2022 31st IEEE International Conference on Robot and
Human Interactive Communication (RO-MAN)
August 29 - September 2, 2022. Naples, Italy.

Salient Keypoints for Interactive Meta-Learning (SKIML)

Wallace Lawson[1], Anthony Harrison[1], Mai Lee Chang[2], William Adams[1], and J. Gregory Trafton[1]

***Abstract*— Learning to recognize new objects in real time in unconstrained environments presents significant challenges for robotic platforms. We present a meta-learning solution to this problem as well as a registered image and events dataset to facilitate work in this domain. Our solution uses interactive motion to isolate the object, and motion-based saliency (from events) to select relevant keypoints from a high-resolution RGB image. Salient keypoints are then passed to a meta-learner to classify the object type. We show that using our interactive isolation and keypoint selection approach, we outperform existing techniques by 6-20%.**

I. Introduction

Imagine a robot being unpacked for the first time. The robot's pre-programmed range of skills will probably include tasks it knows how to accomplish (e.g., vacuuming or replacing a lightbulb), objects it can identify (e.g., a vacuum or a lightbulb), and interaction techniques (e.g., social norms of greeting or accepting instructions). All of these skills are under active study by researchers in interactive task learning [1], computational perception [2], and human robot interaction [3].

A critically important skill is the ability to recognize objects in new environments. Fortunately, computational perception has improved a great deal in the last decade, primarily due to deep networks and big data (e.g., ImageNet [4]). Most of this work has focused on training deep networks on millions of images [5], [6]. Frequently, the evaluation criteria focuses on performing better than a previous best algorithm on a specific dataset; this has lead to both qualitative and quantitative improvements in computational perception / computer vision.

While big data has helped to improve performance, the requirement of using big data can be a limitation. Current perceptual systems require huge amounts of data (1000 samples per class is a common heuristic). Most interactive robotics users will find this to be an unrealistic, burdensome prerequisite. For practical use, a system must be able to learn to recognize an object from very few instances – typically five [7], [8].

This work was supported by the Office of Naval Research (GT, WL, AH). The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Navy.

[1]US Naval Research Laboratory; Washington, DC, USA {ed.lawson, anthony.harrison, greg.trafton, william.adams}@nrl.navy.mil

[2]University of Texas at Austin; Austin, TX, USA mlchang@utexas.edu

Learning from limited data has been a goal in object recognition for quite some time. Although humans do this effortlessly, we are only beginning to develop systems that have this ability. A key insight that has facilitated this work is that we need systems that can learn how to learn [9], [10] (i.e., meta learning). Rather than training our system to recognize a thousand classes, instead we provide it with a number of training "episodes", each of which contains a small learning problem consisting of a few object classes and a few examples from each object class. Over time, our system learns ways that it can quickly and effectively learn to recognize objects given limited training data.

Although this provides the ability to learn quickly, it sometimes overemphasizes contextual cues as the meta-learner does not attempt to isolate the object of interest. This causes the meta learner to fail in rather predictable ways when used interactively. Teaching our robot to recognize objects involves providing objects with limited context, a collaborator usually just holds the object up for the robot to see [11]. During training episodes, the meta learner learns to recognize the differences between images, and much of the time these differences are not exclusively the objects themselves. Thus, as we will show, meta-learners are extremely sensitive to (now irrelevant) contextual cues. To illustrate, consider the case of a competitive meta learner (MetaOptNet [12]) when given a task of recognizing 5 objects with 5 examples per object class. On the the miniImageNet dataset [13] this is completed with a rather acceptable 80% accuracy. On our proposed heldheld objects dataset, this accuracy drops to 48%.

We propose to address this problem in two different ways using a novel learning system called Salient Keypoints for Interactive Meta-Learning (SKIML). We leverage meta-learning [7], [8] that allows training with less data and does not need extensive re-training time. We use *interaction* to allow us to identify the portion of the scene that contains a to-be-learned object, permitting us to remove irrelevant details. Our interactive perceptual learning system allows a human teacher to hold an object, verbally identify it, and focus the robot's attention to the object by wiggling it. A dynamic vision sensor (DVS) camera captures the object motion. DVS cameras have high throughput and dynamic range which means that they can capture rapid motion even when there is poor lighting. We use the motion in two ways. First, this motion provides a means of focusing the attention of the robot. We use motion to isolate object on a registered camera where a standard image can be processed. Second, we use visual saliency [14] to analyze the motion and provide regions whose motion is salient. We use *motion saliency* to

determine which regions of the the object to use as keypoints. We train a meta-learner using the cropped image as well as the regions around each salient motion keypoints.

