Visual Task Recognition for Human-Robot Teams

Prakash Baskaran
Collaborative Robotics and Intelligent Systems Institute
Oregon State University, Corvallis, OR
baskarap@oregonstate.edu

Joshua Bhagat Smith
Collaborative Robotics and Intelligent Systems Institute
Oregon State University, Corvallis, OR
bhagatsj@oregonstate.edu

Julie A. Adams, *Senior Member, IEEE*
Collaborative Robotics and Intelligent Systems Institute
Oregon State University, Corvallis, OR
julie.a.adams@oregonstate.edu

2022 IEEE 3rd International Conference on Human-Machine Systems (ICHMS) | 978-1-6654-5238-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICHMS56717.2022.9980659

***Abstract*—Human teammates in human-robot teams operate in uncertain, dynamic environments to accomplish a wide range of tasks. These tasks often involve multiple activity components: gross motor, fine-grained motor, tactile, cognitive, visual, speech and auditory. Most existing task recognition algorithms focus primarily on detecting tasks involving gross and fine-grained motor components; however, some tasks (e.g., assessing a victim's triage level) may involve little to no motor components. Robots need a holistic understanding of a task's various activity components in order to be aware of the human's current task state. The presented algorithm detects the tasks' visual activity component for a human-robot team operating in a non-sedentary supervisory environment. Metrics acquired from a wearable eye tracker and head motion tracker are used to train the machine learning-based visual task recognition algorithm.**

***Index Terms*—Visual task recognition, Human-robot interaction, Human-machine teams, Wearable sensors**

I. Introduction

Human-robot teams (HRTs) collaborating to achieve tasks under various conditions, especially in unstructured, dynamic environments, will require robots to adapt autonomously to a human teammate's state (e.g., workload level). An important element of such adaptation is the robot's ability to infer the human teammate's current tasks, as understanding human actions and their interactions with the world provides the robot with more context as to what type of assistance the human may need. Environmentally embedded sensors, such as cameras (e.g., motion capture, depth) are infeasible for task recognition in unstructured environments; however, employing wearable sensors in such environments is a viable alternative.

Human teammates perform a wide variety of tasks using a breadth of capabilities. Depending on task complexity, multiple activity components can be involved: gross motor, fine-grained motor, tactile, cognitive, visual, speech and auditory. For example, the triaging a victim task aggregates cognitive, visual and possibly speech and auditory (if the victim is conscious) components in order to assess the victim's triage level prior to taking any necessary medical steps (e.g., applying a tourniquet or performing cardiopulmonary resuscitation).

Most existing task recognition algorithms focus primarily on detecting tasks involving physical movements; however, some tasks are more dependent on other activity components and involve limited physical movement. Robots need a holistic understanding of a task's various activity components in order to detect the human's current tasks accurately. This manuscript's primary contribution is an algorithm for recognizing visual tasks in a non-sedentary environment using a wearable eye tracker and a head worn motion tracker. The algorithm is evaluated by identifying the visual tasks performed by HRTs operating in a non-sedentary supervisory environment.

II. Related Work

Eye movement is closely associated with humans' goals, tasks, and intentions, as almost all tasks performed by visually unimpaired humans involve visual observation. This association makes oculography a rich source of information for task recognition. Fixation, saccades, blink rate, and scanpath are the most commonly used metrics for detecting visual tasks [1]–[3], followed by electrooculography signals [4]–[6].

Classical machine learning using eye gaze metrics (e.g., saccades, fixation, and blink rate) for visual task recognition was pioneered by Bulling et al. [1]. Statistical features extracted from the gaze metrics, as well as the character-based representation to encode eye movement patterns, were used to train a Support Vector Machine (SVM) classifier to detect five office-based tasks. Various visual task recognition algorithms were developed focusing solely on detecting reading tasks using classical machine learning (e.g., [7], [8]). The complexity of the "reading" task varied from as rudimentary as detecting reading or not [8], to as complex as distinguishing between reading thoroughly or skimming through the text [7].

