Modeling Distractibility from Videos

Dominik Graetz[1]

[1] University of Oregon

Author Note

This report was completed as part of the requirements of the Educational Data Science course sequence at the University of Oregon.

Correspondence concerning this article should be addressed to Dominik Graetz. E-mail: dgrtz@uoregon.edu

Abstract

In every moment in time, humans have a choice between following through with a given task (exploitaion) or searching the environment for potentially more rewarding information or tasks (exploration). This paper conceputalizes exploration of economically non-rewarding (but entertaining) stimuli as distraction and aims to predict the time points at which participants are more likely to be distracted. Here, I am presenting results from an experiment in which subjects could work on a simple, rewarding task while distractors in the form of videos are shown. Distractibility from the task caused by videos is measured using eye-tracking on a frame-by-frame basis. Using a dataset with $>$ 80 k observations, I trained lasso regression models, ridge regression models and elasic nets to predict distraction (gazing at videos) from experimental variables alone and from experimental variables and video frame embeddings. Accuracy of these models are relatively high with $>$ 80 %, however, investigation of confusion matrix measures reveals that this level of accuracy is largely due to the relatively lower probability of distraction. These models do not show satisfactory results on key metrics like the true positive rate. However, including video frame embeddings increases the true positive rate significantly. Suggestions for model improvement are discussed.

Link to Github Repo: https://github.com/dgraetz/VideoDistraction

Keywords: Machine Learning, Regularized Regression, Distractibility

Word count: X

Modeling Distractibility from Videos

Research Problem

Research problem (10pts): Describe the task you want to achieve. What is the outcome of interest? What are you trying to predict? Why is it important? What are the potential benefits of having a predictive model for this outcome?

In every moment, humans have the choice between two different actions: *exploitation*, in which they continue engaging in a rewarding task. Alternatively, in the case of *exploration*, humans search the environment for potentially more rewarding, alternative things to do. The factors that determine disengagement from a rewarding task and exploratory behavior are unclear. Previous research suggests that switches from exploitation to exploration can be caused by increasing task uncertainty, elevated error rate in the current task, and lower task reward, or, together, task utility (for overviews, see Aston-Jones and Cohen (2005) and Cohen, McClure, and Yu (2007)). Here, I aim to predict human distractibility from short videos while they are working on a rewarding task. To achieve this, I conducted an experiment in which task utility (i. e., reward), and task utility is systematically manipulated between experimental blocks. Eye-tracking reveals what subjects pay attention to - and this is the outcome I aim to predict from these two experimental variables. Additionally, I obtain frame-by-frame embeddings from ResNet-18 as predictors.

The general approach here is to compare the predictive performance of regularized regression models (i. e., lasso and ridge regression, and elastic net). It is of interest to me to which degree video content determines distractibility - thus, I will examine the degree to which frame-level embeddings improve regularized regression performance by comparing models with and without embeddings.

This research is important because it can help further our understanding about the

situations and the cognitive processes underlying elevated distractibility in populations with ADHD or older adults, given that they often appear more distracted. Moreover, it is crucial for understanding the properties of stimuli that can draw attention away from a target and thus, this research may help improve driver assistance systems, for instance.

Method

Data collection is still ongoing. Currently, data from 16 undergraduate students, collected at the University of Oregon is available. However, for four subjects, eye tracking data is missing - therefore, models were trained with data from twelve subjects.

The experiment was programmed in Matlab 2022a (Inc., 2022) using Psychtoolbox 3 (Brainard & Vision, 1997; Kleiner, Brainard, & Pelli, 2007; Pelli & Vision, 1997). In this experiment, participants perform a series of blocks of trials with a simple computer task. In each trial, an arrow is presented on the screen, pointing into directions of multiples of 90° (up, right, down, left). The arrow can appear in four different colors (red, green, blue, yellow). The subject completes a trial by pressing a key on the numpad dependent on the color (blue - 8, green - 6, red - 2, yellow - 5). Each block is time limited to 60 seconds and participants can earn monetary incentives dependent on accurate performance in a block. Thus, the rational goal would be to "exploit" this task and perform fast and accurately to maximize the reward. However, participants could also "explore" the environment - in 50 % of the trials, videos appeared far or close to the left or right of the arrow. The pool of videos that was generated and used for this experiment was downloaded from the social media website vine.co. and consists of 500 videos with an average duration of 3-6 seconds. For an example, see Figure 1. Using an SR Research Eyetracker, I measured the eye position on the screen at 1000 Hz.

