

Reading Detection in Real-time

Conor Kelton
Stony Brook University
Stony Brook, New York

Zijun Wei
Stony Brook University
Stony Brook, New York

Seoyoung Ahn
Stony Brook University
Stony Brook, New York

Aruna
Balasubramanian
Stony Brook University
Stony Brook, New York

Samir R. Das
Stony Brook University
Stony Brook, New York

Dimitris Samaras
Stony Brook University
Stony Brook, New York

Gregory Zelinsky
Stony Brook University
Stony Brook, New York

Abstract

Observable reading behavior, the act of moving the eyes over lines of text, is highly stereotyped among the users of a language, and this has led to the development of reading detectors—methods that input windows of sequential fixations and output predictions of the fixation behavior during those windows being reading or skimming. The present study introduces a new method for reading detection using Region Ranking SVM (RRSVM). An SVM-based classifier learns the local oculomotor features that are important for real-time reading detection while it is optimizing for the global reading/skimming classification, making it unnecessary to hand-label local fixation windows for model training. This RRSVM reading detector was trained and evaluated using eye movement data collected in a laboratory context, where participants viewed modified web news articles and had to either read them carefully for comprehension or skim them quickly for the selection of keywords (separate groups). Ground truth labels were known at the global level (the instructed reading or skimming task), and obtained at the local level in a separate rating task. The RRSVM reading detector accurately predicted 82.5% of the global (article-level) reading/skimming behavior, with accuracy in predicting local window labels ranging from 72-95%, depending on how tuned the RRSVM was for local and global weights. With this RRSVM reading detector, a method now exists for near real-time reading detection without the need for hand-labeling of local fixation windows. With real-time reading detection capability comes the potential for applications ranging from education and training to intelligent interfaces that learn what a user is likely to know based on previous detection of their reading behavior.

CCS Concepts

• Human-centered computing → User models;

Keywords

Reading Detection, Real-time Applications

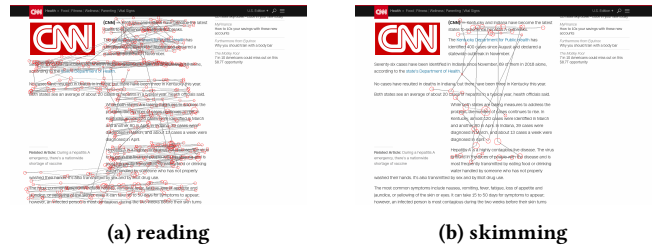


Figure 1: Scanpaths from two subjects tasked with a) reading and b) skimming an article from cnn.com/health. Reading behavior is characterized by horizontal saccades with a small vertical return movement. Skimming is characterized by less organized rightward saccades and occasional large vertical leaps throughout the document.

ACM Reference Format:

Conor Kelton, Zijun Wei, Seoyoung Ahn, Aruna Balasubramanian, Samir R. Das, Dimitris Samaras, and Gregory Zelinsky. 2019. Reading Detection in Real-time. In *2019 Symposium on Eye Tracking Research and Applications (ETRA '19)*, June 25–28, 2019, Denver, CO, USA. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3314111.3319916>

1 Introduction

Despite the importance of reading behavior in our everyday lives, and the many applications that might benefit from accurate and robust reading detection, existing solutions are limited to only two types of methods. One approach is to build a reading detector that globally mines information from entire eye movement scanpaths for the purpose of classifying an extended period of behavior as reading. More specifically, these global methods work by labeling all the eye movement reading behavior occurring over an approximately one-minute period, then use these labeled data, along with several oculomotor features, to train a classifier to predict reading versus non-reading behavior [Ishimaru et al. 2017].

The second approach is more local in space and time and attempts to classify reading behavior using information from only small temporal windows of sequential fixations. Local methods work by having reading “experts” label the fixations occurring within a narrow window of time as indicating either reading or skimming behavior, then using these labels and either saccade length features [Kollmorgen and Holmqvist 2007] or features capturing the forward progress and vertical angle of the eye [Biedert et al. 2012] to train a reading classifier. These methods are local in that they make predictions based on information from fixations

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ETRA '19, June 25–28, 2019, Denver, CO, USA

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-6709-7/19/06...\$15.00

<https://doi.org/10.1145/3314111.3319916>

within a narrow temporal window rather than from an extended scanpath— a prerequisite to real-time reading detection.

