Predicting High-level Cognitive Decisions using Low-Level Features from Eye Gaze Data

Timo de Vries

June 19, 2009

1 Introduction

"The eyes are the windows to the soul."
–William Shakespeare

This thesis aims to determine if high-level choices and performance can be predicted from eye movements. We investigated this using gaze data from people playing the game of SET and a collection of machine learning algorithms provided by Weka.

Visual attention is controlled by high-level *top-down* and low-level *bottom-up* processes in the human brain. Bottom-up processes are stimulus-driven, i.e. attention is "spontaneously oriented towards an oncoming stimulus" [7]. Top-down processes on the other hand are driven by cognitive decisions based on intentions. One needs to learn how these two processes interact to control eye movement and visual attention, in order to understand the mechanism that allows the detection of visual targets and patterns.

In our former research paper [4], we formulated several research questions in the field of visual attention and reasoning. At the beginning of this research project, we have selected five questions to investigate by means of an experiment. These questions can be found in Section 2. Our research makes use of a card game called SET. The basics of this game are explained in Section 3. Subsequently, our experimental setup is described in Section 4. The data from the experiments have been analyzed in different ways, as is explained in Section 5. The results are discussed in Section 6 and finally a conclusion is given in Section 7.

In short, the goal of this research is to investigate the link between high-level and low-level processes, by predicting choices and performance based on eye gaze data.

This research project was developed at the LIACS institute under supervision of Walter Kosters and Joost Broekens. The cognitive experiments were conducted in January 2009 at the Technical University of Delft.

2 Research questions

In our former research paper [4] we proposed several research questions. For this research, we have chosen to investigate five of them, all of which are strongly related to our main research questions. The main research questions and the five previously proposed questions are listed below.

Main research questions

- Does pop-out play a role in the game of SET?
- Can machine learning techniques be used effectively to predict if (*a*) the player thinks he or she has found a set (i.e., makes the decision to call “Set!”) and (*b*) he/she has found a correct set, based on low-level features from eye gaze data?

Related questions

1. Does experience with playing SET affect the quality of predictions based on eye movements?
   If this is true, there must be a relationship between low level features from eye movements, and high-level cognitive decisions based on experience.
2. Is the gaze path length (sum of distances between succeeding fixations), while one is playing SET, correlated to the player’s experience?
   If such a correlation exists, this would also point in the direction of a causal relationship between high-level experience and low-level features based on eye gaze data.
3. Is it possible to discern different search phases based on certain features in the gaze data?
   The idea of “search phases” comes from the separate processing phases in reading tasks: scanning, skimming, rauding, learning and memorizing [4].
4. Does experience in playing SET have any effect on the amount and/or quality of pop-out[1] detection?
   If there is a correlation between SET-experience and pop-out quantity and/or quality, then pop-out detection is probably a technique that is used more often, or in a better way, by experienced SET-players.
5. Is the ability to detect pop-out correlated to the gaze path length in a game of SET?
   Although pop-out occurs on a low cognitive level, the ability to detect it (faster) might very well be the result of high-level cognitive processes relying on experience.

[1] Pop-out is the visual cognitive effect that occurs when an item is standing out relative to neighbouring items [20].

3 Set

SET[2] is a card game with a deck consisting of 81 unique cards, which show different kinds of figures. Each card has four properties: the *number* of figures, their *shape*, their *shading* and their *color*.

The following section will explain the basic rules for SET. Next, Section 3.2 describes how the game is usually played. For a more in-depth examination of the game, the reader is referred to [4].

3.1 Basic rules

There exists a wide variety of variations on SET, all involving the concept of a "set". A *set* consists of three cards, satisfying *all* of the following conditions [21]:

- They all have the same number, or they have three different numbers.
- They all have the same shape, or they have three different shapes.
- They all have the same shading, or they have three different shadings.
- They all have the same color, or they have three different colors.

Figure 1 shows an example of a set.

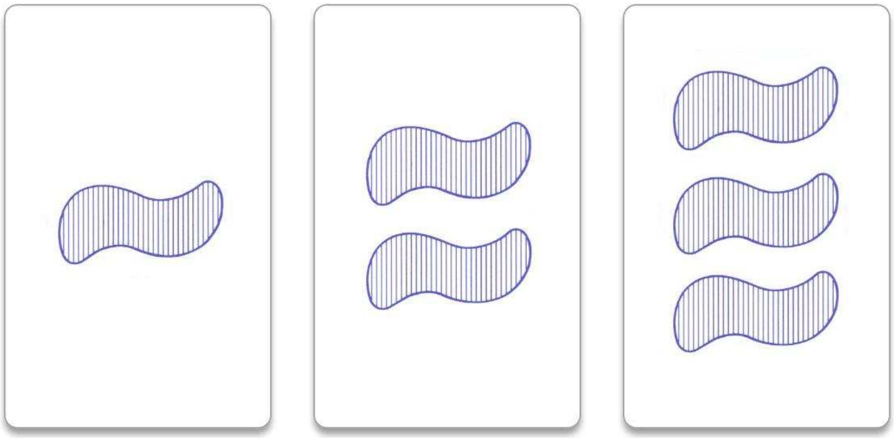


Figure 1: Example of a *set*: each card has a different number of figures, while the shape, shading and color are the same for each card.

[2] SET is a game by Set Enterprises: www.setgame.com

3.2 The game

SET can be played with two or more players. There is no maximum to the number of players. Usually, the dealer lays out twelve cards on the table. If one notices a set between those cards, he or she calls “Set!” and points out the cards that form the set. Once it has been verified as a set, the person takes the three cards, after which three new cards are laid out on the table. If there is no set within the twelve cards on the table, the dealer will add three more and the game continues. From now on we will call the cards on the table “the layout”. When the deck is empty and there is no set left in the layout, the player with the highest amount of sets is the winner.

3.3 Previous research

Making use of the game of SET, Niels Taatgen [9] let eight subjects compete against and evaluate three different cognitive models in order to determine which model was most human-like.

4 Experimental setup

First we will describe the different tasks in Section 4.1, then we will describe our sample group in Section 4.2.

4.1 Experiment tasks

The experiment consisted of three SET-tasks, interposed with dummy tasks such that the participants would not catch the true intent of the experiment. The first SET-task consisted of 10 randomly generated layouts[3] containing multiple sets. The same layouts were presented to each participant in random order. The second SET-task consisted of 5 layouts, randomly generated with strong constraints on the cards in each layout. Each layout contained exactly one set and was generated in such a way that by removing the set and replacing it with new cards, the next layout in the sequence could be formed. The same sequence of layouts was presented to each participant in the same order. The third SET-task consisted of 80 "layouts" containing only *six cards* each (two rows of three cards), generated randomly with strong constraints. The same layouts were presented to each participant in random order. Each layout contained at least four cards of the same color, four cards showing the same number of figures and four cards with the same pattern. 33 layouts did not contain a set and 47 layouts did, of which 27 had the cards forming the set *lined up* horizontally.

