ARL-TR-8356 ● MAY 2018

US Army Research Laboratory

An Initial Approach for Learning Objects from Experience

by Troy Dale Kelley, Sean Michael McGhee, and Jonathan Milton

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-8356 ● MAY 2018

US Army Research Laboratory

An Initial Approach for Learning Objects from Experience

by Troy Dale Kelley, Sean Michael McGhee, and Jonathan Milton
Human Research and Engineering Directorate, ARL

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
May 2018	Technical Report	1 October 2016 – 8 February 2018

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
An Initial Approach for Learning Objects from Experience	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	5d. PROJECT NUMBER
Troy Dale Kelley, Sean Michael McGhee, and Jonathan Milton	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
US Army Research Laboratory ATTN: RDRL-HRF-D Aberdeen Proving Ground, MD 21005-5425	ARL-TR-8356

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The US Army Research Laboratory's Vehicle Technology Directorate (VTD) and the Human Research and Engineering Directorate, as part of VTD's 6.1 refresh program, have initiated a program called Adaptive Perception Processes for Learning from Experience (APPLE). The program's goal is to develop a set of perception capabilities that are sufficient to enable continuous object learning, where new object instances and categories can be learned from experience in an open-set framework. We have shown preliminary results from initial tests using a motion detection algorithm to delineate objects which are then fed to a simple feed-forward neural network without any other processes in the pipeline. Our neural network trains to an asymptote of acceptable performance and our topology can be defined using our nonparametric model. ARL will continue research to determine the best algorithms to use in the pipeline for the APPLE program.

15. SUBJECT TERMS

perception, robotics, learning, neural networks, motion detection

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Troy Dale Kelley
Unclassified	Unclassified	Unclassified	UU	34	19b. TELEPHONE NUMBER (Include area code) 410-278-5869

Contents

List of Figures

List of Tables

INTENTIONALLY LEFT BLANK.

1. Background and Introduction

1.1 Background

The US Army Research Laboratory's (ARL) Vehicle Technology Directorate (VTD) and the Human Research and Engineering Directorate (HRED), as part of VTD's 6.1 refresh program, have initiated a program called Adaptive Perception Processes for Learning from Experience (APPLE). The program's goal is to develop a set of perception capabilities that are sufficient to enable continuous object learning, where new object instances and categories can be learned from experience in an open-set framework.

HRED's Symbolic and Sub-symbolic Robotics Intelligence Control System (SS-RICS) (Kelley 2006) was inspired by the Adaptive Character of Thought – Rational (ACT-R), cognitive architecture developed by John Anderson at Carnegie Mellon University (Anderson and Lebiere 1998). SS-RICS has been under development since 2004, and since that time numerous algorithms have been added to the robotics architecture. In order to leverage previous SS-RICS development, we have developed a software inventory of the algorithms within SS-RICS that are applicable to the APPLE program (Owens and Osteen 2017). Additional leveraging of previous SS-RICS development is being done to develop a baseline pipeline in order to accomplish the task of a continuous object recognition system. The order and sequencing of algorithms in the pipeline is still subject to research and debate, and those questions will be easier to address following the development of a complete pipeline.

This technical report documents an initial object categorization pipeline by HRED. The work leverages current state-of-the-art work, which combines nonparametric models with neural networks, within an object categorization framework. This work documents an initial HRED pipeline, which will be improved as the APPLE program progresses. In order to document improvements to the pipeline, this manuscript also reports on the comparison of 2 different types of neural networks (Section 5). This algorithm comparison is part of the general pipeline improvement process and will be continued with other algorithms as the APPLE program progresses.

1.2 Introduction

Object recognition, or being able to visually identify and semantically label an object, can be a difficult and challenging problem for computer vision systems. A

quote from an article in *The New York Times* illustrates the history of this endeavor: "In 1966 Professor Minsky gave an undergraduate, Gerald Sussman, the task of building an object-recognition device out of a computer and a television" (Horgan 1997). Sussman did not succeed. The problem turned out to be much more difficult than people had imagined during the early years of artificial intelligence (AI). Humans are so capable of being able to label objects from a variety of viewpoints and distances that people take for granted how complex the task can be, especially for a computer system. To better understand these complexities, Scholl (2001) explored how spatiotemporal features could be more tightly coupled with object representations than surface-based features such as color and shape. In fact, when it comes to human development, Scholl highlights studies that show how 10-month-old infants will use spatiotemporal information, but not featural information, in order to assess an object's unity. Scholl further explains that typically, once an infant reaches 12 months, the infant will begin to use both spatiotemporal and feature information processing for object recognition, which then becomes the persistent interactive object recognition process that carries into adulthood. Replicating this learning/developmental process for a computer system is quite challenging, and will require more work in understanding and characterizing both human learning mechanisms and analogous machine learning algorithms.

In order to begin comparing methods for a baseline pipeline for continuous object recognition within APPLE, we are using motion tracking and detection algorithms from the Open Computer Vision (Open CV) library (Shapiro and Stockman 2001), which are already implemented within SS-RICS. This approach of using motion for object recognition is in contrast to many traditional AI algorithms, which use static images as initial stimuli to a pipeline of algorithms (Everingham et al. 2010). Additionally, the Structure from Motion (SfM) field within the AI community largely focuses on motion for 3-D reconstruction and registration—but not for object recognition per se and not for segmentation (Shapiro and Stockman 2001). We feel that the use of static images in the past has limited the success of object recognition pipelines, and therefore we will emphasize incorporating temporal streams of data for the baseline.

Neural networks have long been used for object recognition and classification tasks. However, neural networks suffer from some inherent problems, including the hand crafting of the topology (Arel et al. 2010) and catastrophic forgetting (McCloskey et al. 1989). Recent work has used hybrid architectures, which include nonparametric models, to overcome some of the limitations of traditional neural networks. These models are then combined with convolutional deep learning networks for the final classification, yielding state-of-the-art results (Vinyals et al. 2016). This hybrid approach has the advantage of allowing for one-shot learning,

with small samples of data, while still being able to have good classification performance and not suffer from catastrophic forgetting.

2. Hypothesized Baseline Pipeline

We hypothesized the APPLE pipeline would require several separate components in order to make up the entire algorithm set capable of object categorization. In its simplest form the pipeline needs to have: 1) an input from the camera, 2) a segmented object from the input, 3) an algorithm that categorizes the objects, and 4) a neural network to classify the objects. These are the basic requirements for the pipeline and are discussed in more detail below.

Additionally, we are interested in testing various parts of the pipeline individually, so we have included tests of different types of neural networks (Section 5). The goal is to be able to substitute different algorithms at various stages of the pipeline. This would allow for optimum performance across the entire pipeline.

Based on these requirements we proposed an initial pipeline (see Fig. 1, Owens and Osteen 2017) as part of the 6.1 refresh process. Figure 1 represents the core set of components that are hypothesized to be sufficient to enable continuous object learning. In the project proposal, we state that our central hypothesis is that continuously learning objects from experience requires mechanisms to do the following:

1) Focus attention on things and stuff of interest.
2) Track ego and object-motion for object permanence and common fate.
3) Convert raw data into rich internal instance representations.
4) Detect when a previous instance or category has been observed.
5) Detect when a novel instance has been observed.
6) Generalize instance representations into category representations with little or no supervision.