Other algorithms apply deep learning using electrooculography potentials to detect visual tasks [6], [9]. Two deep networks, a convolutional neural network and a long short-term memory, recognized reading in a natural setting (i.e., outside the laboratory). Three metrics (i.e., blink rate, 2-channel electrooculogram signals, and acceleration) from wearable glasses were used to train the deep learning models. The algorithms detected common visual tasks in *office or desktop-based* environments with sedentary participants. None of existing algorithms detected the visual tasks within context of HRTs, where the human teammates can be in constant motion, operating in an uncertain, dynamic environment.

III. Methodology

The supervisory task environment consisted of a modified version of the NASA Multi-Attribute Task Battery (MATB-II) [10], which required a human operator to supervise a simulated remotely-piloted aircraft. The NASA MATB-II consists of four composite tasks: tracking, system management, resource management, and communication monitoring. These composite tasks are composed of multiple atomic tasks and activity components; however, this manuscript only focuses on detecting the composite tasks' visual component.

A. Experimental Design

The original NASA MATB-II required participants to remain stationary, but real-life HRT scenarios require movement throughout the environment. The NASA MATB-II was modified to physically separate the composite tasks; thus, requiring participants to walk between the tasks, as depicted in Figure 1. Each NASA MATB-II tasks' dedicated computer monitor was stationed such that the participants were unable to visually see more than two composite tasks simultaneously, ensuring that participants walked around to complete the tasks.

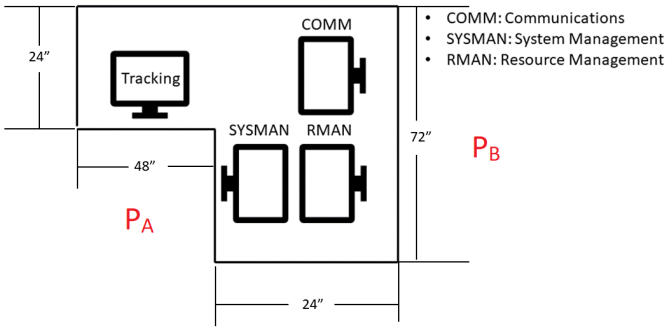
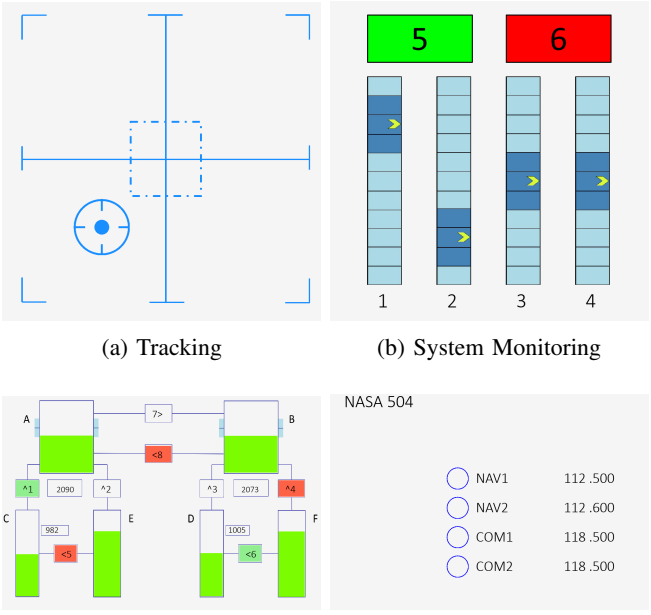


Fig. 1: Physical layout of the modified NASA MATB-II. NOTE: P_A and P_B are the points between which participants walked back and forth in order to complete the tasks.

The tracking composite task (Figure 2a) required participants to keep the circle with a blue dot in the middle of the cross-hairs using a joystick. This composite task's visual component involves *visually tracking* the target.

The system monitoring composite task (Figure 2b) required monitoring two colored lights and four gauges. The gauges' indicator randomly moved up and down, typically remaining in the middle. The green (L5) and the red light (L6) turned on whenever the value was too high or low and required resetting. The lights and gauges were reset by pressing the corresponding number key on the top row of a keyboard. The system monitoring task entails a *visual inspection* task.