The objective of our approach is to improve few shot learning accuracy on handheld tools recognition. Using our approach, we improve performance by 6% up to 20% on our newly proposed shared attention tool dataset.

A. Contributions

- We propose a novel technique that we call *Salient Keypoints for Interactive Meta-Learning (SKIML)* for recognizing objects through interaction. Salient keypoints isolate a region of interest by finding high saliency regions when a user shows an object to the robot.
- We present a novel meta-learner that builds an object-based representation based on both salient motion keypoints and a standard image.
- We show that our complete system is better suited for interactive object learning than current meta-learning approaches.
- We release the Shared Attention Tool Dataset – a publicly available dataset for use by others interested in interactive few-shot learning focused on human-robot interaction (HRI) applications.[1]

II. Related Work

A. Traditional Object Recognition and Datasets

Traditional object learning systems typically include two distinct phases: a learning phase and an evaluation phase. During the learning phase, a large amount of labeled object data is collected. This data can be collected experimentally (e.g., on a turntable or a platform; [15]) or naturalistically (e.g., pictures from mobile phones; [16]). The recognition process includes image segmentation and object classification [4], [17]. A deep network is then trained to learn important features that are useful to identify the object. For a literature review of state-of-the-art object recognition methods, we direct interested readers to [17]. However, this approach does not extend to HRI where users have a smaller dataset that may be specific to their needs. Thus, we created the Shared Attention Tool Dataset that contains high-resolution images of common tools found in most homes.

B. Interactive Perceptual Learning on a Robot Platform

As suggested above, much of the work on object recognition emphasizes methods for collecting and labeling data and not on learning novel objects through interaction. There have been several researchers, however, who have trained novel objects using a robotics platform. Martinson [18] developed a technique to learn novel objects by picking them up, labeling them by entering that information into a script, and showing the object to a mobile sensor. A bounding box was generated by finding the hand holding the object and then creating possible bounding boxes around the hand. The object was then placed on a surface to capture RGB-D images as the person walks around the object to emulate BigBird [15]. The user then placed the object on several other locations and, critically, augmented the collected data with synthetic data. They achieved a 99% precision and a 57% recall.

Narayanan et al. [11] asked users to show an MDS robot a suite of objects and label them. The labeling was accomplished via computer interface by the experimenter and a deep network was trained on each user's data and labels. One consequence of this result was that performance was better (approximately 75%) than when the robot did not show how to present the object to the robot (approximately 60%). We improve upon this work by dramatically shortening the amount of time required to learn to recognize an object. We also provide an improved method for isolating the object of interest using motion from a DVS sensor, rather than segmenting images based on the closest point to the sensor.

Azagra [19] used a very difficult, cluttered dataset to learn and evaluate through incremental learning. They used language, pointing, and explicit showing of approximately 20 objects. Their offline training/evaluation system achieved an accuracy of approximately 18% when they performed automatic segmentation of the object. Their incremental online system achieved an accuracy of approximately 13% when they performed automatic segmentation of the object. Note that while these numbers are not extremely high, their dataset was extremely challenging. We demonstrate our approach on a new dataset with 54 distinct objects organized into 12 general categories.

Pasquale et al. [20] used iCub to train a convolutional neural network to recognize 28 objects (7 categories). iCub was able to hold and track an object as it received a verbal label from a nearby human. Their offline training and evaluation across 4 days achieved an average accuracy of 70%. While our goals are similar, our approach is able to learn new objects online.

All of these systems used a deep network approach for learning, though their approach differed slightly (e.g., Narayanan et al. and Pasquale et al. fine-tuned AlexNet for each participant; Martinson used a deep network to train from scratch). Martinson, Narayanan et al., and Pasquale et al. used offline training, and most used offline evaluation. Critically, none of these systems (or any others that we know of) used an online learning system capable of learning new objects from people within interaction time.

Our proposed method differs from these in several ways, the most prominent difference is that our approach learns online, while all of these methods learn offline. Online learning is especially difficult in cluttered environments and when there are distractors available [17]. We will address this problem through interaction and specifically through joint attention.

Joint Attention A fundamental component of interaction is joint attention. This occurs when two or more individuals focus on the same object [21]. Joint attention between

[1]The dataset will be made available after organizational approval.

humans is typically verified visually, often through body language and eye contact, but can also occur through gestures, language, touch, or other modalities. Accomplishing joint attention between a robot and a human, however, requires deliberate effort to ensure the robot is attending to areas or objects in the environment the human expects. Although there are examples of robots that can follow or learn a human partner's gaze in order to establish joint attention [22], [23], we chose instead to use a slightly different approach. In our case, we use the human's actions to direct the robot's focus to a specific location in space. To accomplish this, we use a Focus of Attention module (discussed below).