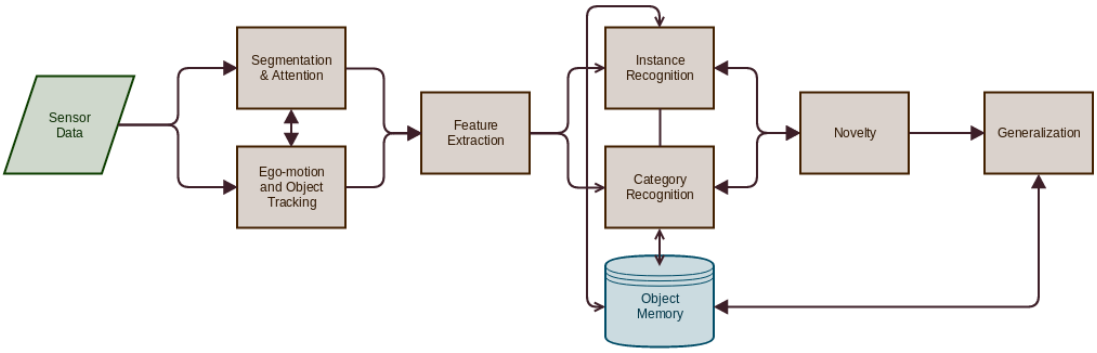

Fig. 1 Core pipeline hypothesized to be sufficient to enable continuous object learning

Figure 2 shows the intermediate pipeline being tested for this report. The intermediate pipeline in Fig. 2 is the first pass, or initial step, in the process of developing a more complete pipeline as defined by Fig. 1. Many of the algorithms used in Fig. 2 were already within SS-RICS and allowed us to leverage these algorithms to quickly begin testing rapidly.

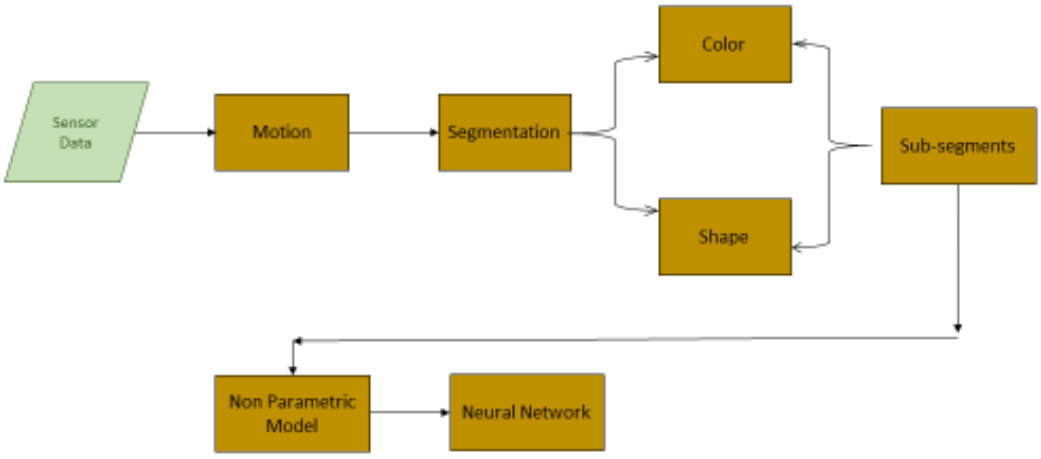

Fig. 2 Potential cognitive-based model to be tested as an alternate approach to the model in Fig. 1

As part of the APPLE program, we have outlined an informal justification for the necessity of each mechanism:

1) Given a raw data stream and the need for a subset of that data, some kind of *segmentation, attention, or object discovery* mechanism is required to retrieve the regions of interest (ROIs).

2) If we do not ignore agent or object motion, then a *tracking facility* must exist to provide data association between successive "frames" of raw data;

this is especially important if the algorithms require processing periods longer than framerate.

3) Given a subset of raw data representing ROIs, instance, category, or novelty detection would not be possible without extracting *some representation suitable for classification*. In addition, it must support storage for recognition at a later time, and be amenable for generalization of instances to categories.

4) Given a stored representation of instances and categories, recognizing a newly observed and described ROI requires some kind of classification mechanism.

5) Without a novelty detection mechanism, there would be no way for the classifier to properly output "unknown instance of class". This is actually a major limitation of almost all popularly used machine learning classification algorithms, and is an instance of the multi-class open-set recognition problem (Scheirer et al. 2014).

6) Finally, a mechanism to derive categories from instances is required, since categories cannot be experienced directly by any embodied system (Owens and Osteen 2017).

3. Intermediate Pipeline

Figure 2 shows the current state of a baseline pipeline, using the SS-RICS cognitive architecture as the pipeline's underlying foundation. The current approach can be summarized as follows:

1) Sensor data: image color (red, green, blue [RGB]) data for the initial baseline is being used. Depth data using point cloud information for shape recognition will be added later, either from stereo cameras, RGB-D cameras, or laser detection and ranging data. For this report, however, we use image RGB data only.

2) The segmentation/attention and ego-motion and object tracking are in parallel for the pipeline in Fig. 1, but the current pipeline uses motion itself as a segmentation cue, and therefore segmentation directly follows from object motion and tracking. The core of the motion detection algorithm is a background subtraction process that yields foreground regions in the presence of object or ego-motion. While solid objects will produce motion detections and yield the possibility of feature extraction, motion can also be detected due to lighting changes in the background that do not correspond

to an object (i.e., part of a wall). Also, during ego-motion, most of the environment will change from the sensor's point of view, triggering motion detection even for static objects. We are exploring refinements to this process that will ameliorate this side effect.

3) SS-RICS has already implemented saliency algorithms to control attention (Itti et al. 1998), but these algorithms are prohibitively slow for the APPLE program. We are constructing a goal stack for SS-RICS to control the focus of attention, and while this work emphasizes the speed of attention shifts, the work is still under development.

4) The first 2 processes are followed by feature extraction, where we are using primarily shape and color. The shape algorithm work is not currently part of the initial baseline pipeline. The second feature, color, is extracted from the image data and we are exploring ways to optimize the color data analysis.

5) The instance and categorical recognition is currently being done by a nonparametric model. Our nonparametric model uses a Pearson's Product Moment Correlation to sort the images found by the motion detection algorithms into visually correlated categories. These highly correlated categories are then used as input to a neural network, where the output size of the neural network is determined by the number of categories used during training. We are still refining the number of categories that are then used to train the neural network, but for this report networks are trained on 3–5 categories. The motion detection can give us too many categories, but we address this problem with a size limitation variable during the image input. In other words, we only look for images that are greater than a certain size for possible inclusion as a category. The nonparametric model also does not currently function in real time, but as a post hoc sorting process, and we are working to include this as a real-time process in the pipeline.

6) While Fig. 1 shows the novelty being done after the categorical recognition, for this work it is being done before the neural network classification, by the nonparametric model. The nonparametric model allows us to determine if we have existing categories, where categories are not semantic level, but rather groups of images that are highly correlated with each other and therefore visually similar. We recognize that this may yield multiple categories for a single semantic object depending on viewpoint changes, and that many images in a category consist of consecutive frames of the same image region. However, our goal is to not feed the network an instance of what we would consider a new category. The nonparametric model

allows us to make this distinction. We also investigated the use of Adaptive Resonance Theory (ART) neural networks (Carpenter et al. 1991), which are already implemented in SS-RICS, and can also account for the new category problem, but we decided to use simple feed-forward networks as an initial implementation because we had more control over the code.

7) Finally, generalization is still an open research question for this project and will be defined more fully as the research progresses.