For each layout in the first two tasks, the goal was to find a set before the "opponent" would. The task description implicitly suggested there was a human opponent, though some did have questions about this. The "opponent" was programmed in such a way that after each layout it automatically adjusted its level towards the participant by either increasing or decreasing its response time. We have chosen the experimental setup to trigger the sense of competition, as time pressure and competition are important aspects that influence how SET is played by altering the strength of different attentional strategies (e.g., extensive search and comparison will be too slow a strategy to win the game in general). In the third task, each participant was given a short amount of time to assess each "layout" of six cards and press spacebar if he or she thought it contained a set. The time limit was decreased gradually down to 400 ms, which resulted in random performance. The primary goal of this task was to find out if pop-out detection works better when the cards forming the set are lined up in a row.

4.2 Sample group

Thirty-one individuals, mostly students, participated in our experiment. All participants had to perform the same SET-tasks in the same order. Each task consisted of multiple smaller tasks. Those smaller tasks, or "trials", were presented in random order, except for the second task in which the order could not be varied due to the chronological nature of the task.

Twenty-three participants had played the game at least once, while only eight participants had never played it before. We obtained a subjective experience score (SES) for all participants using the following formula.

[3] A "layout" is a collection of twelve cards laid out face-up in three rows of four.

$$SES = \left(\frac{Level}{8} + \frac{max(200 - DaysNotPlayed, 0)}{200} + \frac{FrequencyRate}{5} \right) /3$$

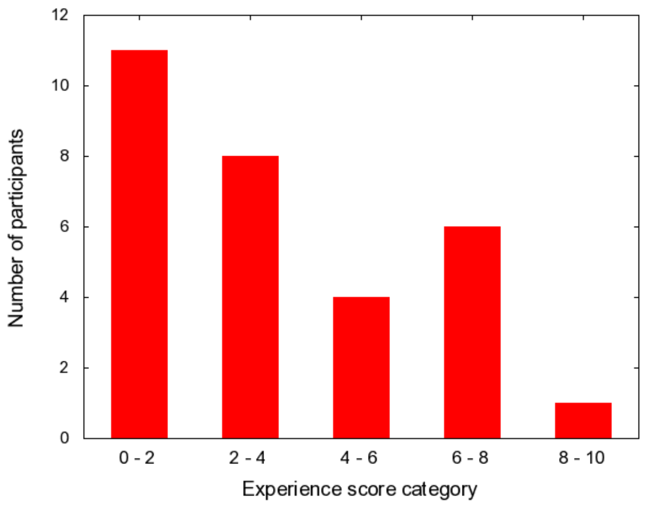


Figure 2: Distribution of the participants' SES.

Level is the subject's self-assessed level of expertise with the game. DaysNotPlayed is the number of days that have gone by since the participant played SET for the last time (200 if he or she had no experience with the game at all). FrequencyRate is a number from 0 to 5, indicating the frequency of playing the game *before* the last time the subject played it. We used the following frequency division:

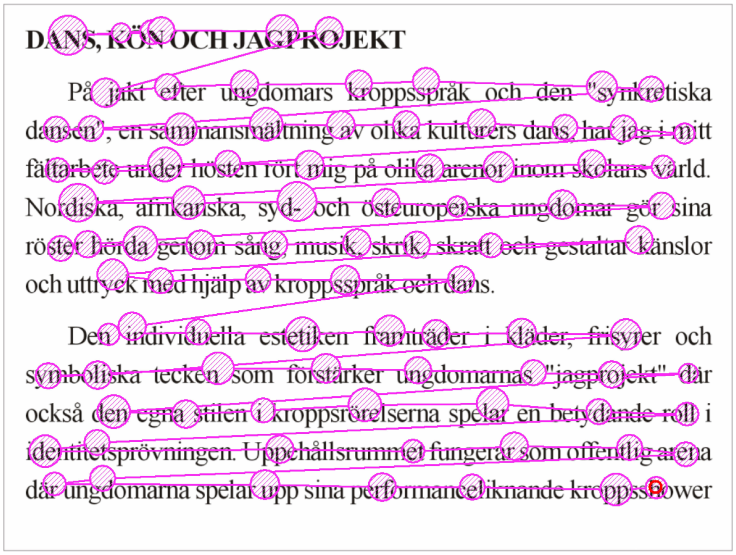
0. Never
1. Less than a few times a year
2. A few times a year
3. Once a month
4. Once a week
5. Several times a week

The distribution of the participants' SES is shown in Figure 2.

5 Data analysis

5.1 The raw data from tasks 1 and 2

Each of the thirty-one participants did two interleaved tasks of playing SET against a computer-simulated opponent, with a distractor task in between. First the eyetracker had to be calibrated. During both tasks, the participant's gaze position was captured every 20 ms and the coordinates of the focal position were written to a log file, along with a timestamp. Unfortunately, the log files show that at some points, several records are missing, sometimes even a hundred or more in a row, causing a "gap" in the record stream of a split second up to several seconds. Moreover, many participants' gaze records seem compressed and shifted in some direction, some more than others. This must be due to calibration errors (more on this in Section 5.6.1). For this reason, we have chosen not to rely on absolute coordinates for feature extraction, but rather on relative positions. As a result, the idea of matching positions to cards had to be discarded, leaving an even greater challenge of finding a relationship between eye movements *only* and other features such as calling "Set!" and indicating a correct set.



DANS, KÖN OCH JAGPROJEKT

På jakt efter ungdomars kroppsspråk och den "synkretiska dansen", en sammansmältning av olika kulturers dans, har jag i mitt fältarbete under hösten rört mig på olika arenor inom skolans värld. Nordiska, afrikanska, syd- och östeuropeiska ungdomar gör sina röster hörda genom sång, musik, skrik, skratt och gestaltar känslor och uttryck med hjälp av kroppsspråk och dans.

Den individuella estetiken framträder i kläder, frisyrer och symboliska tecken som förstärker ungdomarnas "jagprojekt" där också den egna stilen i kroppsrörelserna spelar en betydande roll i identitetsprövningen. Uppehållsrummet fungerar som offentlig arena där ungdomarna spelar upp sina performanceliknande kroppsshower

Figure 3: An example of fixations and saccades over text.

5.2 Preprocessing on data from tasks 1 and 2

A single gaze record, i.e., a set of coordinates with a timestamp, does not hold much information about eye *movement*. More is needed in order to distinguish different kinds of visual search behavior. Therefore we applied some *preprocessing* to the raw data.

Preprocessing is based on the assumption that certain patterns are hidden in the sequence of gaze records. The first step in preprocessing eye gaze data is usually to discriminate between *fixations* and *saccades* [14]. Fixations maintain the visual gaze on a single location, whereas saccades move it to a different location. See Figure 3 for an illustration of this. Since the majority of features will depend on fixations and saccades, it is important to have a good classification mechanism for fixations versus saccades. We have used a fixed distance threshold to distinguish saccades from drifts/tremors [5]. Once the fixations and saccades are identified, we can use them to calculate basic features like fixation duration and saccade distance as well as more advanced features such as velocity and saccade angle.

5.3 Features in tasks 1 and 2

Preprocessing on the raw data results in a number of features that can be used as input for a machine learning algorithm (MLA) in the next stage. The idea is to feed as many meaningful features to the MLA as one can extract from the raw data, given that they could somehow be related to meaningful patterns in attentive behavior. The MLA should be able to learn from the data which features are meaningful and which ones are not. A feature's usefulness might depend on the MLA's strategy, however, so in reality it is difficult to say which features are best and which ones should have been left out.

