3]. However, they mainly focus on indoor navigation, while we are interested in human actions in different scenarios, including indoors, city street, and natural scenes. The goal of a navigation agent is to navigate to a location specified by either a coordinate, or a category of areas, or a category of objects, while our agent aims to perform multiple actions in different scenarios with a camera tracking it.

TRECVID Multimedia Event Detection (MED). Event detection has been studied in TRECVID MED[2], which learns to assign event labels to videos [4, 5, 19, 35, 56]. It also has the one-shot and zero-shot settings. Note that in MED, only testing videos are provided. That is, researchers can use any available source data to pre-train the model. Different from TRECVID MED, our newly proposed setting has a fixed source domain. Hence, it is relatively easier to evaluate and compare the transfer ability of different one-shot video recognition models with the fixed virtual source videos.

3 TASK FORMULATION

We simulate the actions of virtual avatars in our virtual environment. The videos are then generated based on the embodied agents.

3.1 Actions of Virtual Embodied Agents

Actions from Embodied Agents. The video game industry has developed many tools to facilitate building realistic virtual world. We take advantage of this and choose a popular game engine, Unreal Engine 4, to build our simulator. Unreal Engine 4 provides *Blueprint*, a visual script, to control the virtual world. Our simulator is mainly composed of an agent and an environment, and the actions of the agent are recorded by virtual cameras. We use BluePrint to define the activities of the agent, the motion of the virtual camera, and the reaction of the environment. We collect tens of character models as alternative appearances of our agent, 14 action classes of animations as our classification categories, and several game maps as our environment. The 3D character models are in different clothes, hairstyles, genders, and races. They also have skeletons, which enable the characters to perform skeletal animations. We make these animations compatible with all of our character skeletons. Therefore, all the characters can perform all the actions. The game maps include indoor scenes, urban scenes, and natural scenes. Many of these resources can be easily accessed in the Unreal store[3], which is a market providing game developers with needed resources. All of these resources are connected by the BluePrint script.

Action Video Synthesis. The virtual environment allows us to synthesize as many action videos as needed. The virtual videos are recorded by the camera in the simulator. We implement a camera which moves to keep track of the agent all the time in BluePrint. To enrich the diversity of the generated data, our agent changes its appearance by using different 3D character models and moving from one place to another in the virtual world. The agent keeps performing all the 14 actions continuously. We define the time for which the agent performs all these 14 actions in sequence as a period. The agent finds a new place in the map automatically at the beginning of each period and starts performing at that place with a random pose. The camera follows the agent and appears at a random place near the agent. This process is repeated, thus generating different video clips every time. In this way, we synthesize a large number of virtual videos with high diversity. These synthetic videos provide additional information to our recognition model.

Embodied Video Recognition. To facilitate the study of embodied agents based one-shot learning, we release a novel dataset – UnrealAction, which has 14 action classes, and each action class has 100 videos from the virtual domain and 10 videos from the real-world domain. The virtual videos consist of the actions of the agent captured by virtual cameras in our environment. The real-world videos are collected from social media platforms and published datasets, such as YouTube, UCF101 [46], and Kinetics [2].