Multiple experimental manipulations were implemented, and I will focus on the relevant factors here. First, I included a reward manipulation - in half of the blocks,

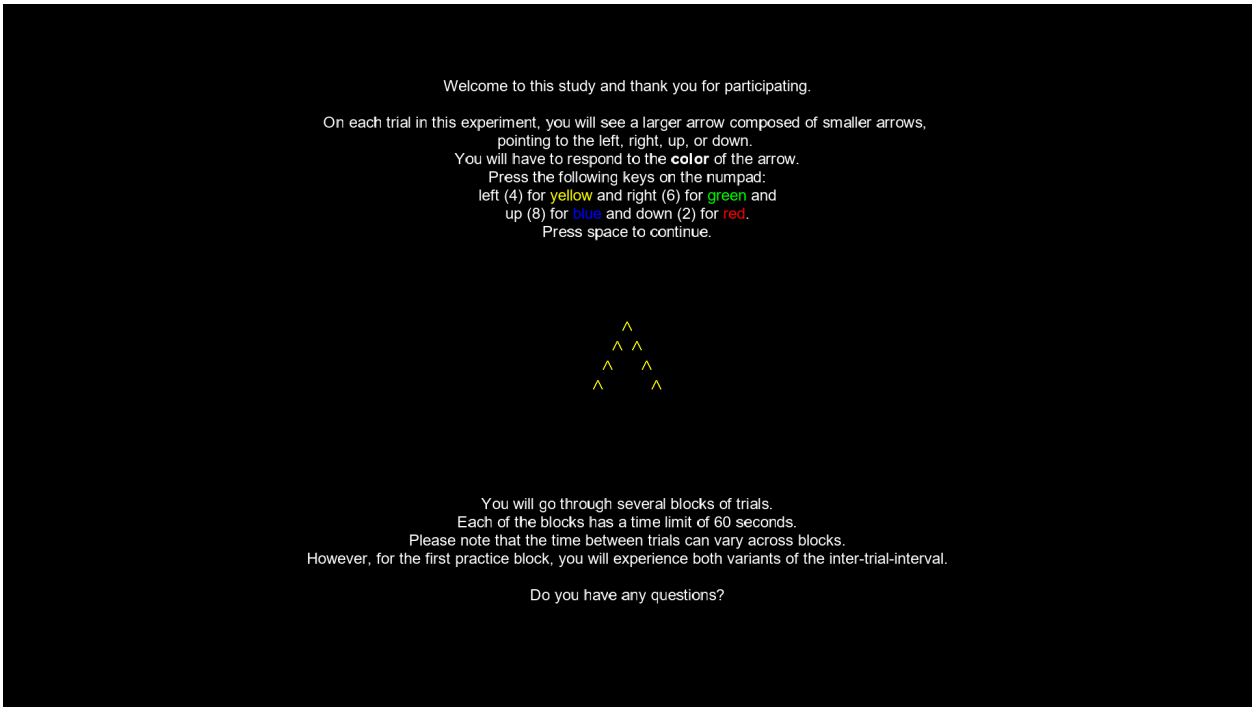

Figure 1. Caption

participants could earn/lose $0.01 per correct/incorrect trial (high reward) whereas in the other half, participants could only earn/lose $0.001 per correct/incorrect trial (low reward condition). Participants typically earned $4 - $5 throughout the experiment.

Second, I manipulated the inter-trial-interval - in half of the blocks, the time between trials with a blank screen was set to 0.5 s, in the other half to 1.5 s. With this manipulation, I changed the possible exploitation rate per block, which has an impact on the rate of exploration, following from this formula:

$$timecost_{exploration} = \frac{time_{exploration}}{time_{exploitation}}$$

This implies that if a trial (including the inter-trial-interval) is long, the cost of exploration is relatively low, hence, a higher overall rate of checking the videos (and thus, distractibility) should be apparent. In other words, if a trial takes longer to complete, the relative time cost of exploration (assuming that the time needed to explore remains

constant) is lower. In so far unpublished studies, our group confirmed that exploratory behavior do indeed vary according to this formula if the denominator is manipulated (Ahmad, Grätz, & Mayr, 2023; Grätz, Fröber, & Mayr, 2022; Grätz & Mayr, 2023).

Behavioral results and checking behavior

Before I am presenting the models, it is helpful to visualize distractability in the current sample, dependent on the experimental conditions. For an initial analysis, for each trial it was determined whether the subject inspected the video (coded as 1 if yes, 0 otherwise). For each condition and subject, the average of this variable was then calculated, representing the average video checking probability. Statistical results will not be presented here, but numerically, average effects are as expected - longer inter-trial-interval and lower reward were expected to reduce the cost of exploring the videos and this trend seems to be present in the data (Figure 2). This figure reveals that video checking behavior shows large variation between individuals.

Data preprocessing

All analyses were conducted using R (R Core Team, 2023) in RStudio (Posit team, 2023). For data processing, wrangling, and visualization, the rio (Chan, Chan, Leeper, & Becker, 2021) and tidyverse (Wickham et al., 2019) packages were used.

To model the degree to which people are distracted in these situations, I used a more fine-grained approach with the goal to predict distractibility for each video frame.