4. Implementation Overview and Nonparametric Model

4.1 Overview

Feature extraction for classification is a difficult research area within computer vision. Many techniques (i.e., neural networks or principle components analysis) are sensitive to artifacts in the sensor data (e.g., shadows, reflections), which reduce the classification performance. Controlling the feature extraction process to improve feature stability and robustness in various environments is one of the goals of APPLE.

It is well known that humans use color and shape as primary features for determining the identity of objects, so we are focusing on those 2 features to begin. Additionally, while a novelty detection algorithm for episodic memories has been previously studied (Kelley and McGhee 2013), the algorithm is probably not appropriate for the detection of new information that has not already been learned by a neural network. Instead, we are using the nonparametric model for such determinations.

4.2 Nonparametric Model

The nonparametric model was implemented to aid in the training process of a neural network, and we will detail our nonparametric model in Fig. 3.

Our nonparametric model works in 5 stages: motion capture, image resizing; cross-correlation/color correlation of images; sorting candidates into categories; and training the neural net based on the topology defined by the nonparametric sort.

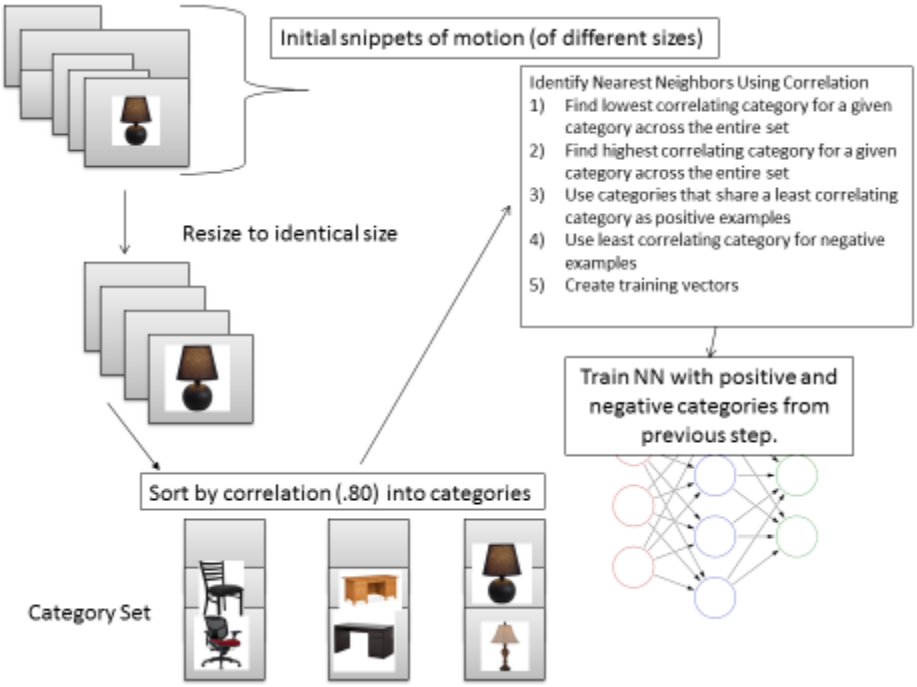

Fig. 3 Nonparametric model showing 5 stages: 1) motion collection of objects; 2) resizing; 3) sorting into correlations; 4) identification of nearest neighbors by correlation; and 5) final neural network training

Stage 1

We are using motion detection to determine object candidates, reducing the complexity of the segmentation problem (see Fig. 4).

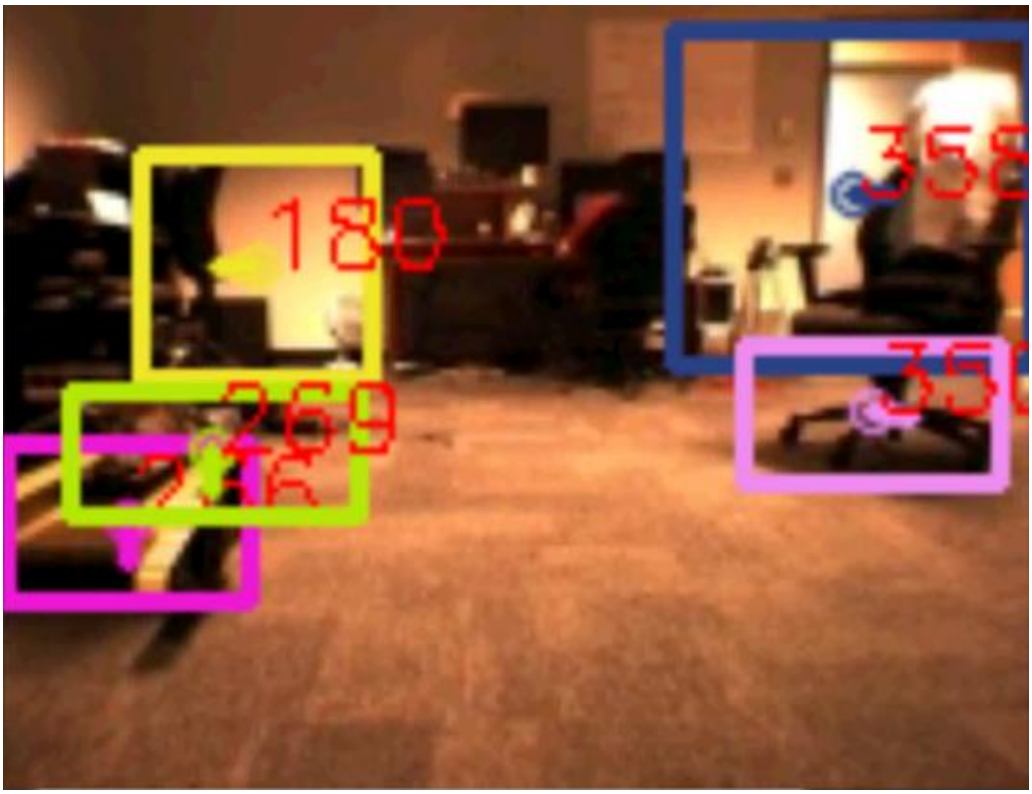

Fig. 4 Screen shot of robot moving through a laboratory. The 5 differently colored boxes indicate possible objects that were acquired by the motion detection algorithm: 1) upper right in blue: top of chair, 2) lower right in pink: base of chair, 3) upper left in yellow: wall, 4) lower left in green: base of chair, 5) bottom left in pink: robot. The numbers in red indicate the angle from the center point of the bounding box.

First, object motion images are gathered by the motion detection process. They are then normalized in size and shape to be cross correlated. The size is arbitrary, which we have set to be 50 × 50 currently. This number can be any number desired or set to the maximum or average size of the images gathered.

Stage 2

Once stage 1 is finished, all the images are correlated using 2 methods: Pearson's Product Moment Correlation (accomplished on a gray-scale of the images) and color histogram comparisons. In each case, we do a sequential comparison of one image to the rest; selecting the highest correlations and then removing all those from the image pool into their own categories. We then proceed recursively through the remainder of images until all have been processed.

Stage 3

The highest correlations are placed in a category and the next highest in another category, and so forth, down to the lowest in the correlation matrix. As each image

is placed in its category, a vector is created by converting RGB to HSV color space to allow for a pure color extraction of the pixel data using the previously mentioned DSpiral algorithm. Each image's pixel vector is written to a file, which is then placed in a category as well. This vector is used to train the neural net.

We then select the highest correlations from a couple of different categories as the positive categories for training the neural net, and correlations around zero with the positive categories for the requisite negative training examples. The resultant training curve (Fig. 5) shows that the error (red curve) begins at around 0.1, and does not deviate throughout the training, demonstrating that error was small. Also, the output (green curve) begins low, but improves over time, to a stable value of around 0.8, indicating that training was relatively successful.