We applied a hard distance threshold of 80 pixels to distinguish saccades from fixations (i.e., a "jump" of more than 80 pixels was classified as a saccade). Based on this classification, we extracted the following features from the raw data. Two examples of *heat maps* that were used to extract the attention spread and maxima values are shown in Figures 4 and 5.

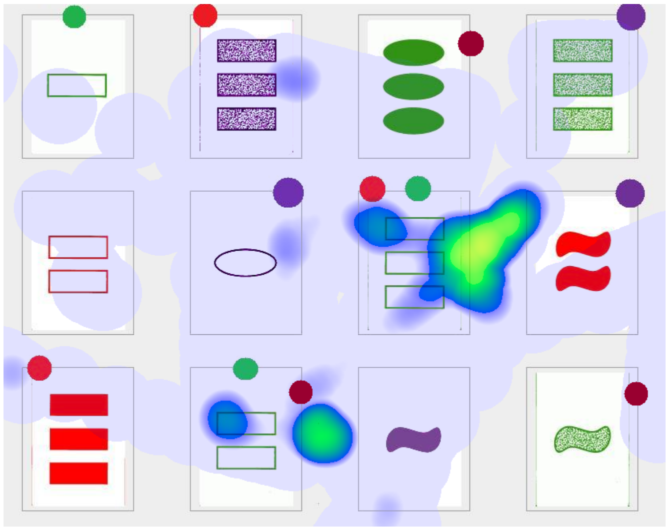


Figure 4: An example of a heat map showing the distribution of attention. The colored dots indicate the combinations of cards that form a set. The subject did not find a set in this case.

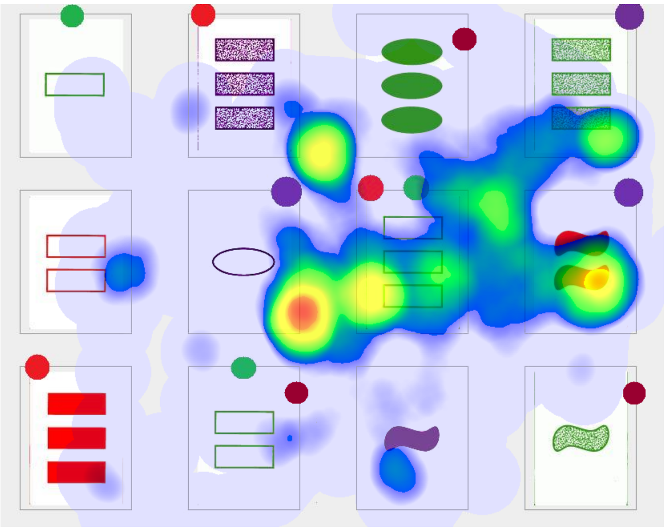


Figure 5: An example of a heat map showing the distribution of attention. The colored dots indicate the combinations of cards that form a set. The purple-dotted set was found in this case.

Fixation / saccade based:

- duration of fixation (in ms)
- saccade length (in pixels)
- saccade angle (in degrees)
- Δ saccade angle (the difference in degrees between the angles of two successive saccades)
- velocity (in pixels per second)
- Δ velocity, i.e., acceleration (in pixels per second)

Distribution based (depending on the perimeter of the “spotlight of attention”, which is used to create the attention heat maps; in our case the perimeter is 150 pixels):

- attention spread (in pixels)
- attention-intensity maxima (based on the time spent gazing at the 10 most frequently visited location)

Frequency distributions

A technique called *binning* [12] was used to obtain *frequency distributions* for each feature except attention spread. For each frequency distribution, we used 10 bins. For features in degrees and units of intensity we used 10 bins of the same “width” (i.e., a linear scale),

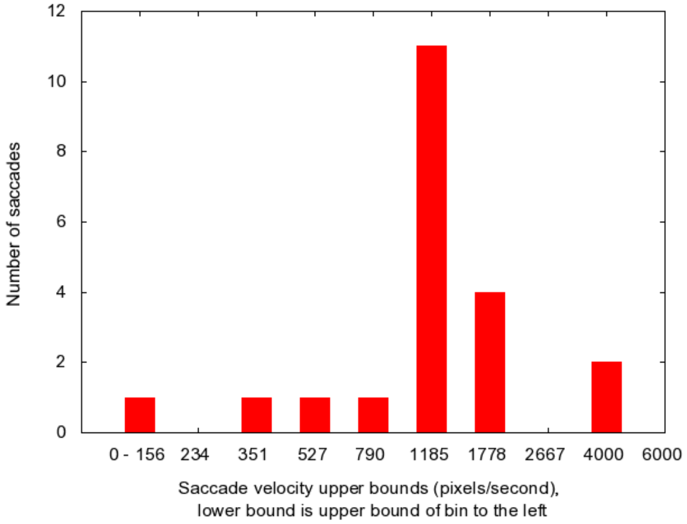


Figure 6: Frequency distribution of saccade velocities

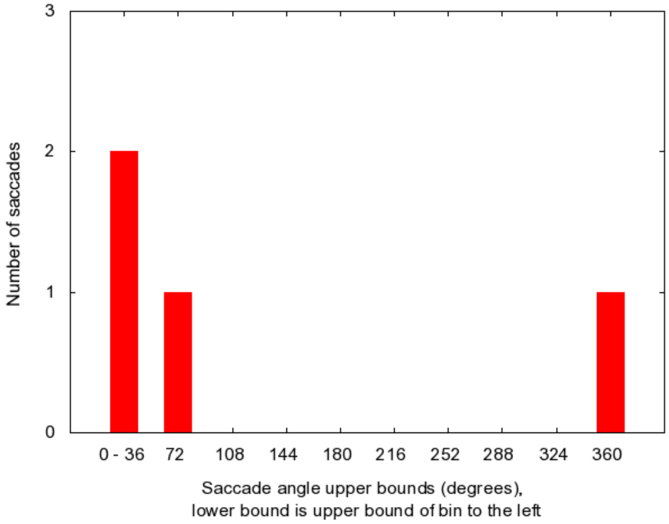


Figure 7: Frequency distribution of saccade angles

whereas other feature values were distributed on an exponential scale, with each bin having a lower bound equal to the upper bound divided by 1.5. We used an exponential scale in most cases because it produced a more even distribution, resulting in a richer feature set with less zero-values.

An example of both types of frequency distributions is given in Figures 6 and 7. Maximum values were used to determine the bin boundaries; any value above the maximum value was treated as the maximum value (i.e., added to the rightmost bin). The maximum fixation time was set to 3000 ms, the maximum saccade distance was set to 1500 pixels, the maximum velocity was set to 6000 pixels per second and the maximum attention intensity was set to 250 units.

5.4 Machine learning

For our research, we made use of the Weka workbench [23] to build several predictive models[4] using different machine learning techniques. Weka incorporates a great number of different *classifiers*, enabling the user to build and train a predictive model in a matter of seconds. This makes it relatively easy to estimate the “predictive power” of a combination of features.