For each experimental session, two data files are present - eye-tracking data contains information about gaze behavior during the experiment, the behavioral data file contains trial-level information about the experimental conditions, responses and video shown.

I first prepared the eye-tracking data. The raw eye-tracking data consists of one row per millisecond with information about time stamp, and x and y position on the screen.

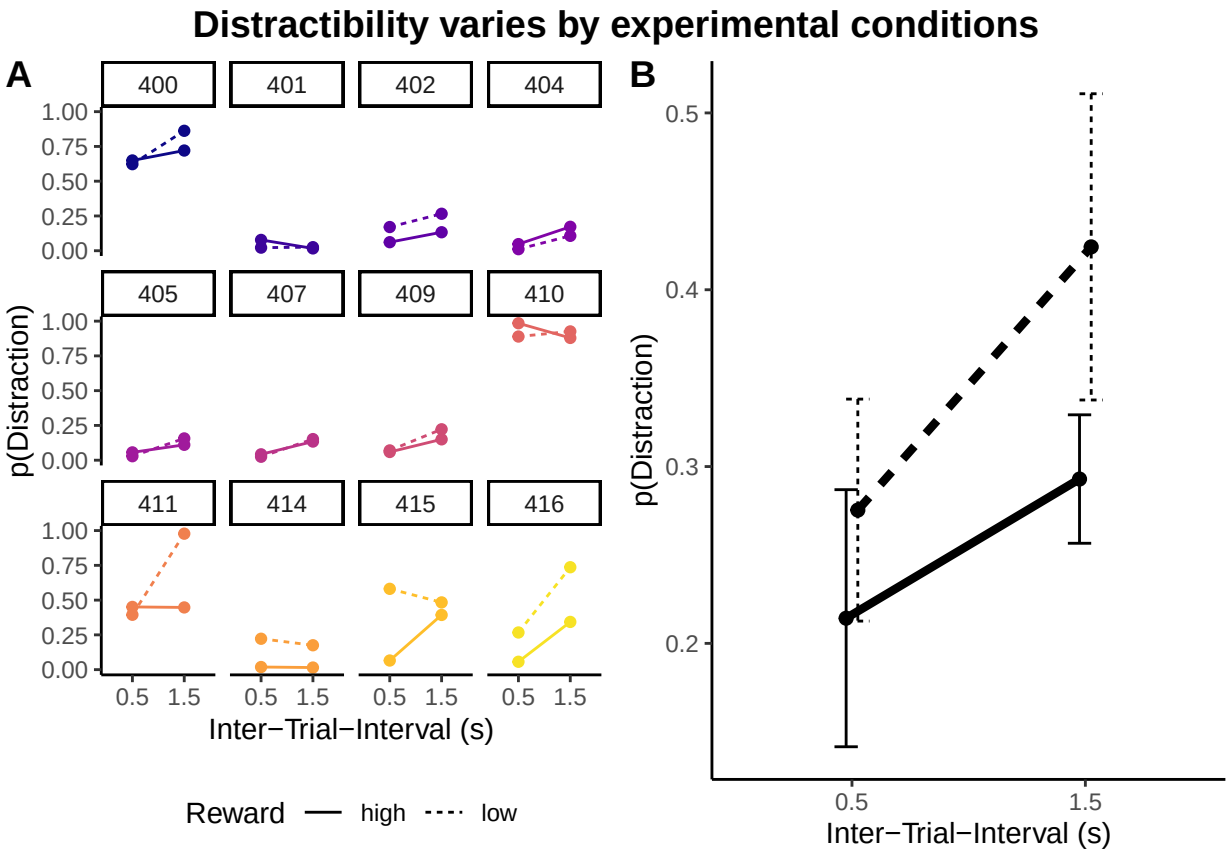

Figure 2. Behavioral results. **A:** Distractibility by subject. **B:** Errorbars represent within-subjects design corrected 95 % confidence intervals. Probabilities for distraction were obtained by dividing the number of trials with an gaze on the video by the number of trials, separately for each condition.

From the message data frame within the eye-tracking data, I identified information about experimental blocks, trials and video frame onsets and merged this with the raw eye-tracking data described above. The eye-tracking data file also provides information about blinks, most importantly blink onsets and ends. Time stamps during which blinks occurred were then removed from the enriched eye-tracking data frame. Now, for each millisecond, information about time (from trial onset), eye position, experimental blocks and trials and video frame number was present. I then merged the eye-tracking data with the behavioral data, specifically to add information about the video that was presented and

its location (left/right, close/far from the task stimulus). For each row, I then determined whether the gaze was inside the area the video was presented (coded as 1), or not (coded as 0) - this is my distraction indicator, and the outcome variable I aim to predict. This resulted in a data frame with highly redundant information because consecutive rows within a given frame could only differ with respect to the time stamp and the distraction indicator. I reduced the redundancy and the size of the data set by keeping only one row of per subject x trial x video frame. The distraction indicator is now coded such that one or more time stamps in which the gaze was inside the video ROI is a 1, and no video inspection a 0.