The resource management composite task (Figure 2c) included six fuel tanks (A-F) and eight fuel pumps (1-8). The arrow by the fuel pump's number indicated the direction fuel was pumped. Participants were to maintain the fuel levels of Tanks A and B by turning the fuel pumps on or off. Fuel Tanks C and D had finite fuel levels, while Tanks E and F had an infinite fuel supply. A pump turned red (i.e., stopped pumping)



(a) Tracking (b) System Monitoring

(c) Resource Management (d) Communications

Fig. 2: NASA MATB-II Tasks: (a) Tracking, (b) System Monitoring, (c) Resource Management, and (d) Communications.

when it failed. This composite task also incorporates a *visual inspection* task.

The communications composite task (Figure 2d) required listening to air-traffic control requests for radio changes. The communication request was similar to: "NASA 504, please change your COM 1 radio to frequency 127.550." Participants were to change the specified radio to the specified frequency by selecting the desired radio and using mouse clicks to change the frequency. Communications not directed to the participants' aircraft (i.e., NASA 504) were to be ignored. The communication composite task entails a *visual locate* task for locating the radio channels.

The visual tasks had varying durations: the tracking task is the longest, typically ranging from 20 to 60 seconds, followed by the locate task ranging from 10 to 15 seconds. The inspection task is the shortest, ranging from 4 to 10 seconds. Additionally, a *Null* visual task is added to the list to indicate the absence of the other visual tasks. The *Null* task accounts for the transitory interval during which no visual task is performed (e.g., walking or transition between visual tasks).

A tutorial video described the NASA MATB-II tasks and how to accomplish the tasks. The tutorial video was followed by a 10-minute training session during which participants gained familiarity with the task environment. The training session cycled through the composite tasks, with each task occurring continuously for a 1-minute period that was repeated one additional time. The 52.5-minute trial switched the composite tasks rapidly, and sometimes overlapped tasks in order to emulate real-world scenarios. Forty-five participants (24 male, 20 female and 1 non-binary) completed the experiment. The mean age and standard deviation (std. dev.) were 30.10

and 9.91, respectively, with a range from 18 to 60.

B. Visual Task Recognition Algorithm

The visual task recognition algorithm employs a multimodal approach, incorporating features extracted from the *Pupil Core* eye tracker's fixations, saccades, and the *Xsens*' forehead inertial metrics. The fixation and saccade gaze features capture the eye movements' spatio-temporal characteristics [2], while the inertial features provide additional context associated with the head movements [11], [12].

The inertial metrics, sampled at 40 Hz, are smoothed using a moving average filter to reduce unwanted signal artifacts. The eye tracker implements an internal dispersion-based fixation detector [13] that converts the noisy raw eye gaze data (sampled at 120 Hz) into a series of fixations and saccades. A t_w-second sliding window (with a 50% overlap) is applied to the sensor stream for each metric.