Both global and local methods have drawbacks. Global methods derive their predictive power from access to a longer reading context, but this coarser timescale means that these methods are poorly suited to applications requiring real-time and spatially-constrained reading detection. Relatedly, because normal reading behavior may involve periods of skimming, and vice versa, global methods would fail to capture this finer-grained structure. Local methods allow for (near) real-time reading detection at the fixation level, but are likewise limited. These methods require the assignment of ground truth labels to local segments of reading behavior for model training, but the collection of these labels is tedious, noisy, and impractical.

In the present study, we introduce a new method for building a reading detector, which we also evaluate in the context of reading versus skimming classification. This method uses Region Ranking SVM (RRSVM) [Wei et al. 2016; Wei and Hoai 2016], which enables the incorporation of both global and local sources of information to perform more accurate reading detection. Details about this method are provided in Sec. 3.3. Briefly, the method uses an SVM classifier to learn a weighting of oculomotor features over narrow fixation windows to predict local reading detection. But critically, as the classifier is optimizing the global reading/skimming classification it is simultaneously learning which of the local windows are most important in the global detection. The RRSVM method, therefore, integrates global and local methods in a single reading detector. Moreover, because the local fixation windows that are important for classifying reading behavior are learned, it also eliminates the need for humans to explicitly label each local fixation window as ground truth for training— the human rater is taken out of the loop. This RRSVM reading detector was trained and evaluated using eye movement data collected from participants reading or skimming modified web news articles. Details regarding this user evaluation study are described next.

2 Reading-Skimming User Study

Thirty-two undergraduate student participants, all native English speakers, were randomly assigned to either a skimming ($n = 16$) or reading for comprehension ($n = 16$) task. No users were removed as outliers for our analyses. Reading and skimming are distinct behaviors [Biedert et al. 2012; Duggan and Payne 2011; Strukelj and Niehorster 2018]. The aim of reading is comprehension and the aim of skimming is to get a general idea of a text’s content [Rayner et al. 2016]. Thus, for the skimming task, participants were given 10 seconds to quickly skim through each news article under instruction to select 3 keywords best summarizing the content of the article. For the reading task, participants were given an unlimited amount of time to read carefully through the same news articles, with each reading measure followed by a single question aimed at evaluating comprehension. These general procedures were iterated over 10 trials, each using a different news article. Training and test sets were constructed by randomly selecting 80% of the articles for training, and the remaining 20% for testing.

2.1 Stimuli, Apparatus, and Procedure

Stimuli were 10 articles selected and modified from cnn.com/health. Each article was presented on a single page Story highlights were

Table 1: Means and standard deviations for measures in the reading and skimming tasks.

Measures	Reading ($n=16$)		Skimming ($n=16$)	
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>
Total Reading Time (sec)	77.94	20.38	9.96	0.28
Fixation Count	288.65	71.56	39.01	5.46
Average Fixation Duration (ms)	226.68	21.49	225.39	37.23
Average Saccade Amplitude (°)	3.05	0.36	3.19	0.45
Leftward Saccades (%)	32.06	6.16	32.92	4.22
Accuracy (%)	88.7	11.47	N/A	N/A

replaced with CNN/Health logos, and pictures in the article were removed to avoid distraction. Figure 1 shows a typical stimulus, with superimposed scanpaths from users tasked with reading or skimming. News article images subtended $31^\circ \times 24^\circ$, with this viewing angle enforced by chin and forehead rest. Eye position was measured using an EyeLink 1000 (SR Research) eye-tracker sampling at 1000 Hz, and was parsed into fixations using EyeLink On-Line Parser with default settings. The time taken to complete the experiment was 5 minutes for skimming and 20 minutes for reading, including 9-point calibration, and the order of presentation of the news articles was randomized. There was no significant effect of the trial order in any of the eye-movement measures reported in Table 1 (all $ps > .05$)

2.2 Descriptive Results

The accuracy rates for reading were higher than 70% across all participants, with mean accuracy rates of 88.7% ($SD = 11.47$). Given this high accuracy, we were convinced that users were indeed reading for comprehension and therefore did not exclude articles where the comprehension question was answered incorrectly. Three keywords generated during skimming task was recorded, but not analyzed. Descriptive statistics for some conventional reading/skimming measures are provided in Table 1. Although, as expected, the total reading time and the number of fixations made during reading were both greater than the corresponding skimming measures ($t(30) = 13.34$, $p < .05$; $t(30) = 13.91$, $p < .05$, respectively), the eye-movement measures did not significantly differ across tasks ($p > .05$). This suggests that there is much skimming behavior in normal reading, and vice versa, highlighting the difficulty of the detection task at hand.