[2]https://www-nlpir.nist.gov/projects/tv2018/Tasks/instance-search/

[3]https://www.unrealengine.com/marketplace/en-US/store

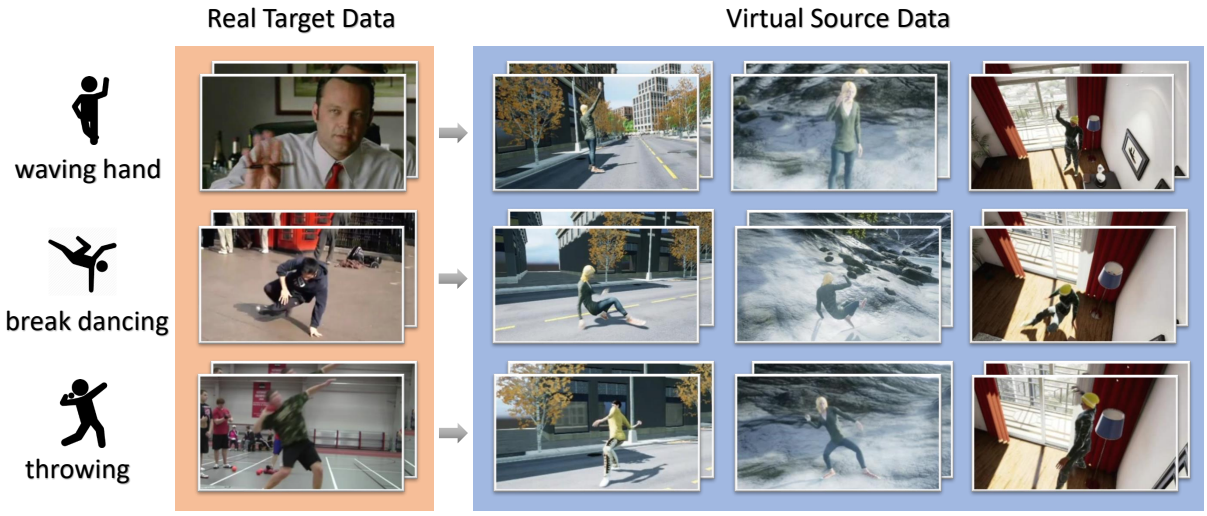

Figure 3: Examples of our UnrealAction dataset. The left part shows the real target videos, while the right part demonstrates the synthetic virtual videos. We generate videos with various camera viewpoints, 3D character models, agent poses, and scenes.

Essentially, our dataset performs as the playground for two tasks defined here, namely, embodied one-shot domain adaptation and embodied one-shot transfer recognition. These tasks use the virtual video classes as the source domain to help classification of real-world videos. Note that the action classes of the source and target domains are the same in the former task, but different in the latter task, as shown in Figure 1.

3.2 Embodied One-Shot Video Recognition

Problem Setup. The basic experimental setup is as follows. Given a base category set C_{base} and a novel category set C_{novel}, we have a base dataset $\mathcal{D}_{base} = \{(\mathrm{V}_i, z_i), z_i \in C_{base}\}$ and a novel dataset $\mathcal{D}_{novel} = \{(\mathrm{V}_i, z_i), z_i \in C_{novel}\}$. A recognition algorithm is learned on $\mathcal{D}_{base}$ and aims to generalize to the novel category which has one or few labeled examples.

Specifically, we present two tasks of embodied one-shot video recognition: (1) Embodied one-shot video domain adaption. For this task, the category sets of the source domain and the target domain remain the same, i.e., $C_{base} = C_{novel}$. We aim to learn a one-shot classifier on the virtual source data and then generalize it to real target data. (2) Embodied one-shot video transfer recognition. This task is consistent with traditional one-shot learning, where the category set of the source data is disjoint from that of the target data, i.e., $C_{base} \cap C_{novel} = \emptyset$. What distinguishes us from classic video one-shot recognition is that our source data is virtual videos, thus making our task more challenging.

Video Representation. Generally, there are image-based and video-based representation learning methods. (1) For image-based representation learning methods, they simply average the image-level features as video-level features. One advantage of image-based based models is that they can leverage ImageNet pre-trained architectures for warm starting, which is very useful. (2) For video-based representation learning methods, they require a significant amount of video instances and classes to help train models [2]. And in many cases, the pre-trained dataset may still contain the videos in the target domain. Thus, we stick to image-based representation learning methods in our setting, and we leave video-based representation learning methods as future work. We use $f_\theta(.)$ to denote the video representation extractor, where θ is the parameter set.

Evaluation Setup. We extend the typical *N-way-k-shot* setting [40] to evaluate the performance on the novel tasks. Specifically, to evaluate the capability of recognizing novel categories, we sample an *N-way-k-shot* episode from $\mathcal{D}_{novel}$ for testing, repeat this process for a certain number of times, and then we average the results. An *N-way-k-shot* task is derived by the following procedure: we first randomly sample N classes from C_{novel}, and then randomly sample k labeled samples per class to construct the support set S ($|S| = N \times k$). An additional unlabeled example q is sampled. This example belongs to one of the N classes and is used for testing.

One-shot Classifier. We have multiple choices of one-shot classifiers, including SVM, KNN, and ProtoNet [44]. We compare the performance of these classifiers in the ablation study, and we adopt ProtoNet as our one-shot classifier. ProtoNet is a metric-learning method and uses Euclidean distance for measuring distance between video representations. For support set S in the testing episode, S is augmented to $\tilde{S}$ by our segment augmentation method. Following [44], we then calculate the prototype vector p_c for each class c in $\tilde{S}$ as follows:

$$p_c = \frac{1}{|\tilde{\mathrm{S}}_\mathrm{c}|} \sum_{(\mathrm{V_i}, \mathrm{z_i}) \in \tilde{\mathrm{S}}_c} f_\theta(\mathrm{V_i}). \tag{1}$$

Given the query video q, its probability of belonging to class c can be computed as:

$$P(z_q = c \mid q) = \frac{\exp(\|f_\theta(\mathrm{q}), p_c\|)}{\sum_{j=1}^{N} \exp(\|f_\theta(\mathrm{q}), p_j\|)}, \tag{2}$$

where $\|\cdot\|$ indicates the Euclidean distance. The class label with the highest probability is the predicted label for the query video q.