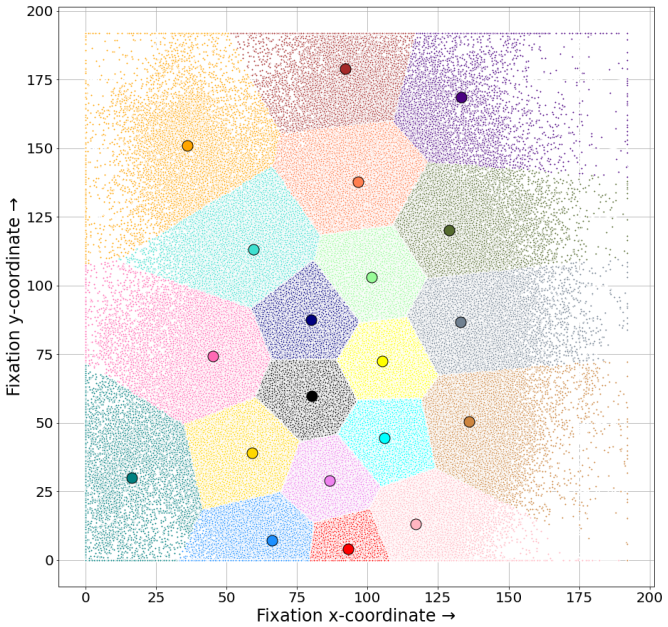


Fig. 3: Fixation Cluster

Initially, the participants' eye movements are analyzed by clustering the fixations and saccades separately, using *K*-means clustering, an N = 20 clusters resulted in the best classifier performance. The fixation x, y coordinates gathered across all participants are grouped into 20 clusters (Figure 3), as are the saccades by grouping the saccadic distances (δ_x, δ_y) in the $\vec{x}$ and $\vec{y}$ axes (Figure 4). A total of 2,780 fixations and saccades were used for clustering. The fixation and saccades' range is given by the 192 x 192 resolution pupil image, captured by the eye tracker. Both clusters are used for constructing the fixation and saccade histograms during feature extraction.

Three different types of feature sets are extracted per sliding window: *fixation, saccadic and inertial*. The fixation features are the fixation rate, fixation histogram, as well as the mean, std. dev. and slope of the fixation duration and dispersion [1], [2]. The fixation dispersion is the angle (degrees) measured

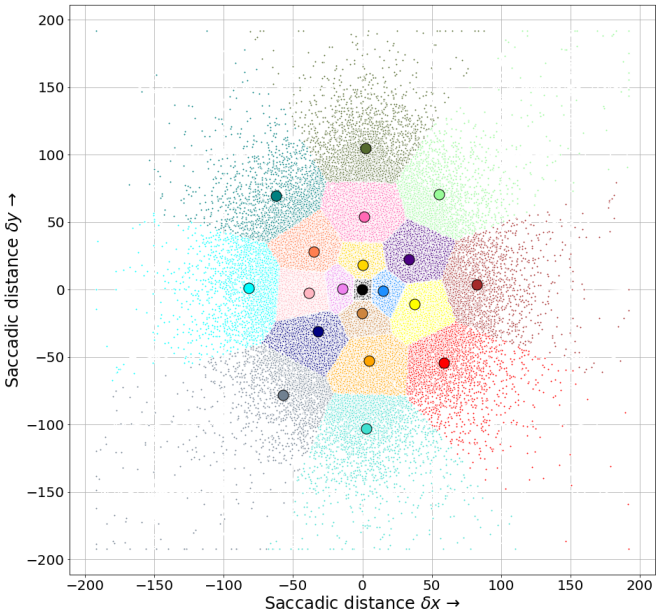


Fig. 4: Saccade Cluster

between a fixation's centroid and the two farthest points dispersed away from the centroid, while the fixation histogram is given by the frequency of the 20 fixation clusters. The saccadic features are the saccade length's mean, std. dev. and slope, as well as the saccadic histogram, which is given by the frequency of the 20 saccadic clusters. Finally, the inertial features consists of the accelerations' and angular velocities' mean, std. dev., and slope.

The extracted features are fed into a Random Forest (RF) classifier with 100 decision trees, and a max depth of 500, where the parameters are chosen based on classifier performance. The RF classifier is an ensemble algorithm that aggregates the decisions of multiple unique individual decision trees, which reduces the overall variance and results in good generalization [14]. The RF classifier tends to outperform most other classification methods, without overfitting [14].

The window size has a significant effect on the number of fixations and saccades available for feature extraction [2]. Smaller time windows allow for near real-time detection, but have poor accuracy, while longer windows have access to more information, resulting in better accuracy [1]–[3], [7], [8], [15]; thus, various window sizes $t_w = \{5s, 10s, 15s, 30s, 60s\}$ with a 50% overlap are investigated.

Multiple tasks can occur within a given window, especially with larger window sizes, but the task that occurs the most frequently is considered to be the given window's desired task. Therefore, each window is labeled by the task that occurred with highest frequency within the window's duration.