Obtaining video frame embeddings

Using ffmpeg and the imager (Barthelme, 2023) package in R, I saved each frame from each of the 500 videos as an image. Embeddings for each of these frames were then obtained using the reticulate package (Ushey, Allaire, & Tang, 2023) and the img2vec library (Safka et al., 2022). Resnet-18 was the model selected because its output vectors are shortest with a length of 512, relative to the other available models in the img2vec library.

The embeddings were then merged with the frame-by-frame data set above.

Description of the final data

> Description of the data (15pts): Describe core features of the data, any additional features you produced from existing features and how, basic descriptive statistics about these features, and any missing data analysis you conduct. The description should be sufficiently clear that the instructor understands all the variables included in your modeling.

The final data set consists of observations. Each observation represents a video frame presented in the experiment with the following (model-relevant) columns: ID, Time stamp,

ITI condition, Reward condition, the 512 embeddings (all predictors) and the outcome, whether the gaze was on the video (coded as 1) or not (coded as 0). For example observations without embeddings, see Table 1.

4

Description of the models

Apply at least three different modeling approaches to predict the outcome in the dataset. Describe any specific setting used during the model fitting (e.g., hyperparameter tuning, cross-validation). Also, discuss how you plan to evaluate model performance.

Models were built in R (R Core Team, 2023) using R Studio (Posit team, 2023) with the packages caret (Kuhn & Max, 2008), and glmnet (Tay, Narasimhan, & Hastie, 2023). Ridge regression, lasso regression and elastic net models were selected as predictive model candidates. All numeric predictors were standardized and factors were one-hot coded (subject ID, inter-trial-interval condition, reward condition). Models were trained on 90 % of the data with 10-fold cross-validation and evaluated on the remaining 10 % test set. For hyperparameter tuning, the respective parameters were initially selected to span a relatively wide possible range at which model fit is expected to be improved. Models were generated multiple times, each time reducing the range of the respective hyperparameters to the range at which model improvement was visible in the previous iteration to find the best hyperparameters. LogLoss was the metric by which the final model was selected for each procedure. Model performance on the test set will evaluated based on overall accuracy, false positive, false negative, true positive and true negative rates.

Table 1

The first ten rows of the training data set without frame embeddings. The training data set with frame embeddings is identical, but has 512 more columns in the format VX (where X stands for a number in the range [1; 512]) for the frame embeddings.

time	ID	ITI	RewardCond	isInVid
550.00	400	0.5	low	no_check
583.00	400	0.5	low	no_check
617.00	400	0.5	low	no_check
650.00	400	0.5	low	no_check
683.00	400	0.5	low	no_check
716.00	400	0.5	low	no_check
750.00	400	0.5	low	no_check
783.00	400	0.5	low	no_check
850.00	400	0.5	low	no_check
916.00	400	0.5	low	no_check