Since the emphasis of this research lies not on classifiers but on *features*, we will not go into detail about the different kinds of classifiers. We selected six classifiers representing a diverse set of standard, easy-to-use machine learning techniques. We purposely used all of these classifiers in each test, even when one or more classifiers performed significantly better or worse than others, so that the results would not form a particular artifact due to the classifier that was chosen.

The following six classifiers were used.

- NaiveBayes: a Bayesian classifier [18]
- LWL: a *locally weighted learning* [2] algorithm using nearest neighbour search
- Bagging: a *meta learning* [16] algorithm that uses a *decision tree* [13] algorithm called REPTree
- Dagging: feeds data chunks to a copy of another classifier, in our case SMO [22] (sequential minimal optimization)
- BayesNet: a Bayes Network [10]
- MLP: a Multi-Layer Perceptron [17]

5.5 Prediction models

We trained each prediction model on data from task 2, using 10-fold cross-validation in order to prevent overfitting[5] as much as possible. The models were trained on each segment size (1, 2, 4, 6, and 10 seconds) separately and then averaged. Each training was done 5 times to compensate for models that do not have a deterministic output, so each final accuracy result is based on $5 \times 5 \times 6$ predictions.

Finally, we used standard statistics to detect relations between performance on the pop-out task on the one hand, and eye movement behavior and performance in the first task, as well as the Subjective Experience Score.

Some instances (i.e., single examples from the dataset) had to be removed because they did not meet the requirements for a specific classification accuracy test. For example, when using a 10-second interval of gaze-data, all trials with a length less than 10 seconds had to be removed.

[4] There are many different kinds of predictive models, some of which are listed here: `http://en.wikipedia.org/wiki/Predictive_modelling`

[5] In machine learning, *overfitting* means that the model is adjusting to “very specific random features of the training data, that have no causal relation to the target function.” [19]

For task 2, we distinguish three different trial outcome types:

A. Opponent called “Set!”

B. Player called “Set!” but did not indicate a correct set

C. Player called “Set!” and indicated a correct set

Class A will be referred to as “non-push”, class B as “incorrect” and class C as “correct”. Note that B and C together constitute “push”.

Since each layout contained at least one set, every trial ended with one of the above outcomes. The following table shows the distribution of all data instances over the different outcome types, for task 2.

A (38%)	B (16%)	C (47%)

First, in order to test if the action of calling “Set!” can be predicted based on our features, we defined classification $\mathcal{C}_1$ as A versus B $\cup$ C (i.e., B and C were considered as equal in this test). Second, in order to test if performance (i.e., finding a correct set) can be predicted based on our features, we defined classification $\mathcal{C}_2$ as A $\cup$ B versus C. Third, in order to determine whether an instance of type B is more often mistaken for an instance of type A or an instance of type C, we divided the dataset in two more ways: $\mathcal{C}_3$ and $\mathcal{C}_4$. For $\mathcal{C}_3$, all non-keypress instances were removed, so only types B and C were left. For $\mathcal{C}_4$, all instances of type C were removed such that we only had instances of type A and B. By comparing the accuracies of our models in predicting $\mathcal{C}_3$ and $\mathcal{C}_4$, we might be able to determine whether B-instances are more similar to A-instances or C-instances, in general.

Before drawing any conclusions however, we should compare the resulting accuracies with the accuracies that could be obtained by predicting the majority class (i.e., the one that is most frequently present in the dataset) in all cases. For predicting $\mathcal{C}_3$, the dataset contained 93 instances of which 23 were of type B and 70 were of type C. For predicting $\mathcal{C}_4$, the dataset contained 79 instances of which 56 were of type A and 23 were of type B. These ratios are far from ideal, but with each instance that is being removed or duplicated, the dataset becomes less diverse and hence less valuable. For this reason, we decided to leave the datasets as it is.

The four classifications are listed in Section 6.3, the results for each classification will be compared in Section 6.4.

5.6 Prediction problems and solutions

This subsection discusses a few problems with the previously described prediction method and propose a solution to each of them.

5.6.1 Calibration errors

Almost all feature values can be distorted by calibration errors, some more than others. Along with the possibility of having tracked a more or less “lazy eye” [15], this could lead to problems and/or biases [11] in predicting decisions and performance.

5.6.2 Segments are duration-dependent

Hoping to uncover different search phases within each trial, which could increase prediction accuracy, we decided to divide each trial into a fixed number of *segments*. At first sight, normalization over time[6] seemed to have eliminated the implicit "trial duration feature". Unfortunately, even after normalization, segments still retained a time-dependent aspect. The following example illustrates this. Participant A may have spent two minutes on a layout while B has spent only two seconds looking at it. In this case, using 10 segments, A's segments would be about 12 seconds long, while B's segments would cover only 200 milliseconds. Even after normalization is applied, some differences will inevitably remain. For example, the heat maps of A will most probably cover a broader area and show more tops than those of B. Such a bias poses a threat to the reliability of our predictions and must therefore be eliminated.

One solution to this problem is to define a *fixed time interval* and use the gaze data falling in that particular interval. E.g., using the last three seconds before someone called "Set!". This effectively removes the possibility of having an implicit time-dependent feature in each instance.

5.6.3 Gaze fingerprints

Another possible bias is what we call "gaze fingerprints": a combination of (approximate) feature values that distinguishes a certain subject from all other subjects. Such a combination of features could betray the subject in question, and an ML model can use that to base its predictions on the subject's "fingerprint" instead of the features that vary within this subjects' trials.

For example, if a model learns to identify subject Peter based on his "gaze fingerprint", and it has also learned that Peter usually calls "Set!" before the "opponent" does, then $\mathcal{C}_1$ (key-press) will get a high success-rate if the model simply predicts "true" for key-press in each trial belonging to Peter. If Tom is a slow player who usually *doesn't* call "Set!" before the opponent does, and the model is able to recognize Tom's trials based on his "gaze fingerprint", then it will get a high success-rate by predicting "false" for key-press in each trial belonging to Tom.

If the model can do this for all subjects at the same time, it has learned something useful. However it is actually not predicting key-press, but recognizing the subject who is playing the game and predicting the same outcome to his or her trials over and over. Assume we would use this model to classify new data instances from *unseen* subjects. If it has only learned to classify trial outcomes for "known" participants, its accuracy on the new data instances will probably be much worse than it was on the old data. It might even drop to around 50% (i.e., pure chance). Furthermore, if Peter would now play against a better opponent (who is faster than Peter), the model will keep predicting "true" for each of his trials, hence obtaining an even lower accuracy, unless the model is trained on these new instances first.

According to Erren *et al* [6], we should believe and doubt our hypotheses at the same time. Assuming that this kind of "fingerprint detection" is applied in the most rewarding manner, classifying those subjects who called "Set!" more often than the "opponent" as

[6] Normalizing over time is dividing a number over its corresponding time duration, thereby removing the time factor. Calculating a velocity is an example of normalization over time.

callers and those who called less than the opponent as *non-callers*, this hypothetical phenomenon would lead to a correct-rate of 0.71 for predicting $\mathcal{C}_1$, given our subject group and their actions in the second task. In the same way, the model might reach a correct-rate of up to 0.72 for predicting $\mathcal{C}_2$ in this task. This means that any model achieving higher correct-rates than these in classifying $\mathcal{C}_1$ and $\mathcal{C}_2$, must be using more than the subjects' "gaze fingerprints" to base its predictions on.