C. Validation and Hypotheses

Two different datasets were used to train the algorithm separately in order to analyze generalizability, resulting in two trained algorithm variants: the ten minute training session's

TABLE I: Visual task recognition algorithms' accuracy % [mean (std. dev)] aggregated across participants by dataset and window size. NOTE: The highest accuracies across the algorithms for each dataset and window size are presented in **Bold**, while the overall highest accuracy for each dataset is highlighted in ***Italics***.

Dataset	Algorithm	Window size				
		5	10	15	30	60
TRN	RF	**66.98 (9.14)**	**68.88 (10.88)**	**68.11 (14.0)**	***71.89 (15.87)***	**70.71 (23.01)**
	ANN	45.54 (10.90)	47.28 (8.12)	45.28 (8.68)	48.32 (8.71)	38.24 (20.42)
	SVM	51.20 (9.99)	54.56 (10.67)	53.71 (13.69)	54.19 (16.42)	47.30 (19.44)
TRL	RF	**54.64 (9.11)**	**56.11 (9.08)**	**57.16 (9.40)**	**57.63 (9.78)**	***61.62 (10.91)***
	ANN	30.86 (5.81)	34.29 (8.19)	32.96 (4.24)	33.87 (5.50)	38.15 (7.19)
	SVM	40.86 (6.54)	43.08 (7.66)	43.31 (8.10)	45.50 (8.55)	49.38 (12.36)

data (TRN), and the 52.5-minute trial session's data (TRL). The algorithm variants were validated using the *leave-one-subject-out* cross-validation scheme, where the average accuracy is reported by training the algorithm repeatedly on all, but one participant's data and validating using the left-out participant's data [16]. Both datasets were balanced by randomly downsampling overrepresented visual task instances in order to ensure that the algorithm's accuracy is not artificially inflated.

The RF algorithm was trained to predict one of the four visual tasks: i) Tracking, ii) Inspect, iii) Locate and iv) Null for each window. The evaluated window sizes inform the impact of the window size on the algorithm's performance. The algorithm's performance was validated against two other classifiers: i) Artificial neural network (ANN) with two rectified linear activated hidden layers, with 16 neurons and a 50% dropout at each layer, and ii) a SVM with a radial basis function kernel. Both ANN and SVM classifiers consumed the same features. SVM is chosen for it popularity among visual task recognition algorithms [1], [2], [7], while ANN is chosen for its ability to better learn the non-linearities in the data [17].

Hypothesis $\mathbf{H_1}$ predicted that the TRN variant's accuracy will be significantly higher than the TRL variant's. Hypothesis $\mathbf{H_2}$ predicted that the algorithms' accuracy will increase, as the window size increases, before reaching a point of diminishing returns. Hypothesis $\mathbf{H_3}$ predicted that the algorithm will detect tasks with $\geq 80\%$ classification accuracy for at least one of the analyzed window sizes.

IV. Results

The Friedman analysis of variance by ranks test is used to determine statistical significance in accuracies between results. If significant differences exist, the Wilcoxon signed-rank test was used to identify the specific significant differences. The non-parametric statistical tests ensured that the outcomes were unaffected by the accuracy distribution across participants.

The RF algorithm had the best performance across all window sizes for both data sets, as indicated in Table I, while the ANN had the worst performance across all window sizes for both the datasets. The Wilcoxon signed-rank test between the algorithms' accuracies across window sizes indicated that the RF's accuracy was significantly higher ($p < 0.01$) than both the ANN's and SVM's accuracies across all window sizes and both datasets.

The RF algorithm's accuracy on the TRN dataset increased gradually with window size, peaking at the 30s window size (71.89%). The Friedman's test revealed that the accuracies were significantly different between window sizes within the RF ($\chi^2(5, 44) = 46.89, p < 0.01$). The Wilcoxon signed-rank test indicated that the RF algorithm's 30s window size's accuracy was significantly higher than the 5s ($p < 0.01$), 10s ($p < 0.05$) and 15s ($p < 0.05$) windows, while the 5s window size's accuracy was significantly lower than the 10s ($p < 0.05$). Other accuracy differences were not significant.