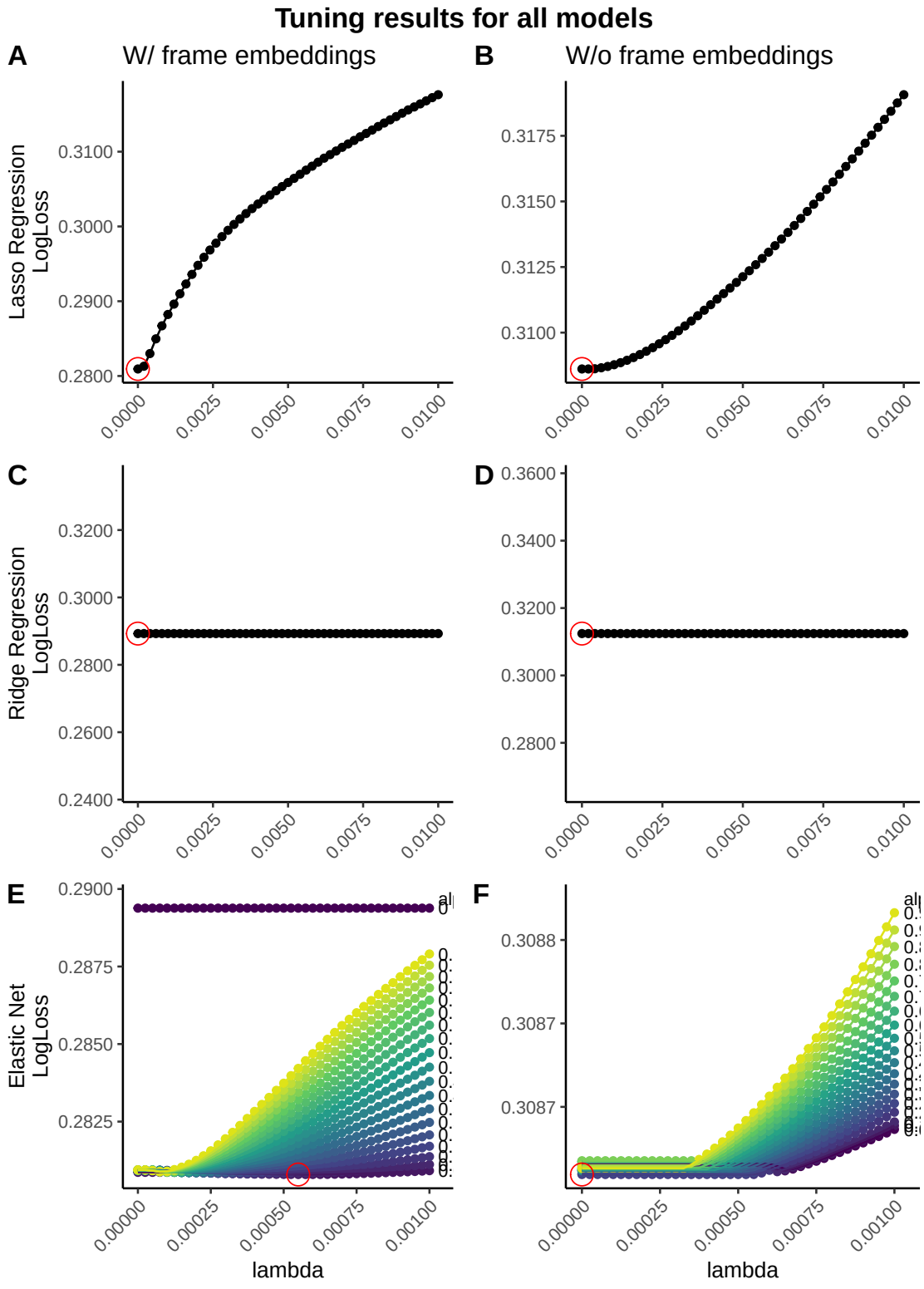

Figure 3. Tuning results for all models. The red circle represents the best tune out of the hyperparameter grid specified. Each row represent the tuning results for the regularized regression types (Lasso Regression: A, B; Ridge Regression: C, D; Elastic Net: E, F). Figures in the left column show the tuning results for the respective models including frame embeddings as predictors (A, C, E) or without frame embeddings as predictors (B, D, F).