Meta Learning

Meta-learning was originally proposed as a method to improve the performance of few-shot learning by explicitly teaching the network how to perform well with less data. Vinyals et al. [10] was one of the first in this area to suggest the concept of meta-learning. They propose MatchingNets, a method where the features of the query samples are contrasted with features derived from a set of support samples. Snell et al. [24] propose a different approach, ProtoNet, where the training samples are instead used to learn a prototype representative of the class as a whole. We use ProtoNets as the basis of our approach, but in addition to the image, the ProtoNet is also given small regions cropped around motion salient keypoints. We further expand it to use a support vector machine for object classes, rather than the single prototype originally proposed.

Recent advances in meta-learning have extended this to fine-tuneable networks [9]. Interventional few-shot learning focuses on methods to improve domain transfer [25]. Chen et al. [26] study the differences in using different backbone architectures in an effort to determine an architecture better suited for transfer learning. More recent work has begun to focus on the problem of few-shot detection [27], [28], but the accuracy of few-shot detection is much lower than recognition and better performing approaches depend on offline learning approaches (fine-tuning).

Here, we show that good performance can be achieved with shallower networks when given better images from which to train and evaluate. Additionally, shallower networks are a better fit for mobile robotics platforms as they typically have very constrained GPU resources.

III. Methodology

A. Interaction

The interaction with the robotic system has been modeled after the observations of natural object learning scenarios (e.g., infants and novices learning new objects) [29]. In this interaction scenario, the operator is teaching the robot a new task sequence involving some set of tools that are unknown to the robot. The operator must teach the tools to the robot first. Each tool is visually presented to the robot along with its verbal label (e.g.,"This is a screwdriver"). When presenting the tool, the operator can utilize any of a number of shared attention strategies, that is, methods to ensure the operator and the robot are paying attention to the same thing in the environment. For the purposes of this paper, we are only focused on one strategy: wiggling the new object, though other methods of focusing attention can be added to the system at a later date.

B. System

The robotic system is broadly outlined in Figure 1. The SCIPRR reconfigurable head [30] is equipped with an Inivation DAVIS-346 DVS camera, a FLIR Grasshopper, and microphones. The cameras are calibrated as discussed below. These provide the raw sensory input to the first system which deals exclusively with the interaction with the user. The Focus of Attention Module is responsible for taking the raw sensor outputs and finding the candidate object within the scene. It then passes cropped and calibrated RGB and event images to the second system to be learned.

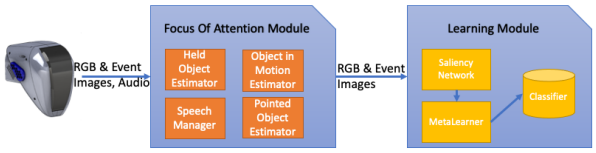

Fig. 1. Simplified robotic system diagram. Here, we show that the focus of attention module provides both images and DVS events to the learning module, which is shown in more detail in Figure 2.

1) Focus of Attention Module: The Focus of Attention Module is designed to handle multiple different styles of interaction, specifically modes of drawing attention to a particular object in space. Using verbal commands to differentiate between modes, the module can find objects that are between two hands (Held Object Estimator, using OpenPose [31]), being pointed at (Pointed Object Estimator, using OpenPose and DVS-based motion detection), or being wiggled (Object in Motion Estimator, using DVS-based motion detection).

C. Learning Module

Here, we describe our learning pipeline. The system diagram is shown in Figure 2, with additional examples of each step on this pipeline in Figure 3. Our calibrated sensors provide both the visual image as well as the DVS events from motion.

1) UNISAL and the Salient Motion Map: Wiggling the object produces DVS motion events, which are then used to both segment the object and locate salient motion keypoints. A DVS motion event is captured as (x, y, t) where (x, y) is the location of the event and t is the observed time interval. The observed motion is captured by summing the observed motion and computing a motion map $M = \sum_{i=t}^{t+\delta t}(x, y, i)$