The RF algorithm's accuracy on the TRL dataset increased gradually with window size, achieving the highest accuracy at the 60s window size (61.62%). The Friedman's test revealed that the accuracies were significantly different between window sizes within the RF ($\chi^2(5, 44) = 29.0, p < 0.01$). The Wilcoxon's test indicated that the RF algorithm's 5s window size's accuracy was significantly lower ($p < 0.01$) than all other window sizes, while the algorithm's 10s window size's accuracy was significantly lower ($p < 0.01$) than the 15s, 30s and 60s window sizes. The test also revealed that the 60s window size's accuracy was significantly higher ($p < 0.01$) than all other window sizes, while accuracy differences between the other window sizes were not significant.

TABLE II: Visual task recognition accuracy [mean % (std. dev.)] by the incorporated metrics for the RF algorithm with the 60s window for the TRL dataset aggregated across participants. The highest accuracy is highlighted in **Bold**.

Metrics	Accuracy
Fixation	39.82 (12.1)
Saccades	48.49 (14.14)
Inertial	**53.80 (10.35)**
Fixation + Saccades	48.49 (13.51)
Fixation + Inertial	56.14 (10.33)
Saccades + Inertial	**60.18 (11.38)**
Fixation + Saccades + Inertial	**61.62 (10.91)**

The incorporated metrics can impact the RF algorithm's performance. Given that the RF algorithm outperformed the SVM and ANN options, the analysis by incorporated metric was performed only for the RF algorithm. Using the TRL dataset with the 60s window size, the RF algorithm was trained by combining the metrics in several combinations. A total of seven combinations were evaluated by incorporating the metrics individually, and by two and three metrics simultaneously (as shown in Table II).

The analysis by individual metric found the highest accuracy

(53.80%) was attained by the head inertial metrics, while the fixation metrics had the lowest accuracy (39.82%). The Wilcoxon signed-rank test revealed that the head inertial metrics' accuracy was significantly higher ($p < 0.01$) than the other two metrics, and the saccade metrics' accuracy was significantly higher ($p < 0.01$) than the fixation metrics.

The highest accuracy (60.18%) when incorporating two metrics simultaneously was achieved by combining the saccades and head inertial metrics, while the lowest accuracy (48.49%) was recorded when the fixation and saccades were combined. The Wilcoxon signed-rank test revealed that the saccades and head inertial combination's accuracy was significantly higher ($p < 0.01$), than the remaining two combinations. The test also revealed that the saccade and head inertial combination's accuracy and the accuracy of all three metrics combined did not differ significantly. A similar trend was observed for the metric combinations across the window sizes (i.e., 5s, 10s, 15s, 30s) for the TRL variant.

V. Discussion

Hypothesis $\mathbf{H_1}$ predicted that the RF algorithm's TRN variant's accuracy will be significantly higher than the TRL variant, which was fully supported across all window sizes. The algorithm's accuracy on the TRN dataset was higher (up to 14%) than the TRL datset, even though the TRL dataset was significantly larger. This difference can be attributed to each training session's tasks occurring over a prolonged time period, allowing the features to be representative of the visual characteristics of each individual task. The multi-tasking nature of the trial session does not result in such task isolation, which reduces the task accuracy.

Hypothesis $\mathbf{H_2}$ predicted that the RF algorithm's accuracy will increase, as the window size increases, before reaching a point of diminishing returns. The hypothesis was fully supported, as the TRN variant's accuracy increased until 30s window size before decreasing at 60s, while the TRL's accuracy continued to increase until the 60s window size. The RF algorithm with the 60s window size had the best overall performance for the TRL dataset; thus, if a single window size variant is to be selected, it is the recommended algorithm and window size using the current metrics.