3 Reading-Skimming Classification

We design a reading-skimming classifier that learns gaze patterns when reading versus skimming. More relevant to our study is how similar gaze movements are when reading versus skimming. To answer this question we need a reading-skimming classifier. We design a Region Ranking SVM (RRSVM) classifier [Wei and Hoai 2016] that exchanges information between global and local labels to build an accurate reading detector.

3.1 Features

Our data consists of scanpaths from user study participants. The task given to the participants are to either read or skim a news article (see Sec. 2).

For the features of each window in our dataset, we utilized the intuitive features for reading detection from scanpaths highlighted by

past ETRA work on reading detectors, i.e. from Biedert et al. [Biedert et al. 2012]. Each window is described by its average forward (left to right) progress, and overall saccade angle. Reading behavior should be shorter in forward progress and lesser in angle, and vice versa. We also used additional features including average fixation duration, document area covered, and average backward progress of the windows. However we saw no quantitative change to our results using these additional features. We thus leave the continued construction of descriptive feature sets to future work.

3.2 Problem Formulation

We now discuss the formulation of our RRSVM detector. Our task here is to have a classifier $\mathcal{F}_\theta$ with learning parameters θ that

- (1) Accurately predicts global (document) level reading or skimming
- (2) Determines the most likely windows that suggest reading or skimming behavior

More specifically, the RRSVM sees the reading detection task as a multiple instance learning problem where the windows in a document are considered jointly to decide if the document is being read or skimmed.

One way to approach the problem of global and local detection would be to simply take a classifier that works reasonably well on a local level, and have the local classes vote on a global class. However past works who predict global knowledge using local information have shown this to be inadequate [Wei et al. 2016; Wei and Hoai 2016]. RRSVM instead learns a distribution for the weighted importance of windows, which we will show also outperforms the naive method for our task.

3.3 Region Ranking SVM Optimization

We now formally define the objective function that RRSVM optimizes to perform our reading detection task. Our RRSVM assumes documents contain windows $\{\mathbf{X}_i\}_{i=1}^n$ and associated binary labels $\{y_i\}_{i=1}^n$ indicating whether this document is representative of reading behavior or not. To account for the uncertainty of individual local windows, RRSVM considers multiple of them, and without loss of generality, and for brevity, we assume each page has N windows. Let m be the number of windows for each page, and d the dimension of the window features. RRSVM represents each page as a matrix $\mathbf{X}_i \in \mathbb{R}^{d \times m}$, but the order of the columns can be arbitrary. RRSVM jointly learns a window evaluation function $\mathbf{w}$ and a window selection function $\mathbf{s}$ for the following:

$$\underset{\mathbf{w}, \mathbf{s}, b}{\text{minimize}} \quad \lambda \|\mathbf{w}\|^2 + \sum_{i=1}^n (\mathbf{w}^T \Gamma(\mathbf{B}_i; \mathbf{w}) \mathbf{s} + b - y_i)^2 \quad (1)$$

$$\text{s.t. } s_1 \geq s_2 \geq \dots \geq s_m \geq 0, \quad (2)$$

$$h(\Gamma(\mathbf{B}_i; \mathbf{w}) \mathbf{s}) \leq 1. \quad (3)$$

Here $h(\cdot)$ is the function that measures the spread of the column vectors of a matrix: $h([\mathbf{x}_1, \dots, \mathbf{x}_n]) = \sum_{i=1}^n \|\mathbf{x}_i - \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i\|^2$. $\mathbf{w}$ and b are the weight vector and the bias term of an SVM classifier, which are the parameters of the window evaluation function. $\Gamma(\mathbf{B}; \mathbf{w})$ denotes a matrix that can be obtained by rearranging the columns of the matrix $\mathbf{B}$ so that $\mathbf{w}^T \Gamma(\mathbf{B}; \mathbf{w})$ is a sequence of non-increasing values. The vector $\mathbf{s}$ is the weight vector for combining the SVM window scores for each document; this vector is common

to all documents of a class. In this case, there will be two sets of $\mathbf{s}$ vectors, one for reading, one for skimming.