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

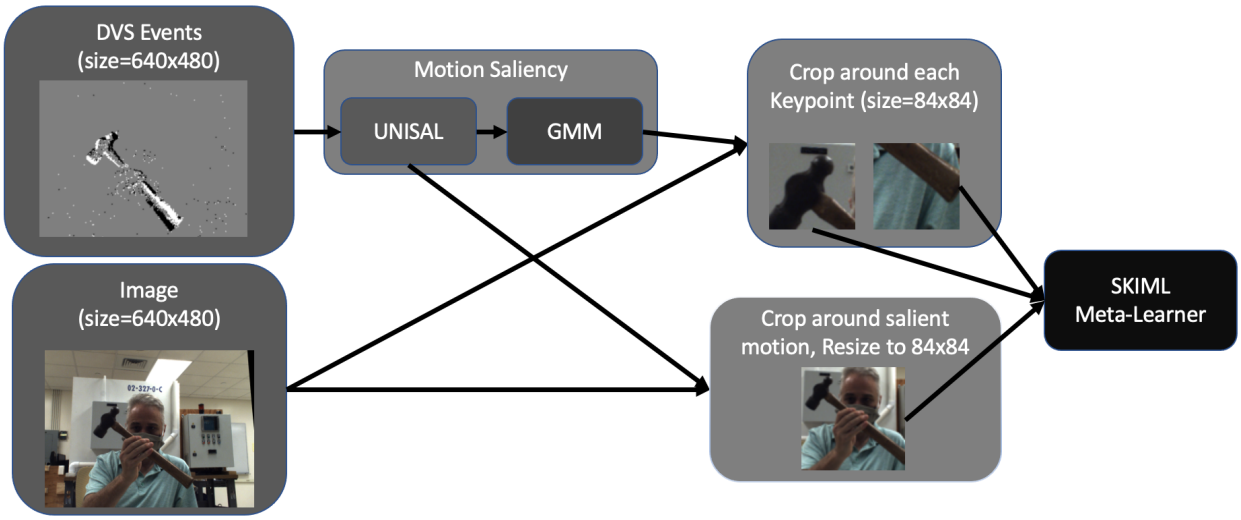

Fig. 2. System flow diagram for the learning module, showing how the learning system uses input from the DVS events to learn representations for the objects. Interaction into the system comes through motion observed by the DVS camera in the events image. The meta learner includes both the embedding network and the support vector machine, as described in our methodology section.

where δt is set to a time interval of around 1 second. Figure 3, second column, shows different colors for positive and negative event polarity, which shows a transition from dark to light regions or light to dark regions, respectively. Note that DVS cameras do not report frames, they report events. For this reason, we are able to capture even very rapid wiggling motions.

Given the motion map M, we hypothesize that components of an object that would draw the attention a human eye are good candidates to facilitate classification. Therefore, we used saliency estimation [14], [32], [33], whose goal is to train a deep network to predict visual saliency when given with a large training set. The SALICON dataset annotates images from the MSCOCO dataset with visual saliency, collected by allowing users to "free view" images (e.g., viewing without any stated objective). We wish to compute the visual saliency $S(M)$ for our motion map. For this, we use the UNISAL neural network recently proposed by Droste et al. [14]. UNISAL models visual saliency on both image and video data. Their results show that UNISAL achieves state-of-the-art performance on the video datasets and is on par with the state-of-the-art for the image datasets.

Figure 3 shows M in the 2nd column and $S(M)$ in the third column. Encouragingly, UNISAL predicts that noise in the image will not draw the attention of the human observer. It also predicts that regions with more motion will draw a greater amount of attention.

We use the motion saliency map $S(M)$ in two different ways. First, by locating highly salient regions, we extract a bounding box around the object shown by the human instructor. We compute the bounding box by applying a small threshold (in our work, we select $S(M) > 50$), although in our observations our approach is not highly sensitive to this threshold. As shown in the system diagram (Figure 2), we crop around the salient motion and resize this to an 84×84 image and use this as one of the images input (X_{ci}) to the meta-learner (discussed below).

Further improvements in accuracy can be realized by examining $S(M)$ to find regions where the attention is locally maximum, which indicates that this is a region will draw the attention of a human observer. The wiggle from the human observer produces more motion at the extreme points of the tool (typically, the handle and the end effector). In this next step, we convert this to keypoints which can be used to "fixate" upon. To locate keypoints, we fit a Gaussian Mixture Model (GMM) to $S(M)$, in order to estimate a mixture of two Gaussians. The two located peaks are shown in the 4th column of Figure 3, showing keypoints using a small green circle. We crop a small region around each of these keypoints and also use these to train our meta-learner.

To summarize, the meta learner is provided with three images, the cropped, resized bounding box around the object (X_{ci}), as well as small regions cropped around each of the keypoints (X_{kp1}, X_{kp2}). Note that the regions are cropped around the center of each keypoint from the original 640×480 image, which provides these images to the meta-learner in greater detail.