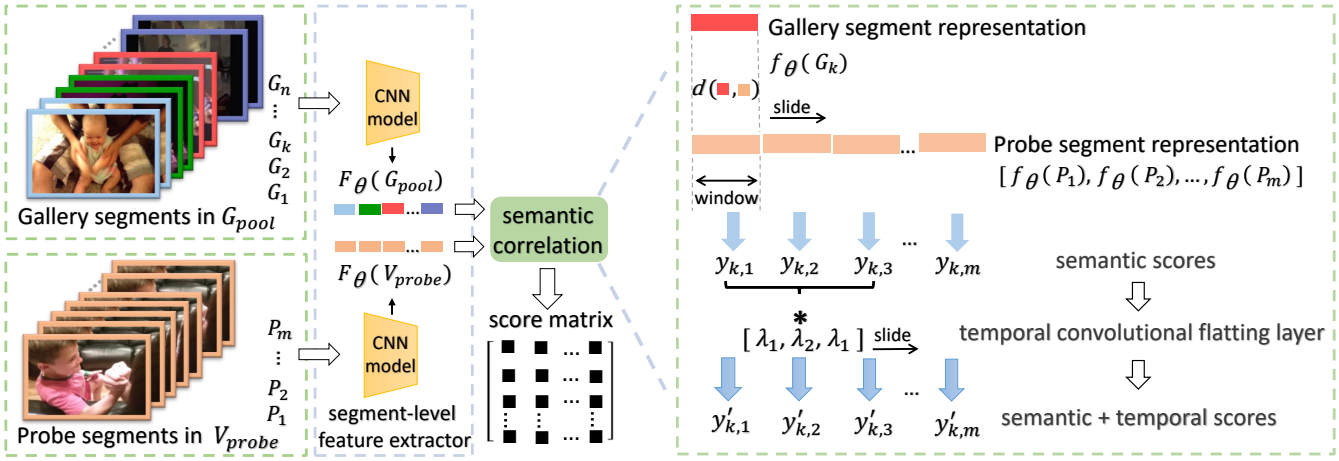

Figure 4: Illustration of our video segment augmentation method. l_{seg} is set as 2. Both gallery segments and probe video segments are fed into a CNN model to get segment-level features, and $F_\theta(.)$ indicates the sequence of segment features. For each gallery segment and probe video segments, the semantic correlation is calculated as shown in right side. A sliding window is used to swipe gallery video segment over probe segments with stride=1 (one segment). For each window, we calculate semantic distance between gallery and probe segments to obtain "semantic scores" and then a convolution operation is applied to semantic scores which is denoted as "temporal convolutional flatting layer" to get the final scores. The gallery segment with the smallest score will be chosen to replace the corresponding probe video segment.

4 RECOGNITION METHOD

4.1 Video Representation Learning

In all of our experiments, we use ResNet-50 as our video feature extractor, and the features before the final layer of each frame are averaged as video-level (segment-level) representation.

To learn a feature extractor, our method mainly contains two stages. We denote the dataset augmented by our algorithm as $\tilde{\mathcal{D}}_{base}$ or $\tilde{\mathcal{D}}_{novel}$. We first fine-tune our video feature extractor on $\mathcal{D}_{base}$ and then fine-tune the network on the augmented dataset $\tilde{\mathcal{D}}_{base}$ generated by our segment augmentation method. When first fine-tuning the network on $\mathcal{D}_{base}$, we expect that our model transfers from the ImageNet domain to our source domain, which we believe is much closer to the target domain in general. And the purpose of fine-tuning our model on $\tilde{\mathcal{D}}_{base}$ is to enforce our model to have the ability of recognizing the augmented videos.

In the testing phase, the fine-tuned model is used as a feature extractor for $\mathcal{D}_{novel}$ and $\tilde{\mathcal{D}}_{novel}$. Then ProtoNet is applied to predict action labels for query videos.

4.2 Video Segment Augmentation Method

We introduce a novel frame-level video segment augmentation method. We randomly select 10 videos per class from the base dataset $\mathcal{D}_{base}$. We use these selected videos to form a gallery set $\mathcal{G}$. More concretely, the centering 16 frames of each video are sampled as $\mathcal{G}$. The same $\mathcal{G}$ is used for augmenting both the base dataset $\mathcal{D}_{base}$ and the novel dataset $\mathcal{D}_{novel}$.

Formally, given a probe video V_{probe} and its corresponding action label z_{probe}, we divide it into several continuous video segments of length l_{seg}. This means that each segment consists of l_{seg} frames. We also perform the same partition for all the gallery videos. Hence, we have $|C_{base}| \times 10 \times 16/l_{seg}$ gallery segments to form the gallery segment pool $\mathcal{G}_{pool}$. By replacing a probe segment in V_{probe} with a gallery segment in $\mathcal{G}_{pool}$ each time, we can generate a new video V_{syn}, whose action category is still labeled as z_{probe} when l_{seg} is small.