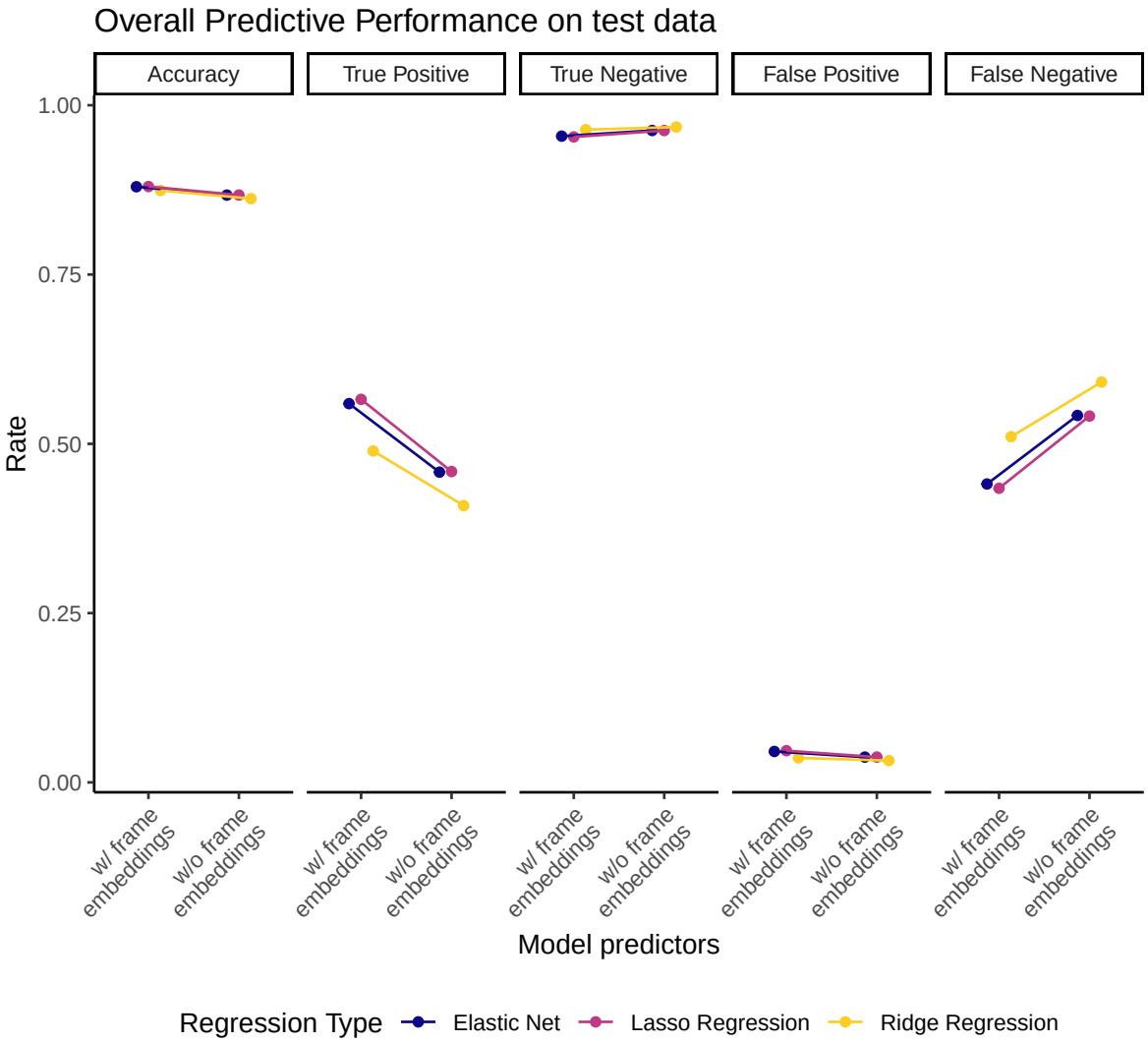

Figure 4. Performance of the regularized regression models with and without frame embeddings as predictors on the test set (10 % of the data). Performance on all metrics is better if frame embeddings are available as predictors on all metrics. This effect is emphasized for the true positive and false negative rate. Performance differences between regularized regression type are negligible for accuracy, true negative and false positive rates and are more pronounced for true positive rates and false negative rates.