The above objective function consists of the regularization term $\lambda \|\mathbf{w}\|^2$ and the sum of squared losses. Note that the classification decision is based on both the window evaluation function (i.e., $\mathbf{w}, b$) and the window selection function (i.e., $\mathbf{s}$), which are simultaneously learned using the above formulation. The learning formulation can be iteratively optimized with block coordinate descent, alternating between the following two procedures:

- (1) Fix $\mathbf{w}$, optimize Eq. (1) w.r.t. $\mathbf{s}$ and b ,
- (2) Fix $\mathbf{s}$, optimize Eq. (1) w.r.t. $\mathbf{w}$ and b .

More details about the optimization process can be found in prior work [Wei and Hoai 2016], that uses a similar idea in the context of image prediction.

3.4 Benefits and Extensions

In summary, the main benefits of using RRSVM are the following:

- RRSVM has been shown to outperform related methods, which use alternatives such as average pooling, for classification tasks that feature global and local labels [Sermanet et al. 2013; Szegedy et al. 2014].
- It works without labeled window data as it frames the problem into a weakly supervised learning framework. As we will show, RRSVM helps localize which windows contribute to the final decision.

Further, we briefly discuss how our RRSVM can extend to common scenarios, such as incremental learning, or if some trusted local labels are already available. To operate incrementally, RRSVM can use a classifier trained on partial data as a good initialization for a classifier of the complete data, thus reducing the number of iterations to converge to optimal values of $\mathbf{w}$ and $\mathbf{s}$. Similarly, RRSVM can use available local labels to learn a better $\mathbf{w}$ for initialization, which would presumably converge more quickly on $\mathbf{s}$. This would save the need to iteratively learn both $\mathbf{w}$ and $\mathbf{s}$.

4 Evaluation

In this section we formally evaluate the performance of our RRSVM reading detector, which is able to distinguish both global and local signals of reading and skimming detection using only global labels. In our analysis, we compare the performance and describe the trade-offs between tuning our method for the global vs. local task. We also provide some intuition as to why the problem of reading detection is difficult to scale with only global information.

4.1 Global Evaluation

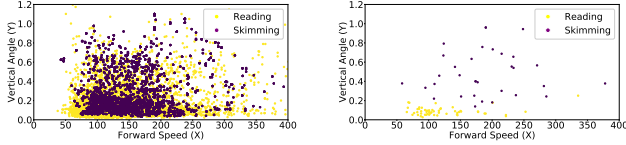
We first evaluate how our model performs on our global article labels. For the labels of each document, we simply used the task given to the user, as discussed in Sec. 2. Our methodology uses a "one against all" approach to document classification, and we thus have a separate set of weights and predictions for each class. We thus consolidate into one final label, y_i ,

$$y_i = \operatorname{argmax}\{\mathbf{w}_s \mathbf{X}_s \mathbf{s}_s + b_s + \delta, \mathbf{w}_r \mathbf{X}_s \mathbf{s}_r + b_r\} \quad (4)$$

where the subscripts r and s denote the parameters learned for reading and skimming respectively. We tune the parameter δ , for the optimal value of our evaluation metric, the balanced classification

Table 2: Balanced Classification Rate (BCR) for different variants of our RRSVM reading detector across global and local tasks.

Method	Performance (BCR)	
	Global	Local
RRSVM Default (BCR)	.751	.951
RRSVM Converged (BCR)	.825	.720

**(a) Globally Annotated Window Features (b) Locally Annotated Window Features****Figure 2: Scatter plots of (a) globally annotated windows from our test data and (b) locally annotated windows where raters agreed. We can observe a difference in separability when using global vs. local**

rate, *BCR*, (i.e. the average between the model’s sensitivity and specificity), over our training set.

Table 2 shows the performance of our method on our global labels. We compare two variants of our RRSVM detector. The first, labeled default, is simply the RRSVM trained for a single iteration to optimize $\mathbf{w}$ with $\mathbf{s}$ initialized to the default $1/n$ where n is the number of windows in the document. The second is our RRSVM after full optimization of $\mathbf{w}$ and $\mathbf{s}$ until convergence. A fully convergent RRSVM is able to capture the reading detection with greatest accuracy at 82%. A fixation window size of 13 was used for computing all features for the documents.

In this regard, we also measured the reading detection task across window sizes. Here, there is a trade-off between capturing more information with larger windows to improve accuracy vs the increased time/spacial lag for real-time prediction. We noticed that both methods decreased in accuracy by about 3% when using smaller windows, confirming this trade off.