2) Meta Learner: Meta learning uses a small support set $L = \{(x_1, y_1), (x_2, y_2)...(x_n, y_n)\}$ of N labeled training images, where each $x_i \in \mathbb{R}^D$ is the input and y represents the ground truth label. Given this, we wish to building an embedding function $f_\theta = \mathbb{R}^D \rightarrow \mathbb{R}^M$ that computes an embedding of the input values [24].

Our embedding function f is based on the architecture suggested by [24], with 4 convolutional blocks each with a 3×3 convolution, batch normalization, rectified linear norm activation, and a 2×2 max pooling. Multiple salient regions are combined with a 2 layer LSTM, which produces the final embedding.

For input i, each of the three input images:

$[X(i)_{kp1}, X(i)_{kp2}, X(i)_{ci}]$ are presented to the meta learner, with the cropped keypoint images presented first and the resized image presented last. This produces an embedding representation $\mathbb{R}^D$.

We train using the approach suggested by Snell et al. [24]. The meta-learner finds a "prototype" in the form of the average embedding for each object class. This is done by first computing the embedding for each training image, with a final step to compute the average of all embeddings in that class, resulting in a single prototype representing that class. The class of query images is predicted by finding the encoding that is closest, with the meta-learner trained to minimize cross-entropy loss.

Rather than using the average embedding for each class, instead we use a support vector machine (SVM) to learn a one-vs-one classification for each class (SVC). This idea is inspired by MetaOptNets [12], but we choose to do this after training while still retaining the smaller network architecture, as this is better suited for most mobile robotics platforms.

We build our meta-learner by pre-training on the Mini-ImageNet dataset [13], using the parameter suggested by [24].

D. Sensors and Calibration

The DAVIS-346 camera simultaneously produces both an events image and a color image. Although the color image is useful, it is generally very low resolution and the image quality is poorly suited for recognizing objects. For that reason, we calibrated the DAVIS-346 with the FLIR Grasshopper. Calibration uses the standard ROS calibrated camera pipeline, slightly modified to accommodate sensors of different resolutions.

E. Shared Attention Tool Dataset

Interactions with robots can take substantial time, especially for repeated online learning of new objects. For the purposes of this particular problem, we have abstracted away portions of the interaction (e.g. speech), resulting in a hybrid image dataset more suitable for automated analyses. We created the Shared Attention Tool dataset [2] which contains high-resolution images (i.e., 640x480) of common tools representative of those found within most homes.

The dataset includes a total of 54 distinct objects (tools), falling into one of twelve classes, with at least four instances in each class. For each tool, eight different poses were collected under varying conditions of presenter appearance, lighting, and background. For each sample, two images were acquired, an RGB image from the color camera before movement and a consolidated event image of the wiggle motion from the DVS camera.

1) Consolidated Event Image: DVS cameras are high-speed devices that capture changes in infrared luminosity. Instead of returning a single image frame at a fixed update rate, the camera provides asynchronous events which report changes to per-pixel luminosity. To support processing with traditional frame-based image processing pipelines, DVS event data can be binned over time to produce consolidated event images. For the purposes of this dataset, event data was consolidated over 500ms, effectively capturing the movement of the tool during enrollment.

2) Tool Classes: Twelve tool classes were selected based on common tools. Each instance within each class is visually distinct from the other (i.e., avoiding classes like *socket wrench* which differ only in size). Some classes have a high degree of similarity both in structure and function (e.g., *hammer* and *mallet*). Others classes may only differ based upon the end effector (e.g., *wire cutters* and *pliers*). The granularity of the classes was determined largely by what features are resolvable at the given sensor resolution. For example, *screwdriver* encompasses both flat and phillips heads because they are identical at 640x480. The twelve classes are: adjustable wrench, box cutters, channel lock pliers, hammer, mallet, pliers, screwdriver, square, tape measurer, tin snips, wire cutters and wrench.

IV. Experimental Results

We evaluate our approach on the Shared Attention Tool dataset described earlier. In our experiment, we evaluate performance on four different interactive teaching and evaluation scenarios. In the first two, a human teaches a robot to recognize five randomly selected tool classes, whereas in the second a human teaches a robot to recognize ten randomly selected tool classes. In each experiment we provide a single example of each (i.e., 5-way or 10-way, 1-shot). A second evaluation analyzes performance when providing five examples of each tool class (i.e., 5-way or 10-way, 5-shot).