Our video segment augmentation method is demonstrated in Figure 4. First, both the gallery and probe video segments are fed into $f_\theta(.)$ to obtain segment-level features. As shown in Figure 4, $F_\theta(.)$ indicates the sequence of segment features. Specifically, the features of segments in V_{probe} are formulated as

$$F_\theta(V_{probe}) = [f_\theta(P_1), f_\theta(P_2), ..., f_\theta(P_m)], \tag{3}$$

where m indicates the number of segments in V_{probe}, and P_m represents the m-th segment belonging to V_{probe}. After that, for each segment $\mathcal{G}_k$ in $\mathcal{G}_{pool}$, we calculate the semantic correction between $f_\theta(\mathcal{G}_k)$ and $F_\theta(V_{probe})$, and the details are shown in Figure 4. Specifically, given the representation of a gallery segment (the red one) and probe segments (the orange one), we compare the Euclidean distance between the segments of videos. This is achieved by applying a sliding window over the representations of probe segments as in Figure 4, and computing Euclidean distance of representations between gallery and probe segments. The computed results $(y_{k,1}, y_{k,2}, y_{k,3}, ..., y_{k,m})$ reflect the similarity between these segments.

We not only seek the semantically closest gallery video segment for a certain probe segment, but also take temporal consistency into consideration. Specially, we apply a convolution operation to the semantic scores with a fixed symmetric kernel template – $[\lambda_1, \lambda_2, \lambda_1]$. We denote it as a "temporal convolutional flatting layer". The final score vector $(y'_{k,1}, y'_{k,2}, y'_{k,3}, ..., y'_{k,m})$ thus helps maintain the temporal consistency in the generated videos.

We compute the score matrix between all segments of each probe video and all gallery segments. As illustrated in Figure 4, each probe video segment is replaced by the segment from gallery videos with the smallest score. This generates a new video V_{syn}. In the testing phase, in order to maximize the generation of new data, we replace each video segment and synthesize a new video for it. For example, for a clip of 16 frames with $l_{seg} = 2$, our method generates 8 augmented videos.

To train the model on the base dataset, we replace one segment clip every 16 frames, so that every 16-frames video clip has exactly one segment replaced after being randomly cropped from the original training video.

5 EXPERIMENTS

5.1 Datasets

UnrealAction. The details of this new dataset are described in section 3.1. We conduct experiments on the two novel tasks on this dataset: one is embodied one-shot video adaptation which uses virtual videos of the 14 classes as source data and real videos of the same 14 classes as testing data, the other one is embodied one-shot transfer recognition, which uses the virtual data of 14 classes as source data and real videos of other classes as testing data. In the latter setting, the testing set of MiniKinetics is leveraged as the target data.

MiniKinetics. Due to the lack of general benchmarks in one-shot video action recognition, we follow the dataset processing method proposed in [58], and we denote it as MiniKinetics. All the videos in MiniKinetics are collected from the Kinetics dataset [2], and 100 classes are selected from the original Kinetics, with 100 videos per class. The 100 classes are divided into 64, 12, 24 classes for training, validation and testing, respectively. There is no intersection between the categories of these three datasets. To better evaluate our video segment augmentation method, we also conduct experiments on MiniKinetics and achieve the state-of-art performance.

5.2 Implementation Details

Video Processing. At the training stage, we randomly sample continuous 16-frames clip for both $\mathcal{D}_{base}$ or $\tilde{\mathcal{D}}_{base}$, and each frame is randomly horizontally flipped for data augmentation. Following the processing procedure in CMN [58], the frames are first rescaled by resizing the shorter side to 256 and then random cropped to a 224×224 region. At the testing stage, we sample the center continuous 16-frames clip and then adopt a center crop to obtain a 224×224 region per frame.

Setup. Stochastic gradient descent (SGD) with momentum=0.9 is used to fine-tune our network for 6 epochs for both of two fine-tuning stages. The batch size is set as 6. When fine-tuning on the initial train dataset, the learning rates of the last layer and the other layers are set to 1×10^{-2} and 1×10^{-3}, respectively. They are divided by 10 when fine-tuning on the augmented training dataset to prevent our network from overfitting. The kernel λ_1, λ_2 in the temporal convolutional flatting layer is set as 0.1 and 1.0, respectively, and l_{seg} is set as 2.

Evaluation. We randomly sample 20,000 episodes and calculate mean accuracy as final results. We also apply *L2* normalization to video features before the one-shot classifier as the CMN does.

Table 1: Classification accuracy (%) of 5-way few-shot video domain adaptation on the test set of UnrealAction.

models	1-shot	2-shot	3-shot
BaseNet+test	44.2	52.3	57.8
VirF + test	43.5	52.2	57.9
VirF + testAug	44.6	52.7	58.2
VirF + testAugD	43.2	51.6	55.9
Ours	**44.8**	**53.2**	**59.0**

5.3 Results on UnrealAction

We conduct experiments on two settings: embodied one-shot video adaptation, and embodied one-shot video transfer recognition.