The definitions and equations that produced these curves can be found in Appendix A.

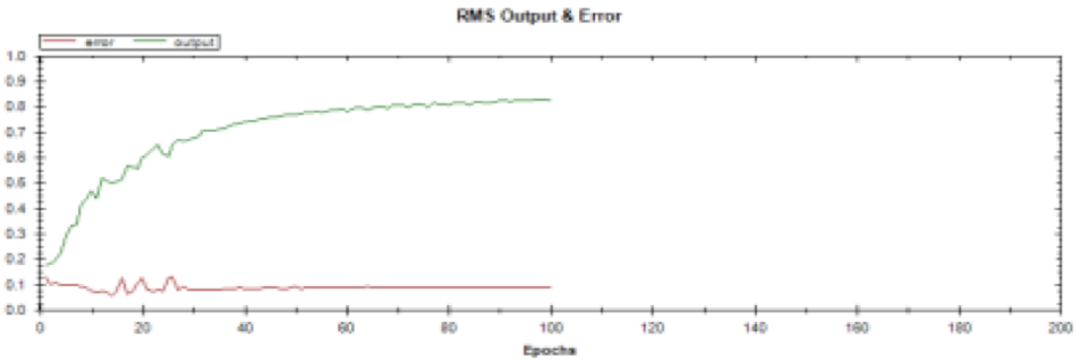

Fig. 5 Sample neural network output trained to asymptotic performance with 5 categories collected from a moving robot. The green line is the learning curve and the red line is the error rate. In this case, our neural network trained to an acceptable level of about 80 percent. This graph was produced with a feed-forward neural network.

5. Comparison of ARTMap and Feed-Forward Neural Network

A goal of APPLE is to use state-of-the-art algorithms for each component within the pipeline. In order to test various components of the pipeline, we tested 2 different types of neural networks against each other. We compared the Simplified Fuzzy ARTMap neural network (Vakil-Baghmisheh and Pavešić 2003) with a feed-forward neural network (see Fig. 6).

Our comparison proceeded as follows:

- Collect images through motion detection (bounding boxes returned from the motion algorithm).

- Resize the images to the same size, in this case 50 × 50 pixels for a total of 2,500.
- Sort the images into categories using a Pearson's Product Moment Correlation on each jpg (gray scaled) scoring 0.8 or higher. Visual inspection showed the images in each category to be fairly homogenous (see Appendix B).
- Extract the pixels using DSpiral from each image in each category and collect them in a file for training the neural nets. Each category then has a file containing the vector extracted from each image.
- Cross correlate categories using the same Pearson's Correlation to find a common "negative" category that scored as close to 0.0 as possible and "nearest neighbors" for each category to use in classification.
- All of the categories that shared a common negative are then used as positive categories against the negative for a training run.
- Limit positive categories to those that contain 4 or more vectors as positive training sets.
- Trained both of the neural nets using the same categories for positive classifications. Note that the ARTMap generates its own negative exemplars, whereas the feed-forward network does not and requires the user to provide them.
- Ran the training data one vector at a time through each neural net and collected the scores for each and tallied the percentage correct for each category and did the same with the nearest neighbor data.
- Tallied the overall percentage correct for each training session and compared the 2 sets of runs.

The nearest neighbors were not cross correlated between themselves and were also not trained. They were used as a "sanity check" since our vector sets were not large enough to withhold positive training data. We have since conducted another run on data collection, which yielded adequately large enough results to do this, and those results are illustrated in Appendix B.

Method	Run										result (% correct)
	50_339	0_347	30_169_241_293	179_233_283	152_340	39_73	18_107	192_234	76_230_270	305_329	
ARTMap	76	92	72	83	100	100	49	50	83	75	78
Netbuilder	100	92	41	62	100	100	52	100	62	75	78.4

Fig. 6 Row labels are each neural network type. Column headers are category numbers trained as positive examples (each category number is separated by underscores) and the results for an outside-the-training-set classification (as percent correct). Results are essentially equal (78 vs. 78.4 – last column) for the ARTMap and Netbuilder (feed-forward neural network).

6. Conclusions

The baseline pipeline continues to be developed, but the initial SS-RICS-based implementation has been described here. The methodology described here has 3 important advantages over the traditional neural network classification of images.

First, the nonparametric front end to the neural network removes the problem of hand crafting the topology of the network before training. The nonparametric model determines the number of categories for training to the network without programmer interference. Second, the use of motion detection gives us "objectness" in a computationally efficient manner (real time using a monocular camera). Third, if additional categories are discovered by the nonparametric model, the network can be retrained autonomously, without any additional input from a programmer. This allows for continuous adaptation over time. Finally, the neural network comparison showed that the ARTMap network was equal to a feed-forward network for out-of-training-set classification. Additionally, the ARTMap claims to not suffer from catastrophic forgetting, in which case it would be the more advantageous choice for the classification network.

We have shown preliminary results from initial tests using a motion detection algorithm to delineate objects which are then fed to a simple feed-forward neural network without any other processes in the pipeline. Our neural network trains to an asymptote of acceptable performance and our topology can be defined using our nonparametric model. ARL will continue research to determine the best algorithms to use in the pipeline for the APPLE program.

7. References

Anderson JR, Lebiere C. The atomic components of thought. Mahwah (NJ): Lawrence Erlbaum Associates; 1998.

Arel I, Rose DC, Karnowski TP. Deep machine learning—a new frontier in artificial intelligence research [research frontier]. IEEE Comput Intell Mag. 2010;5(4):13–18.

Carpenter GA, Grossberg S, Reynolds JH. ARTMAP: supervised real-time learning and classification of nonstationary data by a self-organizing neural network. Neural Networks. 1991;4(5):565–588.

Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A. The pascal visual object classes (VOC) challenge. Int J Comput Vision. 2010;88(2):303–338.

Horgan J. Smarter than us? Who's us? The New York Times. 1997 May 4.

Itti L, Koch C, Niebur E. A model of saliency-based visual attention for rapid scene analysis. IEEE Trans Pattern Anal Mach Intell. 1998;20(11):1254–1259.

Kelley TD. Developing a psychologically inspired cognitive architecture for robotic control: The symbolic and sub-symbolic robotic intelligence control system. Int J Adv Rob Syst. 2006;3(3):32.

Kelley TD, McGhee S. (2013) Combining metric episodes with semantic event concepts within the symbolic and sub-symbolic robotics intelligence control system (SS-RICS). Proceedings of SPIE Defense, Security, and Sensing; 2013 Apr 29–May 3; Baltimore, MD. Bellingham (WA): Society of Photo-Optical Instrumentation Engineers (SPIE). p. 87560.

McCloskey M, Cohen N. Bower GH, editor. Catastrophic interference in connectionist networks: The sequential learning problem. Amsterdam (Netherlands): Elsevier Inc.; 1989. (The psychology of learning and motivation; vol. 24). p. 109–164.

Owens J, Osteen P. Ego-motion and tracking for continuous object learning: a brief survey. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2017 Sep. Report No.: ARL-TR-8167.

Scheirer WJ, Jain LP, Boult TE. Probability models for open set recognition. IEEE Trans Pattern Anal Mach Intell. 2014;36(11):2317–2324.

Scholl BJ. Objects and attention: the state of the art. Cognition. 2001;80(1):1–46.