6 Results and discussion

This paper investigates two primary questions. First, does pop-out play a role in the game of SET, and second, can machine learning techniques be used to predict, based on eye movements, if (*a*) the player thinks he or she has found a set (i.e., calling "Set!") and (*b*), he/she has found a correct set.

Section 6.1 explains why some subjects were identified as outliers and therefore excluded from the analysis. Section 6.2 investigates the role of pop-out in Set. Section 6.3 lists the different kinds of classifications we made in order to investigate our second main research question (i.e., if machine learning models can make accurate predictions based on eye movements). Finally, Section 6.4 discusses our test results accordingly.

6.1 Outliers

In order to do an analysis of the correlations between features like gaze path length and experience, some subject's data had to be removed because they were outliers, i.e., so very different from the average that they may distort the correlations.

Participant 11's average "velocity" (gaze movement over time) during both tasks was significantly higher than the average velocity within the subject group, therefore this data record had to be left out of the correlation analysis. According to Munoz *et al* [8], people who have ADHD –like participant 11– have more involuntary saccades, hence the difference in velocity. Participant 6 spent significantly more time on the second task than the average time that was spent on this task, and was therefore also identified as an outlier, but only in the second task.

6.2 Pop-out in Set

Based on our experimental data, it seems that pop-out indeed plays a role in the game of SET. First, we obtain a significant positive correlation (0.53, $n = 30$, $p < 0.05$) between the Subjective Experience Score as obtained from the questionnaire and the amount of correctly identified sets in the pop-out task (pop-out score). This indicates that players with more experience are better at quickly identifying sets. This is an obvious result, and in line with [3]. Secondly, we find a significant negative correlation (-0.63, $n = 30$, $p < 0.05$) between the total gaze path length in the first task and the pop-out score, as well as a positive correlation between scores on the first task and pop-out score (0.58, $n = 30$, $p < 0.05$).

We interpret these findings as follows: experienced SET players perform better at quickly identifying sets, because they detect patterns more efficiently. As a result they move less with their eyes. As the pop-out task was constructed in such a way that visual search was practically impossible, the best explanation for the performance in this task is that the set pattern is detected due to pop-out. This would mean that, probably as a result of training, complex rule-based patterns and not only simple perceptual-feature based patterns can be detected via pop-out.

Please note, however, that an important characteristic of pop-out is that the speed of detection is more or less constant with respect to the size of the search space and the number of objects therein. This means that we can not claim the responsible mechanism is pop-out, as this would imply that in a very large layout (e.g., containing 100 cards) a person would still be able to detect a set immediately. Obviously that is not possible, unless by chance that set would be consisting of, e.g., one colour while all other cards in the layout have a different colour. This, however, can be explained by normal pop-out based on simple visual features such as color and has nothing to do with rule-based patterns. Probably a better term for what we found is fast, automatic and rule-dependent pattern recogntion.

6.3 Classifications

Prediction accuracy was tested for each of the following classifications.

$\mathcal{C}_1$ (A vs. B∪C): "non-push" vs. "correct" or "incorrect", to predict the decision to push the space bar if a set seems to be found,

$\mathcal{C}_2$ (A∪B vs. C): "non-push" or "incorrect" vs. "correct", to predict correctly identifying a set,

$\mathcal{C}_3$ (B vs. C): "incorrect" vs. "correct", to predict if –after the decision to push– a correct set is indeed indicated, and

$\mathcal{C}_4$ (A vs. B): "non-push" vs. "incorrect", to predict the type of error made by the participant (not found or incorrectly indicated).

6.4 Prediction results

Here we present the outcome of several classification accuracy tests with the same set of instances (or the relevant subset), using different features and parameters, and discuss the results with respect to our research questions.

6.4.1 Prediction accuracies for data from task 2, using different parameter settings

Using different parameter settings during feature extraction, we obtained the following results.

All classifiers performed equally well *with* the translation/scaling parameters applied to the raw data as without. Apparently, calibration errors did not influence prediction accuracy.

For all class types, accuracy was slightly better if *every pixel* that was looked at was counted to obtain the *attention spread* (i.e., how much space was covered by the subject's attention field), than when only those pixels with an "attention intensity" of at least 0.12 units were counted. This means the spread feature must be important to some or all of the selected models.

Taking either the raw data from the whole trial, or only the last segment (i.e., proportion of the trial), we see an *increase* in accuracy for each class type with $1 - 3$ percent points, as illustrated in Figure 8. Considering the fact that accuracy usually *decreases* when using more than the last 1 to 6 seconds of data (as indicated in Figures 9, 10, 11 and 12), an increase –even with a few percent– is an astonishing result. However, due to the time constraint of the task (i.e., the opponent calling "Set!" after a certain amount of time), we think it is better to rely on predictions based on feature values extracted from a

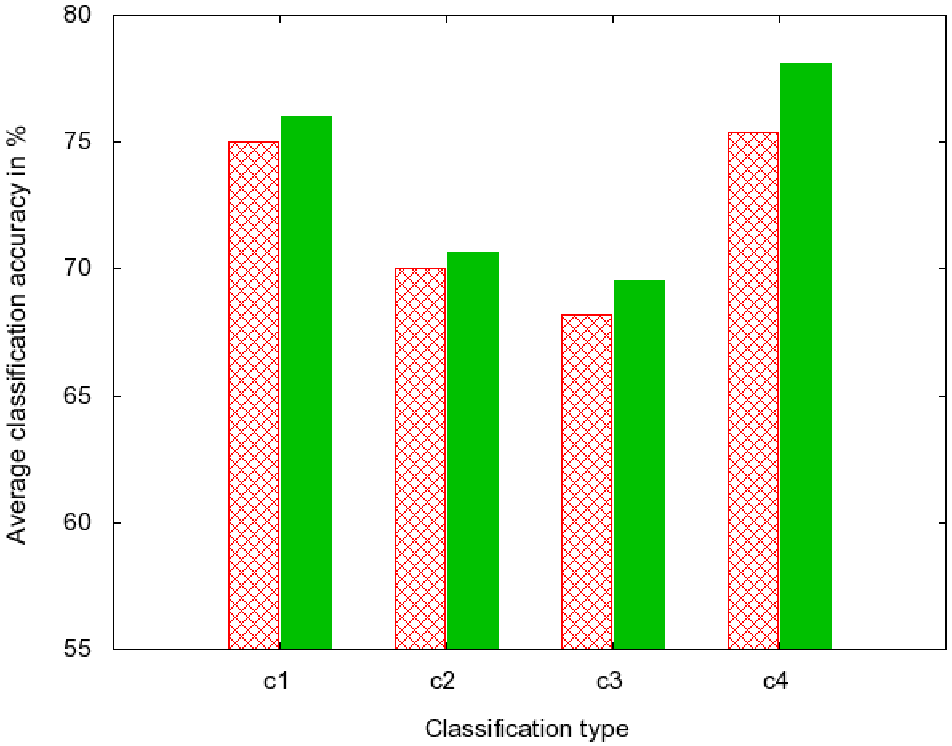


Figure 8: Classification accuracy using a fixed time interval (pattern bars) and the whole layout (solid bars), for each classification type.

fixed time interval, as we did in most accuracy tests.