Baselines and Competitors. The difference between the two tasks lies in the video class set at the testing stage. In domain adaptation, the testing data belongs to the same class set as the virtual source data, while in transfer recognition, the testing data is selected from MiniKinetics. The same baselines and competitors are used for the two tasks. We report 1-shot, 2-shot, and 3-shot results on the 5-way setting.

(1) In the first baseline "BaseNet+ test", we use ResNet-50 pre-trained on ImageNet as feature extractor. (2) Then we fine-tune ImageNet pre-trained ResNet-50 on our virtual source data, and we denote this baseline as "VirF + test". (3) For "VirF + testAug", we explore fine-tuning our model only on D_{base} and applying our segment method in the testing stage. (4) We also adopt another virtual dataset DeepDrive [38] as our gallery. It is a synthetic dataset used in autonomous driving ("VirF + testAugD").

One-shot Video Domain Adaptation. The results are shown in Table 1. The numbers are reported in percentages. We can see that our method achieves the best performance in all the three shots which shows that the synthetic virtual dataset helps models to learn from the real data when they belong to the same classes. Consider that when novel concepts only have one or few available examples, it is difficult to find the corresponding real source data and label them. In contrast, we can utilize our virtual embodied agents and virtual environment to generate massive virtual videos and they can be easily used to train the recognition models over the new video concepts.

From the results of "VirF + testAugD", we note that the results using this irrelevant synthetic dataset as gallery set is worse than those with no gallery in most cases. In contrast, using our virtual dataset as gallery videos improves the recognition results consistently. This also validates the effectiveness of our augmentation methods which replaces the probe segments with semantic correlated and temporal consistent probe segments. In addition, we find that fine-tuning our model on $\tilde{\mathcal{D}}_{base}$ is also helpful in recognizing the generated video data.

As a pilot study on UnrealAction dataset, we report the per-class accuracy on the 5-way 1-shot setting in Figure 5. It is mean accuracy over all the queries in testing episodes, and the numbers are reported in percentages. The "bowing" category has the lowest accuracy, while "holding a baby" and "samba dancing" achieve the highest performance with 70.4% and 63.6 %, respectively.

Table 2: Classification accuracy (%) of 5-way few-shot video transfer recognition on the test set of MiniKinetics.

models	1-shot	2-shot	3-shot
BaseNet+test	63.5	74.4	78.6
VirF + test	54.2	65.6	69.2
VirF + testAug	56.6	65.3	68.8
VirF + testAugD	54.4	64.7	69.2
Ours	58.0	65.9	69.2

One-shot Video Transfer Recognition. The results are shown in Tabel 2. Comparing the results of "BaseNet + test" and "VirF + test", we can find that after fine-tuning our model on the virtual dataset, the performance drops a lot. That is because both domain and action categories are different in the virtual source domain and the real target domain. In such a case, ImageNet domain is much closer to our target video domain, which indicates that this novel task is very challenging. Hence, algorithms that can simultaneously reduce domain gap and learn new concepts with only one or few examples is expected in this novel task.

Comparing our method with "VirF + test", our segment augmentation method is still valid, especially in 1-shot setting. The performance improvement is decreasing as the number of examples increases, and a negative effect is observed in "VirF + testAug". The performance improvement in most cases can be attributed to that, the augmented videos help learning the prototypes of novel classes, as a representation of these classes in metric space.

When the number of well-labeled samples in the support set increases, they are good enough for learning the prototypes. In such a case, additional synthetic samples do not necessarily provide additional useful information, and may introduce some noisy information due to the augmentation process. This is especially the case in the task of embodied transfer recognition. For "VirF + testAug", we only fine-tune our model on D_{base}, so it does not have the ability of recognizing augmented video data, thus leading to the decrease in few-shot learning.

5.4 Results on MiniKinetics

Baselines and Competitors. We compare against several baselines and competitors as follows. (1) For the first baseline "BaseNet + test", we directly adopt ResNet-50 pre-trained on ImageNet as feature extractor. (2) Second, based on the first baseline, we apply segment augmentation method to the testing phase, which we denote as "BaseNet+testAug". (3) We use source videos $\mathcal{D}_{base}$ to fine-tune ResNet-50 pre-trained on ImageNet, and evaluate the fine-tuned model on the augmented test videos ("TrainFinetuned + testAug"). (4) We also compare our method with the state-of-art approaches, such as CMN [58], Matching Net [49], and MAML [15]. Matching Net is a neural architecture for image one-shot learning. MAML is famous for its meta learning strategy. CMN is designed for one-shot video action recognition. In CMN [58], Linchao Zhu et al. expand Matching Net and MAML into video recognition one-shot classifiers. Considering that our dataset and basic experimental settings are the same as them, we quote the experimental results of these three methods reported in CMN [58].