Shapiro LG, Stockman GC. Computer vision. Upper Saddle River (NJ): Prentice Hall; 2001.

Vakil-Baghmisheh MT, Pavešić N. A fast simplified fuzzy ARTMAP network. Neural Process Lett. 2003;17(3):273–316.

Vinyals O, Blundell C, Lillicrap T, Wierstra D. Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R, editors. Matching networks for one shot learning. Advances in neural information processing systems 29 (NIPS 2016). Proceedings of Neural Information Processing Systems 2016 (NIPS 2016); 2016 Dec 5–10; Barcelona, Spain. p. 3630–3638.

Appendix A. Epoch Graph Produced by Neural Net

Figure A-1 shows a graph produced by our neural net, which we call Netbuilder, after training 3 outputs over 100 epochs.

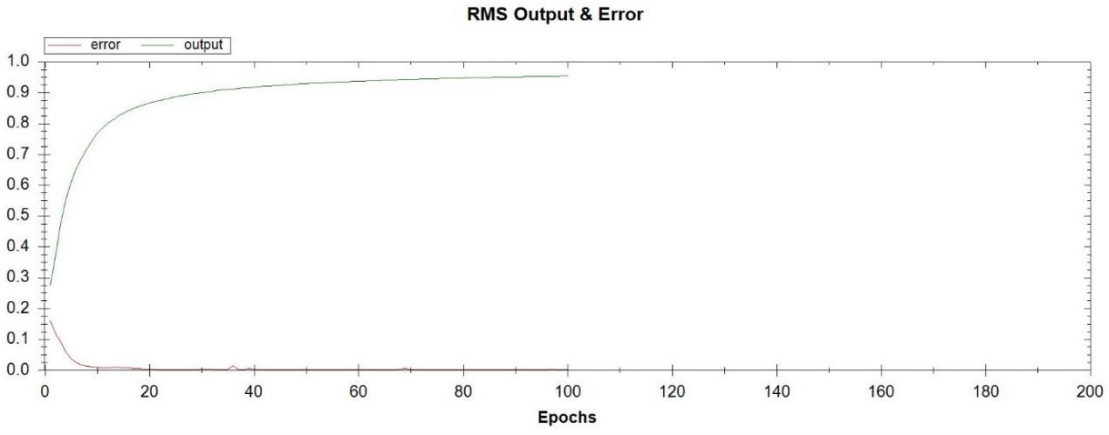

Fig. A-1 Adequate training accomplished

The values in the output graph (in green and above the error graph) are produced by summing the squares of the output values of each layer's propagation (which is its calculated percentage correct) and taking the square root of their mean (root mean square):

Let n = the number of output layers
Let t_i = the output value of each layer's current epoch
Let m = the number of epochs generated
Let p_j = each output point plotted on the graph

Thus

$$for\ (j = 1, m)\ \ p_j = \sqrt{\frac{\sum_{i=0}^{n} {t_i}^2}{n}} \qquad (1)$$

Each value for the error graph is computed taking the sum of the squares of the target; the output of each layer divided by 2:
Let n = the number of output layers
Let a_i = the output value of each layer's current epoch (probability that it belongs to the corresponding output category)
Let t_i = the target value of each layer's current epoch
Let m = the number of epochs generated
Let p_j = each output point plotted on the graph

Thus

$$for\ (j = 1, m)\ \ p_j = \frac{\sum_{i=0}^{n} (t_i - a_i)^2}{2} \qquad (2)$$

The output graph is used to determine whether or not the net has been trained satisfactorily. If the graph is smooth and trends (logarithmically) to 1 from 0 then our assessment is that the training was successful. Deviations from this indicate less-than-optimal training, but possibly still acceptable given the quality of the data obtained.

The error graph is less important until the output graph becomes noticeably irregular and/or not trending toward 1 (which denotes 100% success). But, at such a point, the training is deemed unsatisfactory and the training data are discarded.

Figure A-1 shows a "good" graph. Figures A-2 and A-3 show an "acceptable" graph and a "rejectable" graph, respectively.

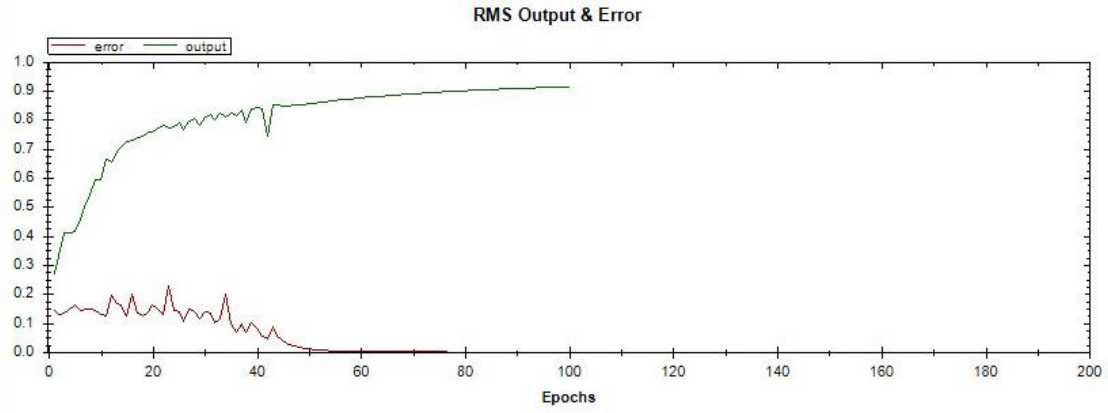

Fig. A-2 "Acceptable" graph

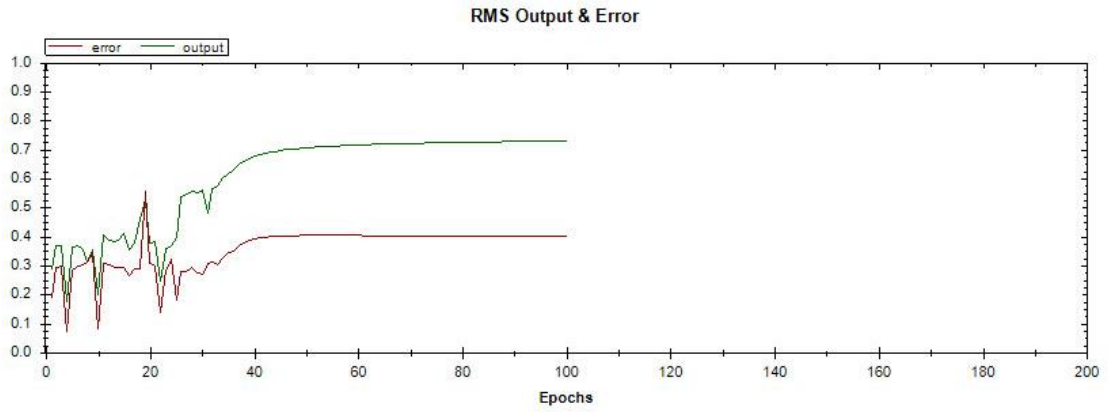

Fig. A-3 "Rejectable" graph

INTENTIONALLY LEFT BLANK.

Appendix B. Test of Efficacy of Categorization through Correlation Clustering

The clustering mechanism uses a Pearson's Correlation (with an acceptance threshold of 0.85) on the resized images harvested during motion detection. This mechanism is not perfect, but in most cases results in a fairly homogenous set of individual categories suitable for training a neural net. In order to provide a check after visual inspection of the categories, a test was run on a set of 3 categories, which were cross correlated to find a common orthogonal category.