In order to see if there was a learning effect during the experiment (i.e., if participants became better at finding sets over time), we added the corresponding layout number as a feature to each instance (i.e., the order in which the layout appeared to the participant) to see if it would improve classification accuracy. On the contrary however, adding the chronological layout number led to a tiny *decrease* in accuracy for most classifications. The effect it had (in percent points) on each classification is as follows. $\mathcal{C}_1$: -0.33%, $\mathcal{C}_2$: -0.18%, $\mathcal{C}_3$: $+0.03\%$ and $\mathcal{C}_4$: -0.22%. Hence we conclude that there either was no learning effect during the second task, or

6.4.2 Accuracies for task 2, with different interval lengths

Figures 9, 10, 11 and 12 show the resulting accuracies (success rates) for $\mathcal{C}_1$–$\mathcal{C}_4$ when using different interval lengths.

6.4.3 Predicting decisions

With regards to predicting decisions based on eye tracking data we found the following. For interpretation of the accuracy we use the performance of the naïve majority class predictor. This predictor acts as a baseline. Detailed outcomes of the experiments can be found in Table 1 and Figure 14. When the eye tracking data features were used (red bars in Figure 14), the average accuracy for the prediction of the decision to call “Set!” ($\mathcal{C}_1$, $75\,\%$)

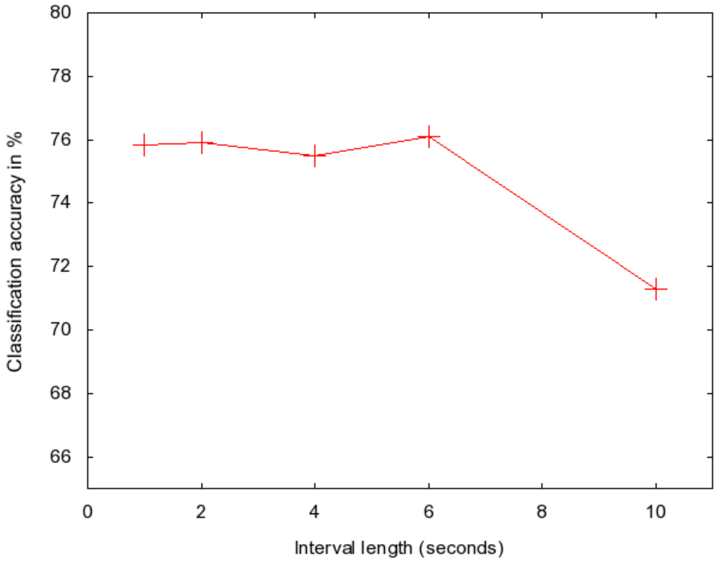


Figure 9: Average accuracy predicting $\mathcal{C}_1$ using different time intervals.

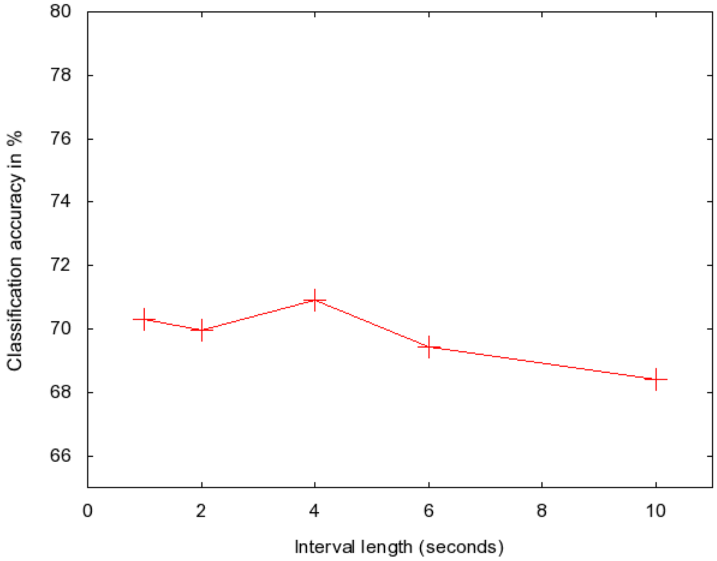


Figure 10: Average accuracy predicting $\mathcal{C}_2$ using different time intervals.

was well above the baseline. This means that even when features abstract away from absolute eye-gaze location and time needed for finding a set (two potentially strong predictors), there is information about a participant's intention to call "Set!". Eye movement therefore tells us something about the decision to call "Set!".

In order to test if the models had overfitted the training data, we tested the models (which was trained on data from task 1) on data from task 2. To our surprise, the accuracy did not decrease, as one would expect, but even *increase* (for small time intervals) or stay the same. This result is shown in Figure 13. The increase is probably related to the higher proportion of "push" in the task 1 (78.3% as opposed to 62% in task 2), meaning that

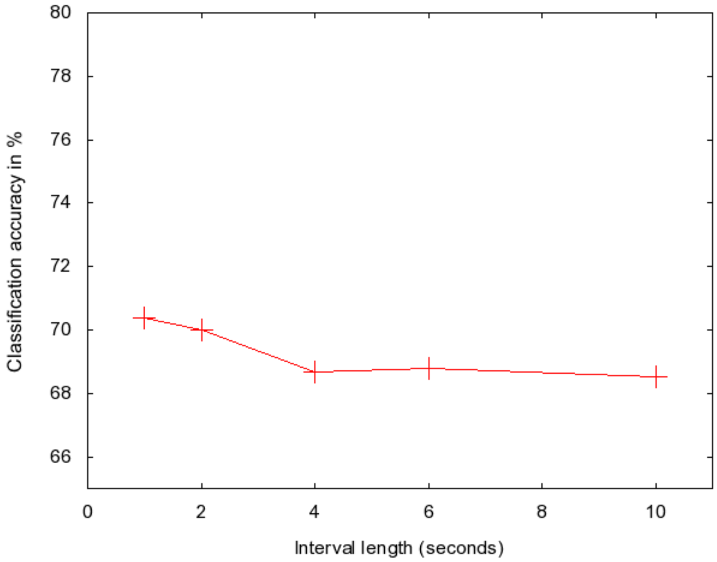


Figure 11: Average accuracy predicting $\mathcal{C}_3$ using different time intervals.