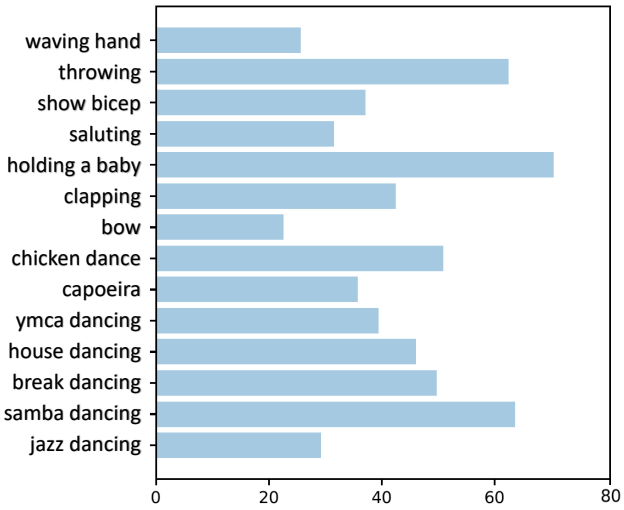

Figure 5: Per class accuracy (%) of our method on video domain adaptation task. We report the results on the 5-way-1-shot setting.

Results. We report our results on 5-way recognition tasks with 1-shot, 2-shot, 3-shot, 4-shot, and 5-shot, respectively. The results are shown in Table 3. We highlight several important results of the experiments. (1) Our video segment augmentation framework achieves the best performance on the MiniKinetics dataset. Even when we sample only 16 frames, our framework is significantly better than all other baselines and competitors. Our method achieves 67.8 % in the 5-way 1-shot task, improving 7.3% over CMN and 4.3 % over "BaseNet + test" baseline. And similar boost can be seen in all shots. (2) Comparing the results of "BaseNet + testAug" and "BaseNet + test", our video segment augmentation method boosts the performance in 1-shot and 2-shot settings even when we only apply it in the testing stage. And the performance drop in 3-shot, 4-shot and 5-shot settings is consistent with the results reported on the UnrealAction dataset. (3) Comparing the results of "TrainFinetuned + testAug" with the results of "BaseNet + testAug", it shows a steady rise. We can draw the conclusion that when the domain gap between the source domain and the target domain is small, fine-tuning on source dataset helps significantly. (4) Finally, the performance improvement from "TrainFinetuned + testAug" to ours indicates that training on $\tilde{\mathcal{D}}_{base}$ enables our model to have the ability of recognizing synthetic videos.

Ablation Study. We first explore different one-shot classifiers and then we conduct experiments on the number of frames. We report our results on 5-way recognition tasks with 1-shot, 3-shot, and 5-shot settings and all of them are performed in the way of "BaseNet + test". (1) As shown in Table 4, K-Nearest Neighbor (KNN), Support Vector Machine (SVM) and ProtoNet are explored. We can see that ProtoNet outperforms SVM in 1-shot and 5-shot, and is consistent with SVM in 3-shot. KNN is relatively poor especially when increasing the number of examples in the support set. (2) Each video in MiniKinetics lasts around 10 seconds and contains

Table 3: Classification accuracy (%) of 5-way few-shot video recognition on MiniKinetics. Our video segment augmentation method achieves the state-of-the-art results.

	Model	1-shot	2-shot	3-shot	4-shot	5-shot
Baselines	BaseNet + test	63.5	74.4	78.6	80.7	82.3
	BaseNet + testAug	65.6	74.5	78.0	78.8	81.5
	TrainFinetuned + testAug	67.6	76.8	79.2	82.2	82.9
Competitors	Matching Net [49]	53.3	64.3	69.2	71.8	74.6
	MAML [15]	54.2	65.5	70.0	72.1	75.3
	CMN [58]	60.5	70.0	75.6	77.3	78.9
Ours	Video Segment Augmentation Method	**67.8**	**77.8**	**81.1**	**82.6**	**85.0**

Table 4: Ablation study of different few-shot classifiers. ProtoNet outperforms KNN and SVM.

One-shot classifier	1-shot	3-shot	5-shot
KNN	63.2	72.8	76.3
SVM	63.1	78.6	80.4
ProtoNet (Ours)	**63.5**	**78.6**	**82.3**

Table 5: Ablation study of different frame numbers. As the number of frames increases, the improvement of accuracy is not significant.