Table B-1 shows which categories correlated least and most with each other, and illustrates the categories chosen for this test.

Table B-1 Spreadsheet generated to identify categories

Run	Cat	Vecs	Least	Corr	Most	Corr
all	746	5	55	–0.09997	750	0.790639
all	1078	26	55	–0.09979	952	0.796281
all	1736	5	55	–0.09993	1738	0.775841

The categories (Cat) chosen as positive examples are 746, 1078, and 1736 with 5, 26, and 5 vectors each, respectively—and the negative (least correlated) category is 55 with 3 vectors. Due to the nature of our collection mechanism, small numbers of vectors in each category can be quite common.

We removed about one fifth of the vectors from the largest category, 1078, and trained the net on the categories mentioned above with the resultant root-mean-square and error curves depicting success (illustrated in Figs. B-1 through B-11).

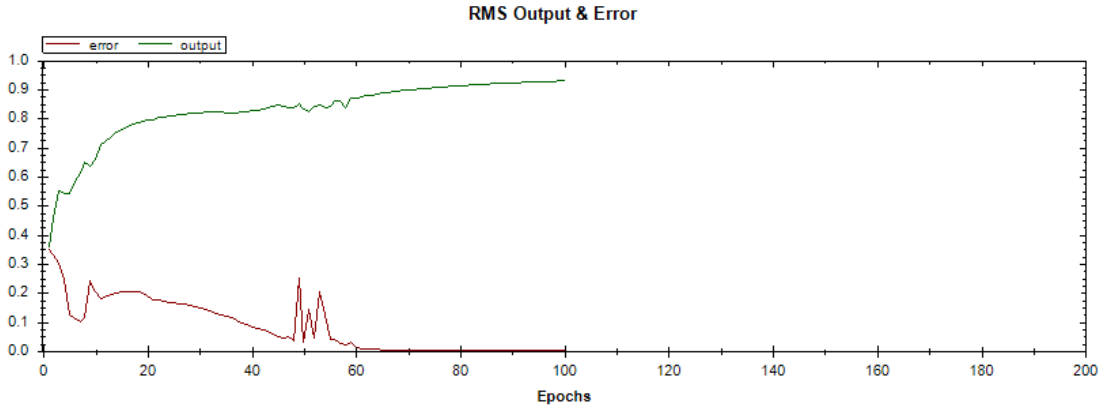

Fig. B-1 Acceptable training for the neural network using the categories in Table 1

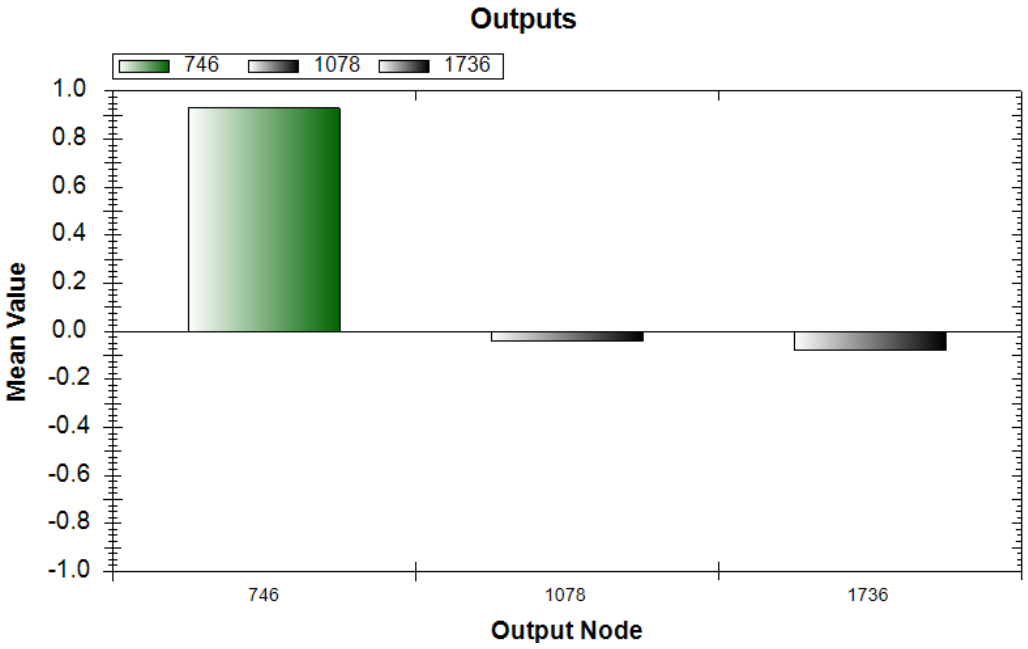

Fig. B-2 Results for training set 746

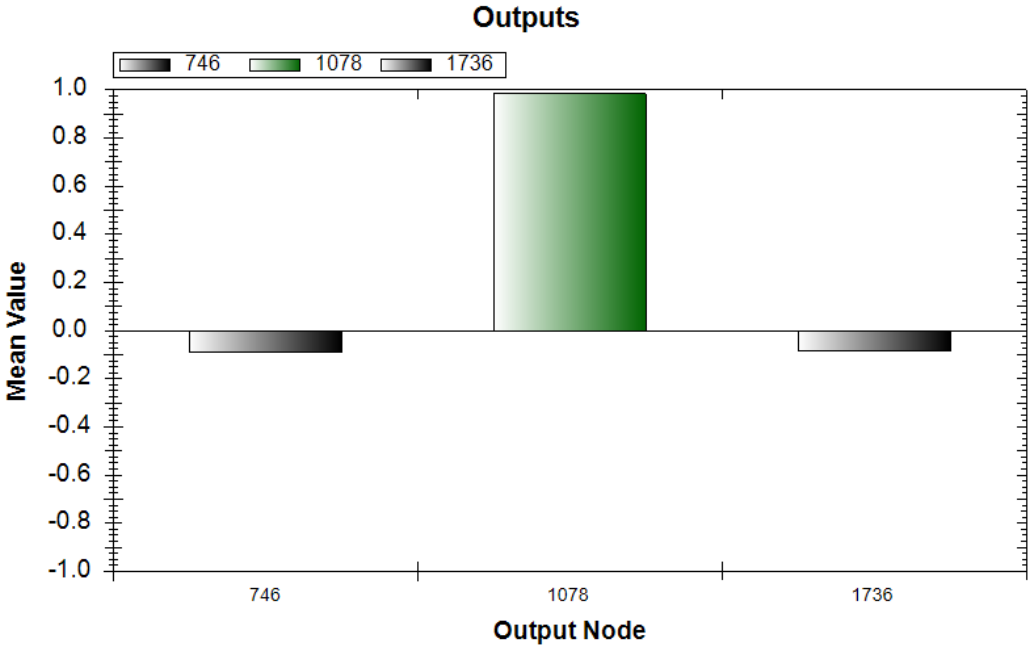

Fig. B-3 Results for training set 1078 (minus withheld vectors)