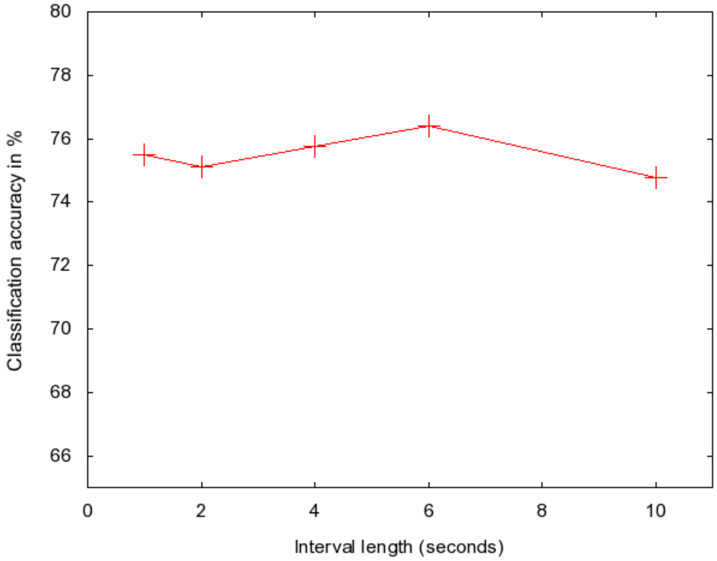


Figure 12: Average accuracy predicting $\mathcal{C}_4$ using different time intervals.

the models are probably better at correctly identifying “push” based on certain patterns in the gaze data, than at identifying “non-push”. However, this is only true for intervals of 1 to 4 seconds, supporting the idea that the last few seconds of eye movements are most important for predicting “press”, since that is when the decision is made to call “Set!”.

6.4.4 Predicting performance

Initially it seems that the models trained on eye movement data also predict the correctness of the set that was found ($\mathcal{C}_2$). However, when the model is trained to predict the correct-

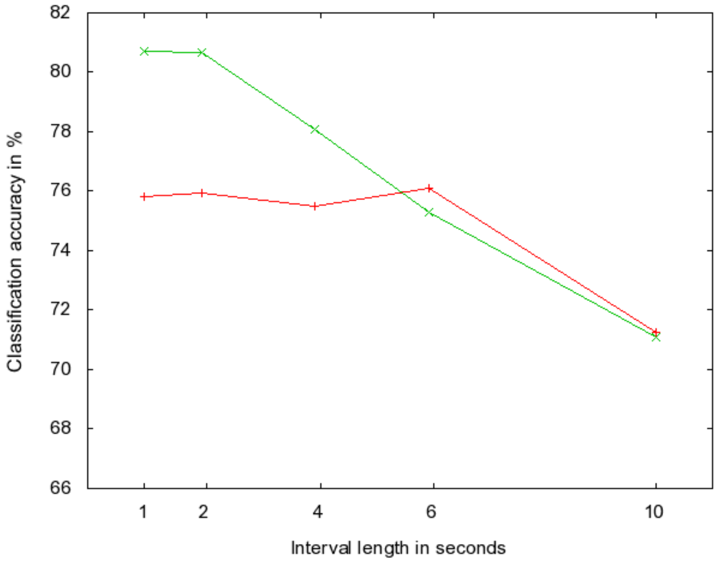


Figure 13: Average accuracy predicting $\mathcal{C}_1$ using different time intervals. Red line: accuracies when training and testing on the same data from task 2. Green line: accuracies when trained on data from task 2 and tested on data from task 1.

ness, given that it knows that the participant has called "Set!" ($\mathcal{C}_3$, correct vs. incorrect), the accuracy is poor. The good performance of the models when predicting correctness ($\mathcal{C}_2$) is thus a result of the good performance of predicting the decision to call "Set!". This means that, based on the currently selected features, there is no difference between thinking that a set is found and finding an actual set. This makes sense, given that the eye movements of participants just before deciding to call "Set!" probably reflect their *impression* that they have found a set: the participant actually believes he/she has found a set. On the other hand, if the experience score of a subject is used (SES[7] and OES[8]) to predict correct vs. incorrect set identification ($\mathcal{C}_3$), this does seem to outperform the baseline predictor (although marginally). This is plausible, as a participants' experience is probably a good indication that he/she has found a correct set instead of an incorrect one.

The interpretation that the Machine Learning models indeed predict the action of calling "Set!" is also shown in Figure 9. The features from the eye-tracking data in the intervals from 1 to 6 seconds make a fairly stable predictor for the decision of calling "Set!", while when the last 10 seconds are used, the accuracy drops significantly.

The claim that eye movement data does not show the difference between correct and incorrect set identification is supported by the fact that, based on eye tracking data features, the accuracy of predicting an incorrectly identified set versus not finding a set ($\mathcal{C}_4$) is comparable to predicting the decision to call "Set!" in the first place. This is obvious, as it essentially is the same as predicting the calling of "Set!" ($\mathcal{C}_1$), with the only difference that the claim was incorrect. As the eye movements do not reveal anything about the set (i.e., whether it is a correct one or not), one would expect to see this similarity.

[7] Subjective Experience Score, as defined in Section 4.2.

[8] Objective Experience Score, the number of correctly identified sets during the task, i.e., class C.

	gaze data only				scores only				all data			
	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_4$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_4$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_4$
NaiveBayes	74.6	68.1	48.9	73.2	59.2	70.7	84.6	68.5	74.3	69.9	52.3	73.5
LWL	72.5	69.8	68.8	73.6	56.7	62.1	79.2	66.4	72.7	69.9	77.6	74.3
Bagging	77.9	73.2	73.8	77.8	56.5	69.7	80.7	69.1	78.1	75.2	80.1	78.3
Dagging	73.1	66.0	74.2	73.9	62.4	65.6	75.4	70.9	71.6	69.7	75.0	74.3
BayesNet	76.6	72.6	74.8	78.1	62.4	65.4	76.1	70.9	75.7	72.9	75.4	78.5
MLP	74.9	69.2	75.1	76.5	62.4	54.5	75.4	70.9	74.3	74.2	74.9	77.9
average	**75.0**	**69.8**	**69.3**	**75.5**	**60.0**	**64.7**	**78.6**	**69.4**	**74.5**	**72.0**	**72.5**	**76.0**

Table 1: Classification accuracies for the classifications tasks $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ and $\mathcal{C}_4$, when using gaze data only, with performance/experience scores only, and for all features, with Naïve Bayes, Locally Weighted Learning, Bagging, Dagging, Bayesian Network and MultiLayer Perceptron.