Hypothesis $\mathbf{H_3}$ predicted that the RF's algorithm will detect tasks with $\geq 80\%$ classification accuracy for at least one of the window sizes. This hypothesis was not supported, as the RF algorithm's maximum accuracy was only $\sim 60\%$ for the TRL dataset, regardless of the window size. The algorithm's poor performance can be attributed to two factors. The participants' eye and head movement patterns may not have been distinct enough between tasks in the TRL dataset, indicating that the multi-tasking nature may have had an negative impact on detection accuracy. Additionally, labeling the visual tasks is non-trivial and highly uncertain, as it is difficult to determine when exactly the participant's visual processes began prior to task execution. This labeling uncertainty may have also exacerbated the poor performance, particularly the TRL dataset.

It is important to determine the incorporated metrics' ability to detect the tasks reliably. The selected metrics were inadequate to capture the participants' visual behavior in a multi-tasking environment. The per metric analysis indicated that Xsens' head motion inertial data was the most useful, followed by the saccade and fixation metrics. The visual task detection analysis determined that the incorporated metrics are less responsive to reliably detect tasks in a dynamic, multi-tasking environment (i.e., switching tasks frequently). Additional metrics (e.g., microsaccades and scanpath) extracted from the eye tracker can be incorporated and may improve the algorithm's performance.

Identifying the appropriate window size for each algorithm informs how the metrics must be segmented, such that the features extracted are representative of the tasks being detected. The analysis indicated that the incorporated metrics generally require larger window sizes ($> 30s$) to detect visual tasks more accurately. The TRN session's tasks were labeled reliably, as the participants performed the tasks roughly for a minute before task switching. Given this labeling certainty, the algorithm achieved its highest accuracy for the 30s window size, suggesting that thirty seconds of data is required to assimilate the context needed to detect the tasks reliably. The TRL session's data is highly uncertain due to the rapid task switching and accompanied labeling difficulty; therefore, it is harder for the algorithm to assimilate the required context at lower window sizes. The 60s window size is recommended, because it is large enough to provide the algorithm with the required context (i.e., equivalent to that of a thirty-second less uncertain data), amidst the uncertainty. This larger window size is believed to enable the algorithm to learn a better relationship between the tasks and metrics, given the uncertainty. Additional analysis with t_w ranging between 30s and 60s is required to determine the peak performing window size for the TRN and TRL datasets.

Visual tasks will have different durations. A short task (e.g., inspection) may require a smaller window, so that the task is not overshadowed (e.g., confused) by all the unrelated data; therefore, it may be necessary for the task recognition algorithm to use an adaptive sliding window method [18], [19]. An adaptive sliding window will permit for expanding and contracting the window size, based on the task, may lead to more accurate detection.

VI. Conclusion

A robot's ability to detect tasks that involve non-physical movements is crucial for HRTs in which humans perform wide range of tasks. The primary contribution is the developed visual task recognition algorithm that incorporated metrics acquired from wearable sensors to detect visual tasks performed by HRTs in a non-sedentary environment. Combining the spatio-temporal eye gaze metrics with the head inertial metrics improved the algorithm's performance by providing additional context. While the developed visual task recognition algorithm did not meet the expected standard ($< 80\%$ accuracy), it is a viable candidate for incorporation into an HRT system

that entails multiple activity components. Future work will improve the algorithm's performance by incorporating additional metrics, such as microsaccades and scanpath. *Multiple instance learning* algorithms will be investigated to address the labeling ambiguity and gain accuracy improvements. Additionally, adaptive sliding window methods to detect visual tasks of varying lengths will be investigated. The task detection algorithm will also be evaluated in an uncertain, dynamic peer-based task environment in order to assess its viability across human-robot teaming domains.

VII. Acknowledgement

The graduate students have been supported by an Office of Naval Research award N00014-21-1-2190. The contents are those of the authors and do not represent the official views of, nor an endorsement, by the Office of Naval Research.

References

[1] A. Bulling, J. A. Ward, H. Gellersen, and G. Tröster, "Eye movement analysis for activity recognition using electrooculography," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 741–753, 2010.