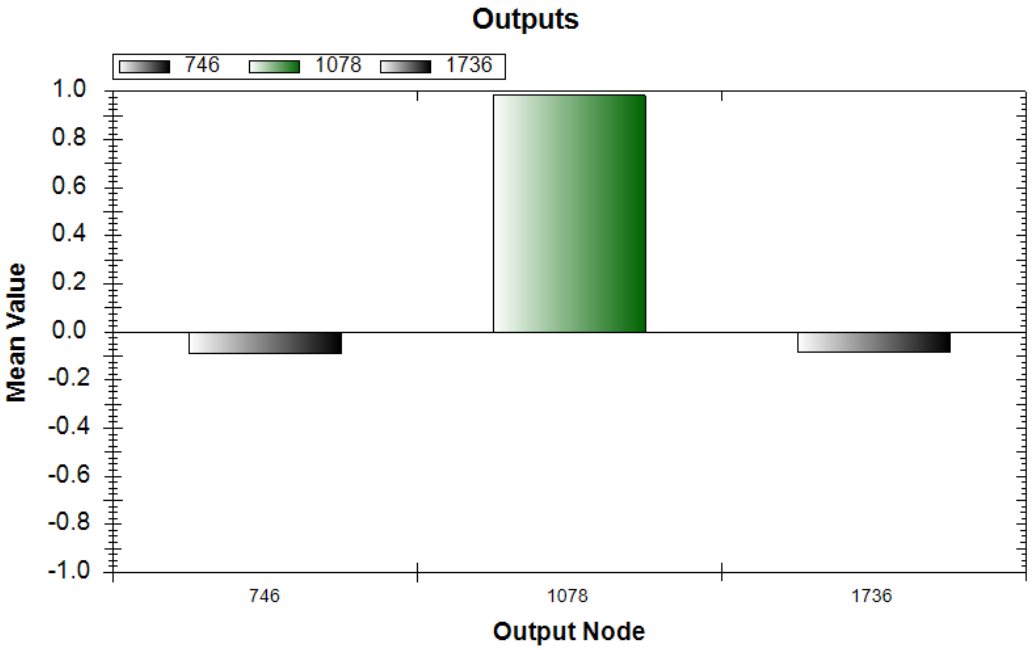

Fig. B-4 Results for training set 1078 (using withheld vectors)

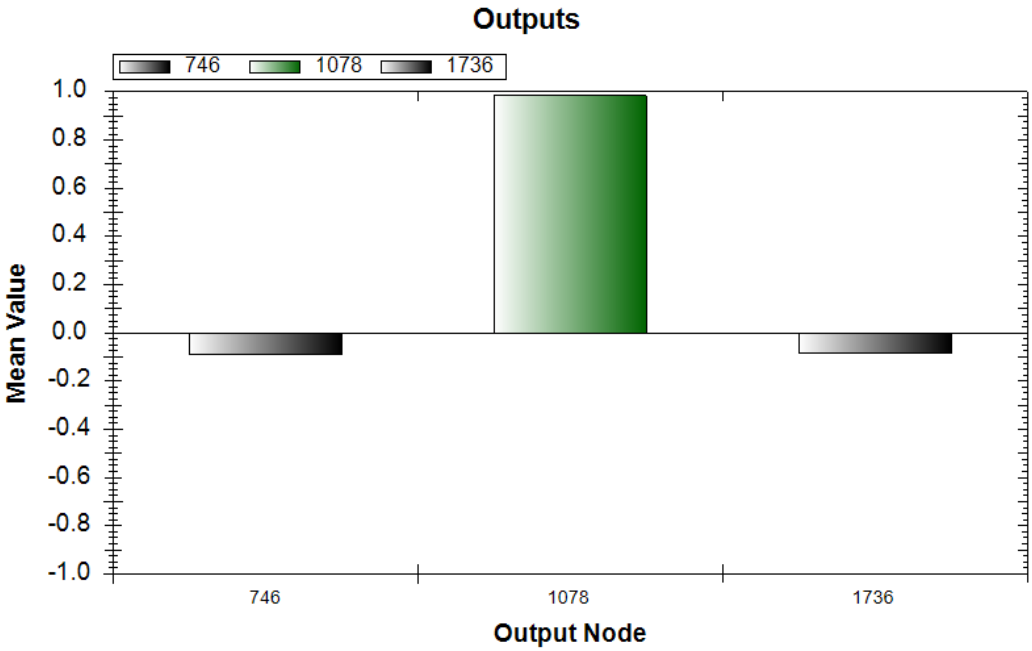

Fig. B-5 Results for training set 1078 (all vectors included)

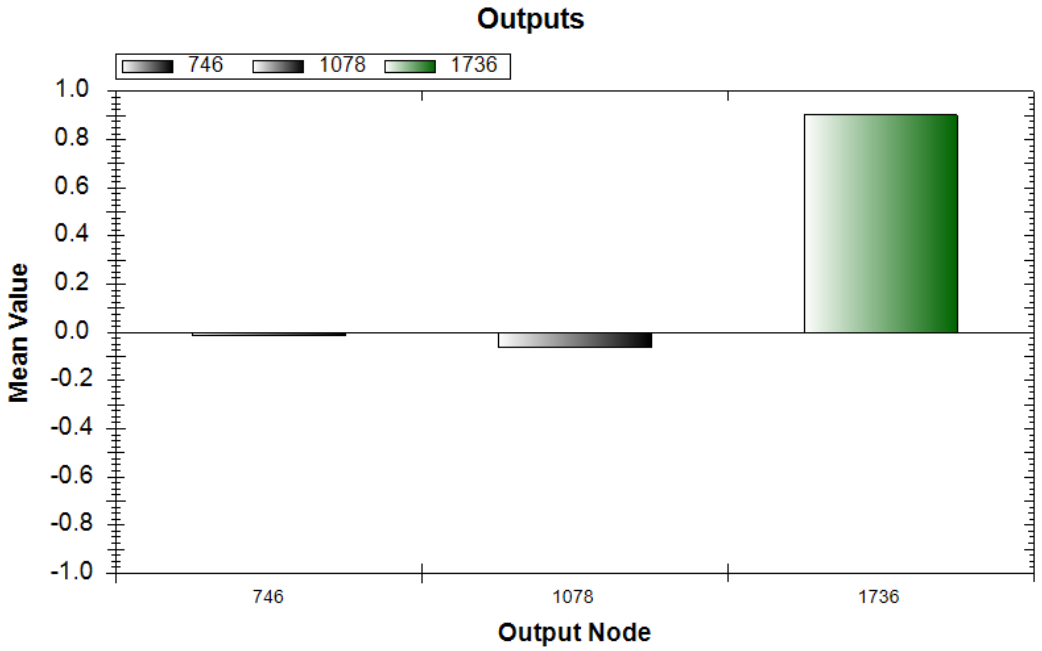

Fig. B-6 Results for training set 1736

The training sets are as follows (note that the images might be uninterpretable due to resizing).