The experiment is performed randomly 2000 times. We find the object using the focus of attention module and classify using the MetaLearner (see Figure 1). In practice, object labels would be provided by the speech manager, but when collecting the shared attention tool dataset we entered these labels manually to encourage image recognition.

In our experiment, objects are randomly sampled from the 12 available tool categories each time. Support (training) images are selected randomly without replacement; 15 query images per class are also selected randomly without replacement. We report the average accuracy and a 95% confidence interval, shown in table I.

To evaluate the accuracy of our approach, we compare against two popular approaches: ProtoNets [24] and MetaOptNet [12]. Both results were computed using the reference implementation provided by the authors of MetaOptNet [3]. The focus of attention module was not used for

[2]available after institutional approval

DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

[3]https://github.com/kjunelee/MetaOptNet

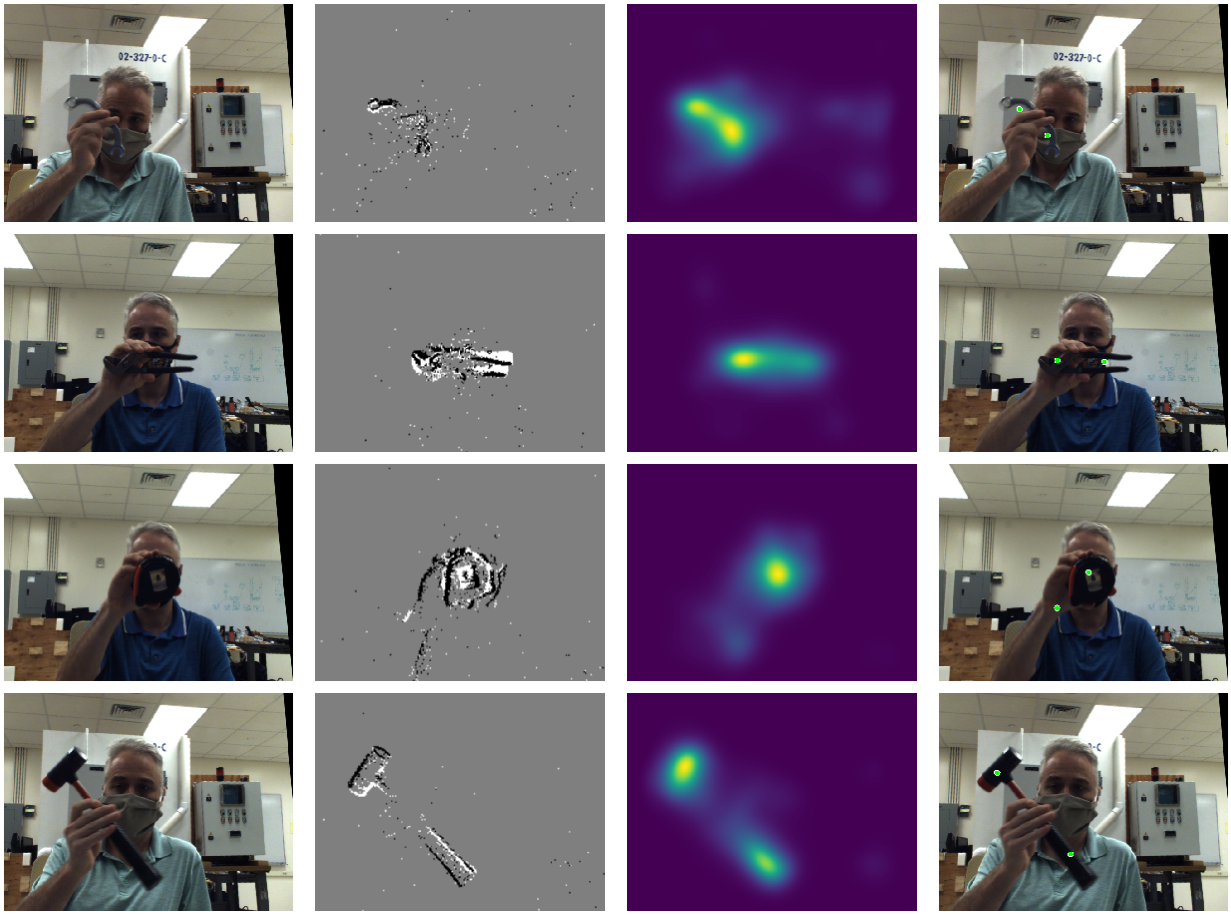

Fig. 3. Our processing pipeline, from left to right: the image of various tools as captured in a cluttered environment; the DVS image capturing events occuring over approximately 1 second; the motion saliency image showing a heatmap of which regions attract attention; keypoints selected

either approach. We used their approach as suggested by the authors.