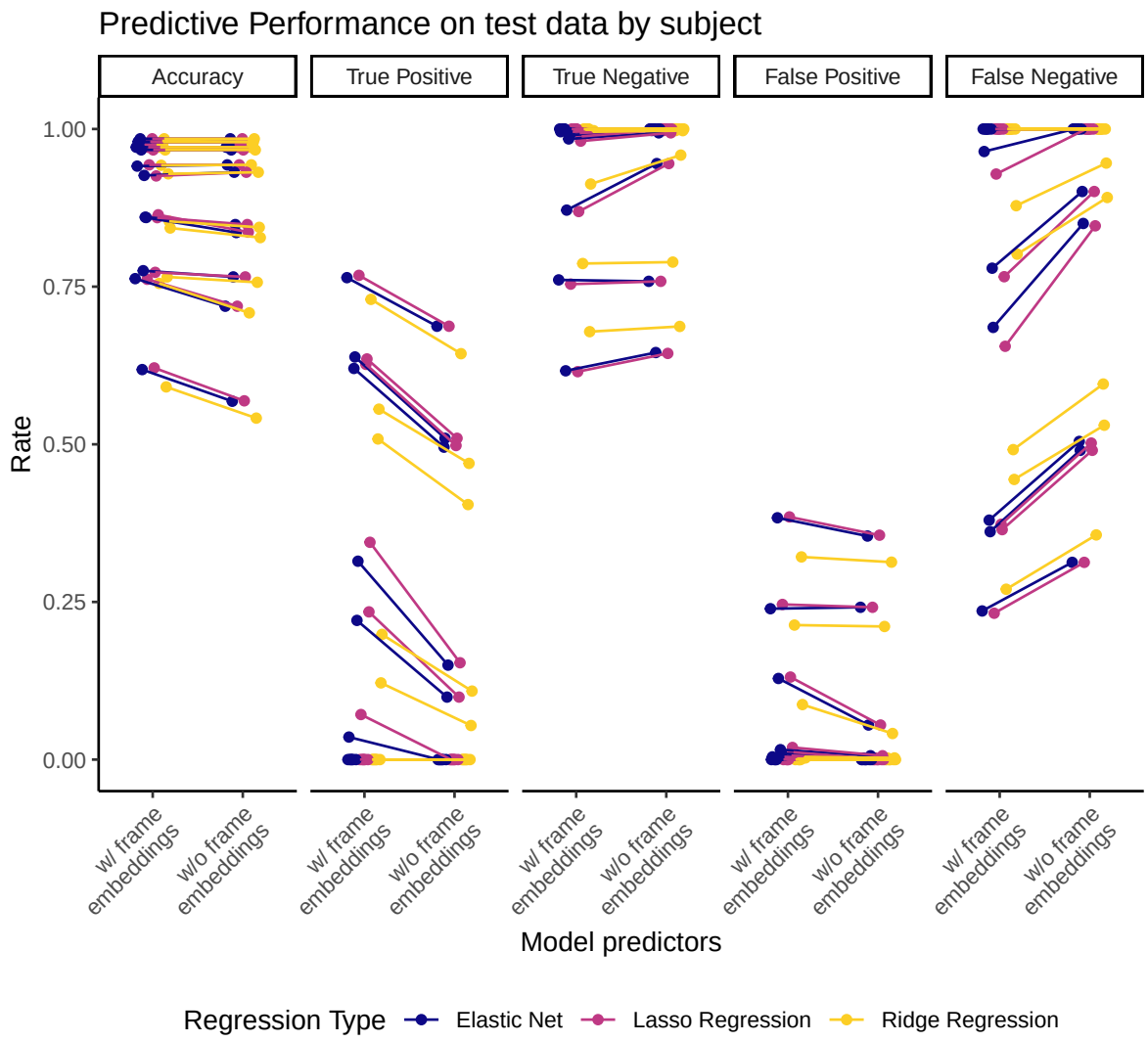

Figure 5. Performance of the regularized regression models with and without frame embeddings as predictors on the test set (10 % of the data), by subject. Note the differences in performance between subjects.

Model Fit

> Model fit (20pts): Provide the results of your model evaluation. Compare and contrasts results from different modeling approaches, including a discussion of model performance. Discuss your final model selection and the evidence that led you to this selection. If it is a classification problem, how did you choose a cut-off point for binary predictions? Did you consider different cut-off points?

The models were tested on 10 % data that was withheld from training the models. Accuracy, true positive, false positive, true negative and false negative rates were calculated for the entire test set, separate for regularized regression type (ridge regression vs. lasso regression vs. and elastic net) and predictors included (including frame embeddings vs. not including frame embeddings). Additionally, these metrics are also calculated within individuals so that model performance can be evaluated per subject.

Overall performance

Accuracy, true positive, true negative, false positive and false negative rates of all models are communicated in Figure 4. First of all, all models perform better when frame embeddings are added as predictors. Accuracy for all models is above 0.86. Performance differences between the regression types are negligible for the accuracy metric. While this initially sounds like a good model performance, it is important to evaluate the other metrics. Importantly, given the relatively low rate of video inspection (see Figure 2), we must consider the other metrics. The true positive rate for all models is above 0.41. Relative to all other models, lasso regression with frame predictors identifies most of the video checks, with a rate of 0.57. This rate is close to chance level which indicates that the models at hand perform poorly in this key metric. Performance of all models is good when it comes to identifying frames that were not checked (true negative rate), with a minimum of 0.95 across all models. Additionally, few frames were incorrectly classified as inspected,

with a maximum false positive error rate of 0.05 across all models. Performance is again poor for the false negative classification - errors are lower than 0.59 for all models, and the lasso regression model with video frame predictors performs best with a false negative rate of 0.43.

The pattern of true negative and false positive metrics indicates a strong bias of the model to classify a frame as not being checked. This is clearly driven by the baseline differences in video checking rate (which is relatively low, see Figure 2). Since true positives and false negatives, as well as true negatives and false positives are not independent metrics, the symmetric pattern we observe is not surprising. While models with frame embeddings generally do perform better than those without, this difference is only meaningful for the true positive rate and the false negative rate - for these measures, 'knowledge' of image content significantly improves the prediction of video inspection 23 %. It is also only for these metrics that performance differences by model become pronounced - lasso regression performs better than regardless of the availability of frame embeddings as predictors. However, even the best fitting model is still not very good at predicting frame-by-frame distraction. Additionally, it should be noted that the relatively high accuracy score cited above is misleading and driven by baseline differences in checking behavior.

Model performance on the subject-level

Model performance per individual for all models with and without frames as predictors is presented in Figure 5. There is a significant amount of between-subjects variability in how well the models can predict distractibility on a frame-by-frame basis. The range for accuracy is [0.54; 0.98], for the true positive rate [0; 0.77], for the true negative rate [0.62; 1], for the false positive rate [0; 0.38], and for the false negative rate [0.23; 1] across all models. Thus, for some subjects the model does relatively well, for some it does poorly. Interestingly, for nearly all subjects the model with frame embeddings

predicts distractibility much better on all metrics.

Discussion & Conclusions

> Discussion/Conclusion (25pts): Discuss and summarize what you learned. Which variables were the most important in predicting your outcome? Was this expected or surprising? Were different models close in performance, or were there significant gaps in performance from different modeling approaches? Are there practical/applied findings that could help the field of your interest based on your work? If yes, what are they?

Overall, model performance is better when frame embeddings are included as predictors, specifically, they improve the true positive rate and reduce false negatives significantly. The best regularized regression model on nearly all metrics and even within subjects is lasso regression with the tuning hyperparameters of $\alpha = 1$ and $\lambda = 0$. Even without frames embeddings as predictors, lasso regression performs best, with the tuning hyperparameters of $\alpha = 1$ and $\lambda = 0.00$. However, the predictive power of these models is severely limited, with a high false negative and low true positive rates.