Frame No.	1-shot	3-shot	5-shot
300	63.9	81.3	**83.9**
128	63.1	**81.5**	82.8
64	**64.4**	78.5	83.2
16 (Ours)	63.5	78.6	82.3

about 300 frames. In CMN [58], they propose a multi-saliency embedding algorithm to encode all the video frames into a fixed-size representation. In contrast, we only sample 16 frames-clip at a time yet achieves good results. This promotes us to explore whether it is necessary to leverage so many frames. We set the frame numbers as 16, 64, 128, 300 and make a comparison between them. The results are shown in Table 5. Results show that, when we increase the number of frames, the recognition accuracy improves steadily, which is consistent with the empirical conclusion that more frames introduce additional information. However, the improvement is not that significant, but the cost of time is several times than before. Additionally, if using all the frames of a video for training, we cannot crop clip from video during training, and the diversity of training data is decreased to some extent.

Visualization. To provide an intuitive sense of the capability of our video segment augmentation method, we visualize six classes in Figure 6. We compare our method with baselines "BaseNet+test" and "BaseNet + testAug". They are all conducted in a 5-way 1-shot setting. We first extract video-level features for each video in $\mathcal{D}_{novel}$ or $\tilde{\mathcal{D}}_{novel}$, and then we apply the t-SNE [31] algorithm to map them into a two-dimension metric space. Figure 6 (a) shows the distribution of $\mathcal{D}_{novel}$ videos when ImageNet pre-trained ResNet-50 is used as the feature extractor. The same color represents the same action class. Figure 6 (b) is the result of applying our video segment augmentation method on the testing stage. We can see that the newly generated videos are still close to the initial cluster, which shows that our method synthesizes videos while keeping the semantic information. Figure 6 (c) demonstrates that, after we fine-tune our video representation learning network on $\mathcal{D}_{base}$ and augmented base data $\tilde{\mathcal{D}}_{base}$, the distribution of classes shifts and the distance between inter-class increases especially when compared to Figure 6 (a).

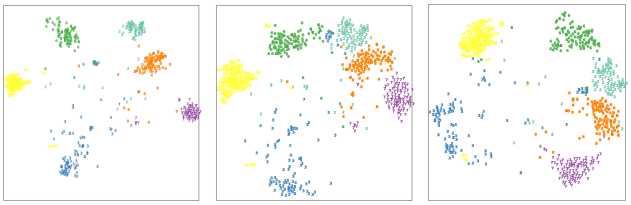

Figure 6: t-SNE visualization. Different colors represent different action classes. Six action classes are visualized in total. We compare our method with baselines "BaseNet + test" and "BaseNet + testAug".

6 CONCLUSION

To study the one-shot learning task in video domains, We present a novel setting — embodied one-shot video recognition, and introduce the corresponding UnrealAction dataset as a benchmark. The source videos of UnrealAction are created by capturing the actions of a virtual embodied agent in a virtual environment. Our setting is further split into domain adaptation and transfer recognition. In addition, we introduce a novel video segment augmentation method to synthesize new videos for limited datasets which performs well in practice. Extensive experiments are conducted on UnrealAction and MiniKinetics datasets, and we show that our method achieves the state-of-art performance.

ACKNOWLEDGMENT

This work was supported in part by National Key Research and Development Program of China under Grant 2018YFB1004300 and National Natural Science Foundation of China under Grant U1509206. We would like to thank Dr. Ye Pan for his help.

REFERENCES

[1] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. 2018. On evaluation of embodied navigation agents. In *ECCV*.

[2] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*.

[3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3D: Learning from RGB-D data in indoor environments. In *3DV*.

[4] Xiaojun Chang, Yi Yang, Alexander G. Hauptmann, Eric P. Xing, and Yao-Liang Yu. 2015. Semantic concept discovery for large-scale zero-shot event detection. In *IJCAI*.

[5] Xiaojun Chang, Yi Yang, Guodong Long, Chengqi Zhang, and Alexander G. Hauptmann. 2016. Dynamic concept composition for zero-example event detection. In *AAAI*.

[6] Xinlei Chen and C Lawrence Zitnick. 2015. Mind's eye: A recurrent visual representation for image caption generation. In *CVPR*.

[7] Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. 2018. Image deformation meta-network for one-shot learning. In *CVPR*.

[8] Zitian Chen, Yanwei Fu, Yinda Zhang, Yu-Gang Jiang, Xiangyang Xues, and Leonid Sigal. 2019. Multi-level semantic feature augmentation for one-shot learning. *TIP* (2019).

[9] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*.

[10] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. 2015. Flownet: Learning optical flow with convolutional networks. In *ICCV*.

[11] Li Fei-Fei, Rob Fergus, and Pietro Perona. 2003. A Bayesian approach to unsupervised one-shot learning of object categories. In *ICCV*.

[12] Li Fei-Fei, Rob Fergus, and Pietro Perona. 2006. One-shot learning of object categories. *PAMI* (2006).

[13] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. 2016. Convolutional two-stream network fusion for video action recognition. In *CVPR*.