As Table I shows, the Shared Attention Tool dataset is quite challenging. The 5-way, 5-shot data represents what may be a common scenario where a 5 tools are needed to perform a task and an instructor provides 5 different images of each. Protonet and SKIML with only the focus of attention performs the worst, at approximately 40%. MetaOptNet provides an improvement, operating at approximately 50% accuracy. SKIML with both the focus of attention and keypoints performs the best at 63% accuracy.

One trend that is evident in this experiment is that the performance difference between SKIML and MetOptNet grows with additional support (training images). With 5-way, 1-shot, SKIML outperforms MetaOptNet by 6.5%; with 5-way, 5-shot SKIML outperforms MetaOptNet by a much wider margin (about 15%). This trend is also present with the 10-way experiment. Here, SKIML outperforms MetaOptNet by 10% with 1 training image and by over 20% with 5 training images. This trend is encouraging and suggests that both SKIML, the focus of attention and keypoints are making good use of the limited amount of training data.

We also note the importance of using salient motion keypoints. SKIML without keypoints somewhat comparably to ProtoNets and substantially worse than MetaOptNet. However, once keypoints are added to the system architecture, SKIML outperforms existing approaches by 10-20%.

Finally, our approach can quickly learn new objects. Training the meta-learner and support vector machines takes on average 0.15 seconds per batch, which we evaluate on a 5-way, 5-shot scenario. GMM adds an additional 0.15 to 0.2 seconds per image. In sum, this means we can learn a new object in less than 0.35 seconds. Both training and evaluation are within a reasonable interaction time, which achieves our stated goal of interactive object learning.

Like many recent meta-learners, MetaOptNet uses the deeper ResNet-12 architecture as some [26] have shown that these have good generalization properties. They do present a trade-off for SWAP constrained robotics platforms, which may not have the GPU resources to run (possibly several) deep networks. We designed SKIML to work with limited GPU resources typically available on mobile platforms, which is why it is based on the much smaller ProtoNet architecture.

TABLE I

ACCURACY RESULTS FOR PERFORMANCE EVALUATION. ALL RESULTS ARE PRESENTED WITH 95% CONFIDENCE INTERVALS.

Classes	Support	MetaOptNet	ProtoNet	SKIML Focus of Attention	SKIML Focus of Attention and Keypoints
5-way	5-shot	48.21 ± 0.67%	35.35 ± 0.43%	39.37 ± 0.34%	**63.19 ± 1.00%**
5-way	1-shot	29.54 ± 0.60%	28.76 ± 0.38%	28.20 ± 0.28%	**35.09 ± 0.81%**
10-way	5-shot	36.65 ± 0.41%	23.23 ± 0.25%	26.45 ± 0.17%	**57.27 ± 0.68%**
10-way	1-shot	20.08 ± 0.37%	19.56 ± 0.24%	17.31 ± 0.16%	**30.37 ± 0.63%**

V. DISCUSSION

To summarize, we present an approach that permits us to learn new objects using only a few examples. Our approach can learn new objects and can recognize objects quickly and does not require a lengthy or cumbersome offline learning process. Interaction is key in our approach as this allows us to both segment the object and to identify keypoints.

We are hopeful that with further study we can continue to reduce the number of training images required. This experiment represents the typical performance that might be achieved if a person were to randomly select a few tools to meta-train the robot for immediate recognition. We have shown that performance increases with additional examples. So, although the performance of 10-way, 5-shot is preferable, it requires providing 50 different images to the robot for training. Future work should continue to expand on this idea on how to do more with less training data.

A meta learner could be one piece of a larger system designed to robustly recognize a wide number of objects. Our meta-learner can be used for rapid learning with an occasional error, which then provides data to train a larger network whose training would take more time (e.g., over night to build a more robust and accurate network). Thus, over time we continually provide a method to both learn and to improve performance. It may be possible to use the learning experiences from different robots to build an even more robust network.

Finally, it's important to relate our problem (object recognition) to the detection problem which both identifies the object and provides a bounding rectangle. We provide a solution on how to recognize handheld objects. For detection, one approach may be to use the focus of attention module to build a region of interest, which could then be used to train a region proposal network. By combining motion, pointing, body pose, etc. we could develop multiple ways to generate these regions of interest which could then be used to interactively train an object detector in the style of the R-CNN two stream object detectors.