4.2 Understanding the Problem Difficulty

In order to further understand the jump in task difficulty resultant from performing reading detection with only globally annotated ground truth, we provide visualizations using the intuitive features from Biedert et. al. Figure 2 a) shows the globally labeled reading (yellow) and skimming (purple) windows against their two dimensional features, whereas b) shows the smaller set of locally labeled windows against the same feature types.

We can observe that the separability of the dataset varies drastically when utilizing global ground truth as opposed to local ground truth. As stated in Sec. 2, this is due to the existence of skimming behavior when reading and vice versa. This is why the ranking of the windows by importance by RRSVM helps significantly to pick the appropriate windows for the global classification task.

4.3 Local Verification

We now move on the performance of the second task of our detector, fixation level reading detection. As the task of labeling individual fixations for their class is quite arduous, we perform local labeling and analysis on only a subset of the total windows offered in our dataset. For the local analysis, we asked two participants, affiliated with the department of cognitive science at the authors’ institution, to label windows from our test set, overlaid on top of our stimuli, as reading or skimming. The total number of windows labeled was 120, made up from 2 distinct and non-overlapping windows chosen from the 31 reading and 29 skimming articles in our test set. The labeling was done via a Web interface. For the local ground truth, we only utilized labels for windows in which the raters agreed, which was 75% of all windows labeled.

Similar to the global labels, we can calculate local labels as

$$y'_{ij} = \operatorname{argmax}\{\mathbf{w}_{s_j}\mathbf{X}_j + b_s + \delta', \mathbf{w}_{r_j}\mathbf{X}_j + b_r\} \quad (5)$$

where this y'_{ij} represents the j th local window predictions for the i th document. Here δ' was tuned subjectively based on the model’s perceived performance on local windows in the training set.

The results of the local labels are also shown in Table 2. We can see that in the best case, the RRSVM is able to detect $\approx 95\%$ of the local labels given by the raters. Further, we can see that while the fully converged RRSVM increases the accuracy on the global labels by about 7.5%, there is a noticeable drop in accuracy for the converged model when predicting locally. Thus we can see the trade off for local versus global prediction. As the RRSVM converges, it gives more priority in predicting the global labels. We thus note that the RRSVM detector can be tuned depending on the priority of the task for reading detection, whether it be global or local.

Lastly, we trained exclusively on the local labels to find an upper bound for the local classification performance using an SVM. We find the *BCR*, generated via a 5-fold cross validation, to be 96.2%. This implies that at its best, the RRSVM performs close to the optimal local classification, even just using global labels, which can be obtained at a small fraction of the annotation cost.

5 Conclusion

Real-time reading detection requires the detection of reading behavior in narrow spatio-temporal windows of fixations. Previous methods required for training a human-labeling of these scanpath snip-its as either reading or skimming. Our work is important because it does not require this hand-labeled training data, thereby extending the reach of reading detection to the many applications for which extensive local window labeling would be impractical. Our work is also novel in that it is the first to apply the RRSVM method to the problem of reading detection. RRSVM breaks the dependency on hand-labeled training data by iteratively selecting (learning) local features while optimizing for global classification, a dynamic that was inspired by how attention locally biases features that were selected based on a parallel (global) analysis of visual input. In the current context, the RRSVM-based reading detector learns labels for local windows as part of its optimization for global (article level) reading/skimming classification, making it well suited

to applications requiring real-time reading detection in the absence of human-labeled local windows for training.

6 Discussion and Future Directions

The learned RRSVM parameters w and b require global information to train, during testing for local labeling no global information is required. This involves obtaining the features for a given fixation window, X , and performing the matrix multiplication to obtain the label. The computation delay for matrix multiplication is negligible. In terms of obtaining features, we obtain δ fixations preceding and following i . Given that the average fixation duration in our data was ≈ 225 ms, these past and future δ -fixation windows would each require $\delta * 225$ ms. For $\delta = 6$, we would add 1.35 second delay between when a fixation is made, and when it is labeled.

Recent works have delved into the task of determining cognitive performance in real-time [Katsini et al. 2018; Raptis et al. 2017; Toker et al. 2017]. Our work aims to determine reading/skimming in general *across subjects*; related work focus on predicting individual subjects. Related works also use of conventional classification techniques, while we use a state-of-the-art RRSVM motivated to our reading-skimming task.