[2] N. Srivastava, J. Newn, and E. Velloso, "Combining low and mid-level gaze features for desktop activity recognition," *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 4, pp. 1–27, 2018.

[3] F. Martinez, E. Pissaloux, and A. Carbone, "Towards activity recognition from eye-movements using contextual temporal learning," *Integrated Computer-Aided Engineering*, vol. 24, no. 1, pp. 1–16, 2017.

[4] S. Ishimaru, K. Kunze, Y. Uema, K. Kise, M. Inami, and K. Tanaka, "Smarter eyewear: Using commercial EOG glasses for activity recognition," in *ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 2014, pp. 239–242.

[5] Y. Lu, C. Zhang, B.-Y. Zhou, X.-P. Gao, and Z. Lv, "A dual model approach to EOG-based human activity recognition," *Biomedical Signal Processing and Control*, vol. 45, pp. 50–57, 2018.

[6] S. Ishimaru, K. Hoshika, K. Kunze, K. Kise, and A. Dengel, "Towards reading trackers in the wild: Detecting reading activities by EOG glasses and deep neural networks," in *ACM International Joint Conference on Pervasive and Ubiquitous Computing and ACM International Symposium on Wearable Computers*, 2017, pp. 704–711.

[7] C. Kelton, Z. Wei, S. Ahn, A. Balasubramanian, S. R. Das, D. Samaras, and G. Zelinsky, "Reading detection in real-time," in *ACM Symposium on Eye Tracking Research and Applications*, 2019, pp. 1–5.

[8] M. Landsmann, O. Augereau, and K. Kise, "Classification of reading and not reading behavior based on eye movement analysis," in *ACM International Joint Conference on Pervasive and Ubiquitous Computing and ACM International Symposium on Wearable Computers*, 2019, pp. 109–112.

[9] M. R. Islam, S. Sakamoto, Y. Yamada, A. W. Vargo, M. Iwata, M. Iwamura, and K. Kise, "Self-supervised learning for reading activity classification," *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 3, pp. 1–22, 2021.

[10] J. R. Comstock and R. J. Arnegard, "The multi-attribute task battery for human operator workload and strategic behavior research," *Technical Report NASA Tech. Memorandum 104174*, 1992.

[11] S. Ishimaru, K. Kunze, K. Kise, J. Weppner, A. Dengel, P. Lukowicz, and A. Bulling, "In the blink of an eye: Combining head motion and eye blink frequency for activity recognition with google glass," in *ACM Augmented Human International Conference*, ACM, 2014, pp. 1–4.

[12] P. Lagodzinski, K. Shirahama, and M. Grzegorzek, "Codebook-based electrooculography data analysis towards cognitive activity recognition," *Computers in Biology and Medicine*, vol. 95, pp. 277–287, 2018.

[13] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *The Symposium on Eye Tracking Research and Applications*, ACM, 2000, pp. 71–78.

[14] S. Misra and H. Li, "Chapter 9 - noninvasive fracture characterization based on the classification of sonic wave travel times," in *Machine Learning for Subsurface Characterization*, S. Misra, H. Li, and J. He, Eds., Gulf Professional Publishing, 2020, pp. 243–287.

[15] P. Kiefer, I. Giannopoulos, and M. Raubal, "Using eye movements to recognize activities on cartographic maps," in *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2013, pp. 488–491.

[16] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data mining, Inference, and Prediction*. Springer New York, NY, 2009, vol. 2.

[17] Y. Goldberg, "Neural network methods for natural language processing," *Synthesis lectures on human language technologies*, vol. 10, no. 1, pp. 1–309, 2017.

[18] C. Ma, W. Li, J. Cao, J. Du, Q. Li, and R. Gravina, "Adaptive sliding window based activity recognition for assisted livings," *Information Fusion*, vol. 53, pp. 55–65, 2020.

[19] M. H. M. Noor, Z. Salcic, I. Kevin, and K. Wang, "Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer," *Pervasive and Mobile Computing*, vol. 38, pp. 41–59, 2017.