Overall, information about the frame content as measured with embeddings is beneficial for the true positive and false negative rates, however, it is questionable whether the improvement on these measures justifies the computational effort necessary to obtain frame-wise video embeddings.

Where does the low performance come from? There are individual differences between subjects regarding distractibility, the degree they react to differences in Reward and inter-trial-intervals and personal interests. Given that one-hot coded dummy variables were included for each subject, overall distractibility is accounted for in the model. However, how people react to the experimental manipulations may depend on between-subjects variables, such as socio-economic status, the willingness to exert mental

effort, and self-control, among other variables. Additionally, subjects have different interests and it seems reasonable to assume that subjects will be more distracted by videos that seem more interesting to them.

An alternative to the current models would be models that are trained individually for each subject or models that include the interaction between subject, experimental variables and video frame embeddings. This approach will be particularly helpful for subjects who deviate from the average behavioral pattern shown in Figure 2. Moreover, the audio component of the videos was not used as a predictor in the current models. Future models could obtain more holistic representation of entire video sequences or the audio component and add these representations as additional predictors.

Another problem with the current models is that they assume that the gaze reacts instantly to a frame on the screen. It is unclear what exactly the saccadic reaction time would be in this task, but it typically is in the order of 200 - 400 ms (Braun, Weber, Mergner, & Schulte-Mönting, 1992). The relationship between frame embeddings and gaze could be explored using cross-correlation and identifying the time lag at which these two variables correlate the strongest. For the models, the frame embedding predictors could be shifted such that they have by that lag relative to the gaze data. As this increases the correlation between frame embeddings and gaze behavior, it should improve predictive model performance.

References

Ahmad, S., Grätz, D., & Mayr, U. (2023). *Rational distraction: The adaptive nature of reliance on external action-relevant information.* https://doi.org/10.13140/RG.2.2.11482.93125

Aston-Jones, G., & Cohen, J. D. (2005). Adaptive gain and the role of the locus coeruleus–norepinephrine system in optimal performance. *Journal of Comparative Neurology*, *493*(1), 99–110.

Barthelme, S. (2023). *Imager: Image processing library based on 'CImg'.* Retrieved from https://CRAN.R-project.org/package=imager

Brainard, D. H., & Vision, S. (1997). The psychophysics toolbox. *Spatial Vision*, *10*(4), 433–436.

Braun, D., Weber, H., Mergner, T., & Schulte-Mönting, J. (1992). Saccadic reaction times in patients with frontal and parietal lesions. *Brain*, *115*(5), 1359–1386.

Chan, C., Chan, G. C., Leeper, T. J., & Becker, J. (2021). *Rio: A swiss-army knife for data file i/o.*

Cohen, J. D., McClure, S. M., & Yu, A. J. (2007). Should i stay or should i go? How the human brain manages the trade-off between exploitation and exploration. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *362*(1481), 933–942.

Grätz, D., Fröber, K., & Mayr, U. (2022). *Does distraction make you slow, or does being slow make you distracted? Testing a rational choice model of consulting the environment for action-relevant information.* https://doi.org/10.13140/RG.2.2.12311.04004

Grätz, D., & Mayr, U. (2023). *Slower Response Speed Moves Us from Exploitation to Exploration.* https://doi.org/10.13140/RG.2.2.36435.96802

Inc., T. M. (2022). *MATLAB version: 9.12.0 (R2022a).* Natick, Massachusetts, United States: The MathWorks Inc. Retrieved from https://www.mathworks.com

Kleiner, M., Brainard, D., & Pelli, D. (2007). *What's new in psychtoolbox-3?*

Kuhn, & Max. (2008). Building predictive models in r using the caret package. *Journal of*

Statistical Software, *28*(5), 1–26. https://doi.org/10.18637/jss.v028.i05

Pelli, D. G., & Vision, S. (1997). The VideoToolbox software for visual psychophysics: Transforming numbers into movies. *Spatial Vision*, *10*, 437–442.

Posit team. (2023). *RStudio: Integrated development environment for r*. Boston, MA: Posit Software, PBC. Retrieved from http://www.posit.co/

R Core Team. (2023). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Safka, C., yuiseki, Yedidi, K., Kozłowski, M., keherri, Bar, N., & Cobanov, M. (2022). Image 2 vec with PyTorch. https://github.com/christiansafka/img2vec; GitHub.

Tay, J. K., Narasimhan, B., & Hastie, T. (2023). Elastic net regularization paths for all generalized linear models. *Journal of Statistical Software*, *106*(1), 1–31. https://doi.org/10.18637/jss.v106.i01

Ushey, K., Allaire, J., & Tang, Y. (2023). *Reticulate: Interface to 'python'*. Retrieved from https://CRAN.R-project.org/package=reticulate

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., ... Hiroaki Yutani. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686