As part of future work, we will be to explore the performance of the reading detector in more naturalistic contexts, such as normal Web usage. In everyday practice, reading behavior is likely to be continuously interspersed with periods of skimming, glances to pictures and their captions, video viewing, etc., and the specific composition of these behaviors will certainly depend on the web content and task (e.g., a news article versus a shopping site). With our reading detector we will be able to quantify when reading behavior occurs during the course of normal web usage, thereby better understanding a key component in a users web engagement. Previous studies have shown that a better understanding of web engagement leads to tangible benefits, ranging from targeted performance improvements to better content personalization [Buscher et al. 2009; Eraslan et al. 2016; Kelton et al. 2017].

Acknowledgments

The authors would like to thank our anonymous reviewers whose valuable feedback greatly improved the final version of this paper, and Yupei Chen for her help with our user study design and data collection. This work was supported by the National Science Foundation under grant CNS-1718014.

References

- Ralf Biedert, Jörn Hees, Andreas Dengel, and Georg Buscher. 2012. A robust realtime reading-skimming classifier. In *Proceedings of the Symposium on Eye Tracking Research and Applications*. ACM, 123–130.
- Georg Buscher, Edward Cutrell, and Meredith Ringel Morris. 2009. What Do You See when You're Surfing?: Using Eye Tracking to Predict Salient Regions of Web Pages (*CHI '09*). ACM, 21–30.
- Geoffrey B Duggan and Stephen J Payne. 2011. Skim reading by satisficing: evidence from eye tracking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1141–1150.
- Sukru Eraslan, Yeliz Yesilada, and Simon Harper. 2016. Eye Tracking Scanpath Analysis on Web Pages: How Many Users?. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications (ETRA '16)*. ACM, New York, NY, USA, 103–110.
- Shoya Ishimaru, Kensuke Hoshika, Kai Kunze, Koichi Kise, and Andreas Dengel. 2017. Towards Reading Trackers in the Wild: Detecting Reading Activities by EOG Glasses and Deep Neural Networks. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers (UbiComp '17)*. ACM, New York, NY, USA, 704–711.
- Christina Katsini, Christos Fidas, George E. Raptis, Marios Belk, George Samaras, and Nikolaos Avouris. 2018. Eye Gaze-driven Prediction of Cognitive Differences During Graphical Password Composition. In *23rd International Conference on Intelligent User Interfaces (IUI '18)*. ACM, New York, NY, USA, 147–152. <https://doi.org/10.1145/3172944.3172996>
- Conor Kelton, Jihoon Ryoo, Aruna Balasubramanian, and Samir R. Das. 2017. Improving User Perceived Page Load Times Using Gaze. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*. USENIX Association, Boston, MA, 545–559.
- Sepp Kollmorgen and Kenneth Holmqvist. 2007. Automatically Detecting Reading in Eye Tracking Data. *Lund University Cognitive Studies*.
- George E. Raptis, Christina P. Katsini, Marios Belk, Christos Fidas, George Samaras, and Nikolaos M. Avouris. 2017. Using Eye Gaze Data and Visual Activities to Infer Human Cognitive Styles: Method and Feasibility Studies. In *UMAP*.
- Keith Rayner, Elizabeth R Schotter, Michael EJ Masson, Mary C Potter, and Rebecca Treiman. 2016. So much to read, so little time: How do we read, and can speed reading help? *Psychological Science in the Public Interest* 17, 1 (2016), 4–34.
- Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. 2013. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *CoRR abs/1312.6229* (2013).
- Alexander Struelj and Diederick C Niehorster. 2018. One page of text: Eye movements during regular and thorough reading, skimming, and spell checking. *Journal of Eye Movement Research* 11, 1 (2018).
- Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2014. Going Deeper with Convolutions. *CoRR abs/1409.4842* (2014).
- Dereck Toker, Sébastien Lallé, and Cristina Conati. 2017. Pupillometry and Head Distance to the Screen to Predict Skill Acquisition During Information Visualization Tasks. In *Proceedings of the 22Nd International Conference on Intelligent User Interfaces (IUI '17)*. ACM, New York, NY, USA, 221–231.
- Zijun Wei, Hossein Adeli, Minh Hoai Nguyen, Greg Zelinsky, and Dimitris Samaras. 2016. Learned Region Sparsity and Diversity Also Predicts Visual Attention. In *Advances in Neural Information Processing Systems* 29, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 1894–1902.
- Zijun Wei and Minh Hoai. 2016. Region ranking SVM for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2987–2996.