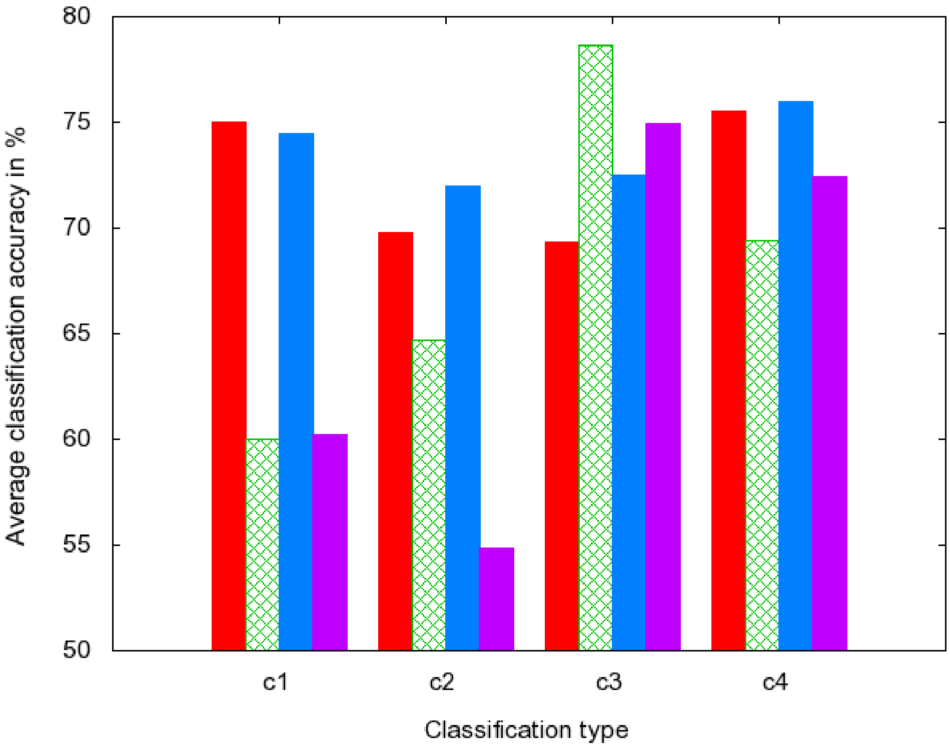


Figure 14: Average classification accuracy of the six machine learning methods with eye-tracking data features from task 2 only (red solid), performance/experience scores only (green pattern) and using all features (blue solid), for each classification task $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ and $\mathcal{C}_4$. The purple columns indicate the performance of the naïve majority class predictor and act as a baseline.

7 Conclusion

In this research, we have investigated the following two questions. First, does pop-out play a role in the game of SET, and second, can machine learning techniques be used to predict, based on eye movements, if (*a*) the player thinks he or she has found a set (i.e., calling "Set!") and (*b*), he/she has found a correct set.

Our results seem to indicate that pop-out indeed plays a significant role in the game of SET. However, this should not be called pop-out, but "fast automatic rule-based pattern recognition", as detection time will in most cases depend on the size of the search space (i.e., the number of cards on the table). Therefore we conclude that not pop-out, but fast automatic pattern recognition plays an important role in the game of SET.

Our second question is two-fold. Concerning subquestion (*a*), the action of calling "Set!" (i.e. the subject pressing a key, thinking he/she has found a set), we claim that eye movements disclose the decision to call "Set!". Especially the last 1 to 4 seconds of a trial (prior to an event, i.e., the player or opponent calling "Set!") seem to contain patterns indicating that a subject is going to call "Set!". A logical explanation would be that since most players (in our subject group) take a few seconds to check if the presumed set is indeed a set, machine learning models learn to recognize this "validation step" in order to predict a positive outcome for "press" (i.e., calling "Set!").

Concerning subquestion (*b*), finding a correct set, we claim that eye movement data does not show the difference between correct and incorrect set identification. On the other hand, *experience scores* can be used by machine learning techniques to predict the finding of a set, but only if the player has already called "Set!". This makes sense, as experienced players (with a high experience score) usually make less mistakes than novice players, leading to a higher correct/call-rate. For example, indicating 4 correct sets within 5 times of calling "Set!" gives a correct/call rate of 4/5). Based on the subject's experience, machine learning models probably learn to estimate the correct/call rate and base their predictions on that. In practice, this means predicting "correct" when an experienced player calls "Set!" and "incorrect" when a novice player calls "Set!".

This research project gave rise to some new questions as well. First, do machine learning models have a tendency to learn the "gaze fingerprint" of certain subjects and base their predictions on those subjects' personal characteristics? Second, which low-level features in eye gaze data disclose information that is useful in predicting the decision of calling "Set!"? And third, what would happen to prediction accuracy if one would add (1) the absolute coordinates of the peaks of focal attention and (2) the properties and positions of the cards in the layout? In future research efforts, these questions might be worth investigating.

References

[1] Adler, S.A. and Orprecio, J. (2006), The eyes have it: Visual pop-out in infants and adults, Developmental Science 9(2), 189–206, 2006.

[2] Atkeson, C., Moore, A. and Schaal, S. (1997), Locally Weighted Learning, AI Review 11, 11–73, 1997.

[3] J.M.B. Craig. (2004), Experiments with SET, Two studies on SET game play, The University of Virginia, USA. http://people.virginia.edu/˜jmc4as/docs/JMBC-710-Research.doc

[4] Vries, T. de (2008), Investigating Visual Attention and Reasoning through Cognitive Modeling, Research Thesis, Universiteit Leiden.

[5] Ditchburn, R.W. and Ginsborg, B.L. (1952), Involuntary Eye Movements During Fixation, page 6.

[6] Erren, T.C., Cullen, P., Erren, M. and Bourne, P.E. (2007), Ten Simple Rules for Doing Your Best Research, According to Hamming, Computational Biology 3(10), e213.

[7] Hahn, B., Ross, T.J. and Stein, E.A. (2006), Neuroanatomical dissociation between bottom-up and top-down processes of visuospatial selective attention, Neuroimage 32(2), 842-853.

[8] Munoz, D.P. *et al*. (2003), Altered Control of Visual Fixation and Saccadic Eye Movements in Attention-Deficit Hyperactivity Disorder, Neurophysiology 90, 503–514.

[9] Taatgen N.A., van Oploo M., Braaksma J. and Niemantsverdriet J. (2003), How to Construct a Believable Opponent using Cognitive Model- ing in the Game of SET, Proceedings of the fifth international conference on cognitive modeling, 201–206.

[10] Wikipedia, *Bayesian network*. Retrieved on May 28, 2009. http://en.wikipedia.org/wiki/Bayesian_network

[11] Wikipedia, *Bias*. Retrieved on May 18, 2009. http://en.wikipedia.org/wiki/Bias

[12] Wikipedia, *Data binning*. Retrieved on May 14, 2009. http://en.wikipedia.org/wiki/Data_binning

[13] Wikipedia, *Decision tree*. Retrieved on June 15, 2009. http://en.wikipedia.org/wiki/Decision_tree

[14] Wikipedia, *Eye movement (sensory)*. Retrieved on May 14, 2009. http://en.wikipedia.org/wiki/Eye_movement_(sensory)

[15] Wikipedia, *Amblyopia*. Retrieved on May 14, 2009. http://en.wikipedia.org/wiki/Amblyopia

[16] Wikipedia, *Meta learning (computer science)*. Retrieved on May 28, 2009.
`http://en.wikipedia.org/wiki/Meta_learning_(computer_science)`

[17] Wikipedia, *Multilayer perceptron*. Retrieved on June 15, 2009.
`http://en.wikipedia.org/wiki/Multilayer_perceptron`

[18] Wikipedia, *Naive Bayes classifier*. Retrieved on May 28, 2009.
`http://en.wikipedia.org/wiki/Naive_Bayes_classifier`

[19] Wikipedia, *Overfitting*. Retrieved on May 28, 2009.
`http://en.wikipedia.org/wiki/Overfitting`

[20] Wikipedia, *Salience (neuroscience)*. Retrieved on May 28, 2009.
`http://en.wikipedia.org/wiki/Salience_(neuroscience)`

[21] Wikipedia, *Set (card game)*. Retrieved on May 6, 2009.
`http://en.wikipedia.org/wiki/Set_(game)`

[22] Wikipedia, *Sequential Minimal Optimization*. Retrieved on June 15, 2009.
`http://en.wikipedia.org/wiki/Sequential_Minimal_Optimization`

[23] Witten, I.H. and Frank, E. (2005), Data Mining: Practical Machine Learning Tools and Techniques (Second Edition).