[14] Basura Fernando, Efstratios Gavves, Jose M Oramas, Amir Ghodrati, and Tinne Tuytelaars. 2015. Modeling video evolution for action recognition. In *CVPR*.

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

[16] Yanwei Fu, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. 2014. Learning multimodal latent attributes. *PAMI* (2014).

[17] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*.

[18] Amir Habibian, Thomas Mensink, and Cees Snoek. 2014. VideoStory: A new multimedia embedding for few-example recognition and translation of events. In *ACM MM*.

[19] Nakamasa Inoue, Shanshan Hao, Tatsuhiko Saito, and Koichi Shinoda. 2009. Titgt at TRECVID 2009 workshop. In *Proc. TRECvid*.

[20] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2013. 3D convolutional neural networks for human action recognition. *PAMI* (2013).

[21] Yu-Gang Jiang, Zuxuan Wu, Jinhui Tang, Zechao Li, Xiangyang Xue, and Shih-Fu Chang. 2018. Modeling multimodal clues in a hybrid deep learning framework for video classification. *IEEE Transactions on Multimedia* (2018).

[22] Johan C Karremans, Wolfgang Stroebe, and Jasper Claus. 2006. Beyond Vicary's fantasies: The impact of subliminal priming and brand choice. *JESP* (2006).

[23] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. 2008. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*.

[24] Orit Kliper-Gross, Tal Hassner, and Lior Wolf. 2011. One shot similarity metric learning for action recognition. In *International Workshop on Similarity-Based Pattern Recognition*.

[25] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML*.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NeurIPS*.

[27] Brenden M. Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B. Tenenbaum. 2011. One shot learning of simple visual concepts. In *CogSci*.

[28] Ivan Laptev. 2005. On space-time interest points. In *ICCV*.

[29] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. 2008. Learning realistic human actions from movies. In *CVPR*.

[30] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. SSD: Single shot multibox detector. In *ECCV*.

[31] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *JMLR* (2008).

[32] Erik G. Miller, Nicholas E. Matsakis, and Paul A. Viola. 2000. Learning from one example through shared densities on transforms. In *CVPR*.

[33] Ashish Mishra, Vinay Kumar Verma, M Shiva Krishna Reddy, S Arulkumar, Piyush Rai, and Anurag Mittal. 2018. A generative approach to zero-shot and few-shot action recognition. In *WACV*.

[34] Timothy E Moore. 1982. Subliminal advertising: What you see is what you get. *Journal of marketing* (1982).

[35] Paul Over, George Awad, Martial Michel, Jon Fiscus, Wessel Kraaij, and Alan F. Smeaton. 2011. TRECVID 2011 – An overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2011*.

[36] Weichao Qiu and Alan Yuille. 2016. Unrealcv: Connecting computer vision to unreal engine. In *ECCV*.

[37] Zhaofan Qiu, Ting Yao, and Tao Mei. 2017. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*.

[38] Craig Quiter and Maik Ernst. 2018. Deepdrive/deepdrive: 2.0.

[39] Thomas Zoëga Ramsøy and Morten Overgaard. 2004. Introspection and subliminal perception. *Phenomenology and the cognitive sciences* (2004).

[40] Sachin Ravi and Hugo Larochelle. 2017. Optimization as a model for few-shot learning. In *ICLR*.

[41] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *CVPR*.

[42] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. 2016. Playing for data: Ground truth from computer games. In *ECCV*.

[43] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*.

[44] Jake Snell, Kevin Swersky, and Richard S. Zemeln. 2017. Prototypical networks for few-shot learning. In *NeurIPS*.

[45] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas Funkhouser. 2017. Semantic scene completion from a single depth image. In *CVPR*.

[46] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human action classes from videos in the wild. *CRCV* (2012).

[47] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. The new data and new challenges in multimedia research. *Commun. ACM* (2016).

[48] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*.

[49] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *NeurIPS*.

[50] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *CVPR*.

[51] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. 2018. Low-shot learning from imaginary data. In *CVPR*.

[52] Yu-Xiong Wang and Martial Hebert. 2016. Learning from small sample sets by combining unsupervised meta-training with CNNs. In *NeurIPS*.

[53] Yu-Xiong Wang and Martial Hebert. 2016. Learning to learn: Model regression networks for easy small sample learning. In *ECCV*.

[54] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. 2017. Learning to model the tail. In *NeurIPS*.

[55] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.

[56] Ming yu Chen, Huan Li, and Alexander Hauptmann. 2009. Informedia @ TRECVID 2009: Analyzing video motions. In *Proc TRECvid*.

[57] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. 2015. Beyond short snippets: Deep networks for video classification. In *CVPR*.

[58] Linchao Zhu and Yi Yang. 2018. Compound memory networks for few-shot video classification. In *ECCV*.