Finally, a note on our dataset. We provide a large dataset that has registered motion and visible images. Our hope is that this dataset can be used in the future to study the problem of handheld object recognition. To the best of our knowledge, this is the largest existing dataset that contains interaction and registered images from a visible and DVS camera.

ACKNOWLEDGMENT

Thanks to Magda Bugajska and Eric Vorm for comments on a previous draft.

REFERENCES

[1] J. E. Laird, K. Gluck, J. Anderson, K. D. Forbus, O. C. Jenkins, C. Lebiere, D. Salvucci, M. Scheutz, A. Thomaz, G. Trafton *et al.*, "Interactive task learning," *IEEE Intelligent Systems*, vol. 32, no. 4, pp. 6–21, 2017.

[2] C. Devin, P. Abbeel, T. Darrell, and S. Levine, "Deep object-centric representations for generalizable robot learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7111–7118.

[3] B. Mutlu, T. Shiwa, T. Kanda, H. Ishiguro, and N. Hagita, "Footing in human-robot conversations: how robots might shape participant roles using gaze cues," in *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*. ACM, 2009, pp. 61–68.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[7] J. Lu, P. Gong, J. Ye, and C. Zhang, "Learning from very few samples: A survey," *arXiv preprint arXiv:2009.02653*, 2020.

[8] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *arXiv preprint arXiv:2004.05439*, 2020.

[9] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *International Conference on Machine Learning*, 2017.

[10] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *Advances in neural information processing systems*, 2016, pp. 3630–3638.

[11] P. Narayanan, M. D. Bugajska, W. Lawson, and J. G. Trafton, "Impact of embodied training on object recognition," in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2017, pp. 1338–1343.

[12] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 657–10 665.

[13] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," *International Conference on Learning Representations*, 2016.

[14] R. Droste, J. Jiao, and J. A. Noble, "Unified image and video saliency modeling," *European Conference on Computer Vision*, 2020.

[15] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, "Bigbird: A large-scale 3d database of object instances," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 509–516.

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[17] M. Bansal, M. Kumar, and M. Kumar, "2d object recognition techniques: State-of-the-art work." *Archives of Computational Methods in Engineering*, vol. 28, no. 3, 2021.
[18] E. Martinson, "Interactive training of object detection without imagenet," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.
[19] P. Azagra, A. C. Murillo, M. Lopes, and J. Civera, "Incremental object model learning from multimodal human-robot interactions," *Neural Information Processing Systems*, 2018.
[20] G. Pasquale, C. Ciliberto, F. Odone, L. Rosasco, and L. Natale, "Teaching icub to recognize objects using deep convolutional neural networks," in *Machine Learning for Interactive Systems*, 2015, pp. 21–25.
[21] V. Corkum and C. Moore, "Development of joint visual attention in infants." *Joint attention: Its origins and role in development*, 1995.
[22] B. Scassellati, "Imitation and mechanisms of joint attention: A developmental structure for building social skills on a humanoid robot," in *International Workshop on Computation for Metaphors, Analogy, and Agents*. Springer, 1998, pp. 176–195.
[23] J. Gregory Trafton and A. M. Harrison, "Embodied spatial cognition," *Topics in cognitive science*, vol. 3, no. 4, pp. 686–706, 2011.
[24] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4077–4087.
[25] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot object detection via feature reweighting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8420–8429.
[26] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," *International Conference on Learning Representations*, 2019.
[27] X. Wang, T. E. Huang, T. Darrell, J. E. Gonzalez, and F. Yu, "Frustratingly simple few-shot object detection," *International Conference on Machine Learning (ICML)*, July 2020.
[28] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot object detection via feature reweighting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8420–8429.
[29] C. Yu and L. B. Smith, "Joint attention without gaze following: Human infants and their parents coordinate visual attention to objects through eye-hand coordination," *PloS one*, vol. 8, no. 11, p. e79659, 2013.
[30] A. M. Harrison, W. M. Xu, and J. G. Trafton, "User-centered robot head design: A sensing computing interaction platform for robotics research (sciprr)," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '18. New York, NY, USA: ACM, 2018, pp. 215–223. [Online]. Available: http://doi.acm.org/10.1145/3171221.3171283
[31] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields," in *arXiv preprint arXiv:1812.08008*, 2018.
[32] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "Salicon: Saliency in context," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
[33] I. Ullah, M. Jian, S. Hussain, J. Guo, H. Yu, X. Wang, and Y. Yin, "A brief survey of visual saliency detection," *Multimedia Tools and Applications*, vol. 79, no. 45, pp. 34 605–34 645, 2020.