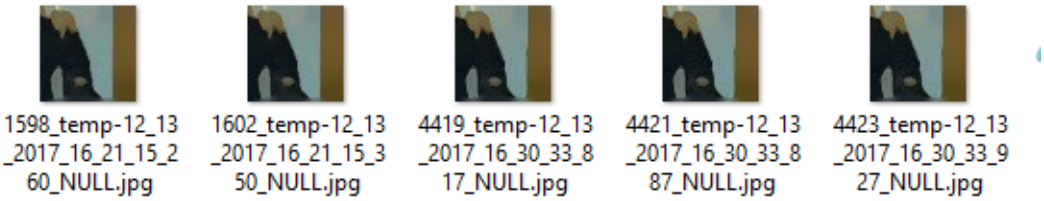

Fig. B-7 Training set 746

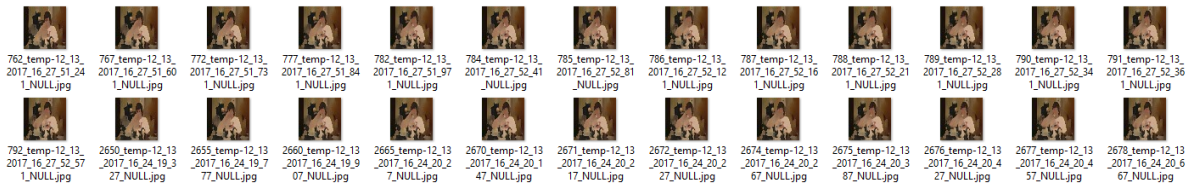

Fig. B-8 Training set 1078 (all shown; 5 were withheld for test)

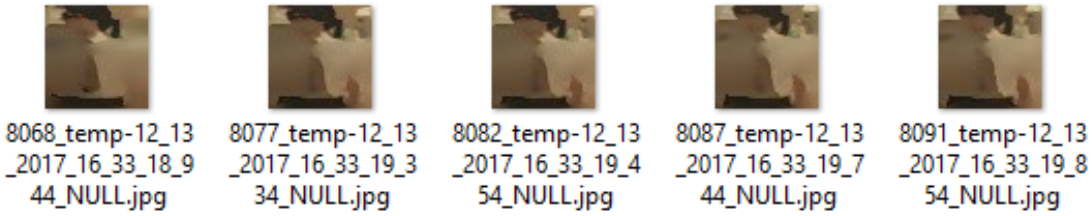

Fig. B-9 Training set 1736

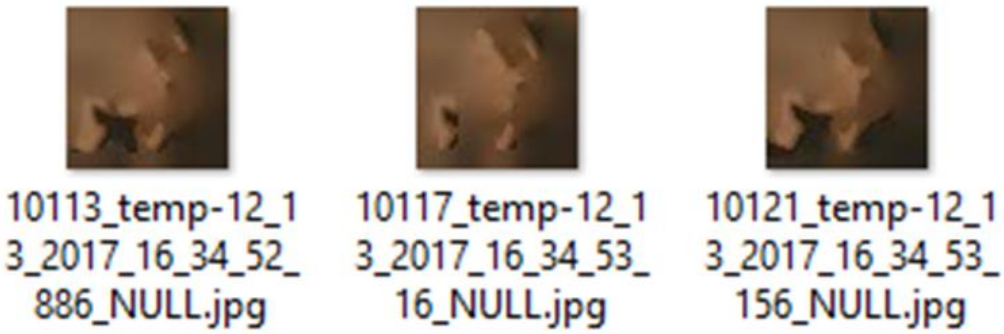

Fig. B-10 Training set 55 (negative category)

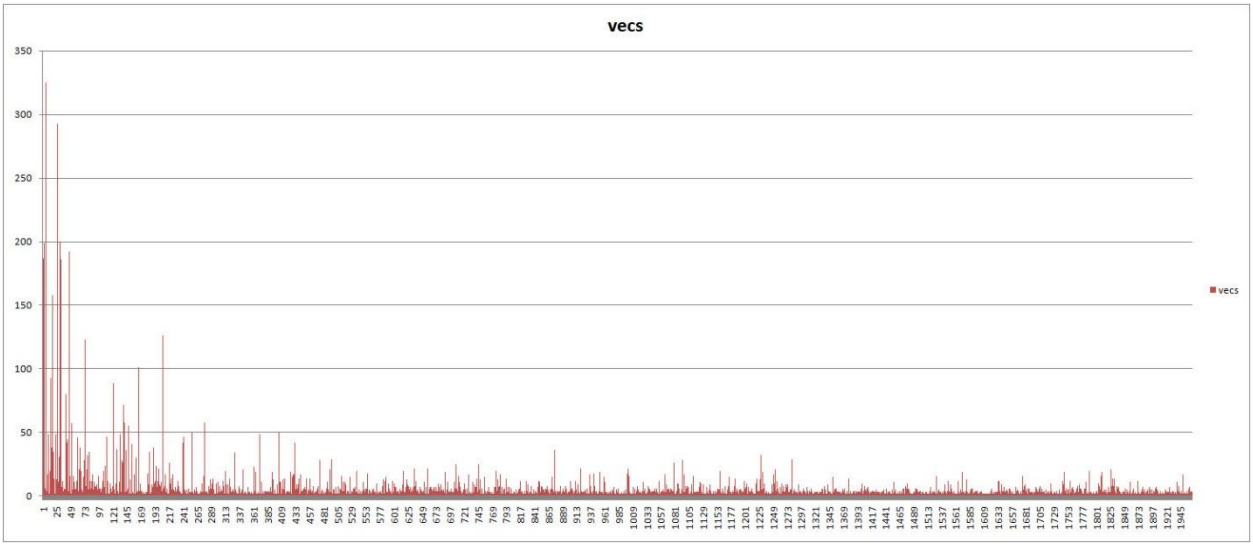

Fig. B-11 The distribution of vectors per category for the run that produced the results; 1,964 categories were produced

List of Symbols, Abbreviations, and Acronyms

3-D	3-dimensional
ACT-R	Adaptive Character of Thought – Rational
AI	artificial intelligence
APPLE	Adaptive Perception Processes for Learning from Experience
ARL	US Army Research Laboratory
ART	Adaptive Resonance Theory
HRED	Human Research and Engineering Directorate
Open CV	Open Computer Vision
RGB	red, green, blue
ROI	region of interest
SfM	Structure from Motion
SS-RICS	Symbolic and Sub-symbolic Robotics Intelligence Control System
VTD	Vehicle Technology Directorate

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

2 DIR ARL
(PDF) IMAL HRA
RECORDS MGMT
RDRL DCL
TECH LIB

1 GOVT PRINTG OFC
(PDF) A MALHOTRA

1 ARL
(PDF) RDRL HRB B
T DAVIS
BLDG 5400 RM C242
REDSTONE ARSENAL AL
35898-7290

8 ARL
(PDF) SFC PAUL RAY SMITH CENTER
RDRL HRO COL H BUHL
RDRL HRF J CHEN
RDRL HRA I MARTINEZ
RDRL HRR R SOTTILARE
RDRL HRA C A RODRIGUEZ
RDRL HRA B G GOODWIN
RDRL HRA A C METEVIER
RDRL HRA D B PETTIT
12423 RESEARCH PARKWAY
ORLANDO FL 32826

1 USA ARMY G1
(PDF) DAPE HSI B KNAPP
300 ARMY PENTAGON
RM 2C489
WASHINGTON DC 20310-0300

1 USAF 711 HPW
(PDF) 711 HPW/RH K GEISS
2698 G ST BLDG 190
WRIGHT PATTERSON AFB OH
45433-7604

1 USN ONR
(PDF) ONR CODE 341 J TANGNEY
875 N RANDOLPH STREET
BLDG 87
ARLINGTON VA 22203-1986

1 USA NSRDEC
(PDF) RDNS D D TAMILIO
10 GENERAL GREENE AVE
NATICK MA 01760-2642

1 OSD OUSD ATL
(PDF) HPT&B B PETRO
4800 MARK CENTER DRIVE
SUITE 17E08
ALEXANDRIA VA 22350

ABERDEEN PROVING GROUND

12 ARL
(PDF) RDRL HR
J LOCKETT
P FRANASZCZUK
K MCDOWELL
K OIE
RDRL HRB
D HEADLEY
RDRL HRB C
J GRYNOVICKI
RDRL HRB D
C PAULILLO
RDRL HRF A
A DECOSTANZA
RDRL HRF B
A EVANS
RDRL HRF C
J GASTON
RDRL HRF D
A MARATHE
T KELLEY