Object class definition in image using eye-tracking technology and machine learning methods

Master Thesis

Submitted to the
Department of Computer Science and Languages at
Anhalt University of Applied Sciences
in fulfillment of the requirements for the degree of
Master of Science

Student: I. Spirin
(Matr. Nr.: 4065059)
supervisor: Dr. A. Carôt

June 2017

Annotation

In the master's thesis considered the problem of the need to predict class in image for target object detection with high accuracy by means of mathematical model is considered. To solve this problem, data are collected using eye-tracking technology. A technique is developed for detecting an object class in image, which is checked on the classification model of objects.

Contents

Introduction

Recently, researchers in cognitive science have a great interest in computational models that predict the human eye's movement behavior. However, some authors began to use the data obtained from eye-tracker device for computer vision problems. In such studies claimed that using eye-tracking data in image segmentation, the detection performance is increased and the computation time is reduced in comparison using conventional object detection algorithms.

The work urgency is due to need to predict the object class in image, that to find target object with high accuracy through a mathematical model. Object detecting task in image is the first step in process of solving more complex problems, for example, faces recognition, certain object contour outlining, or technical vision of automated systems. The use of mathematical classification models that predict position of an object in image is actual today and requires additional research to improve the quality of detection in modern television systems, identification systems, robot vision, computer animation and other field.

Object of master's thesis is process of object class defining in image and constructing a bounding box around target object.

Purpose of the master's thesis is to develop a mathematical classification model for detecting objects using eye-tracking technology, which allows to more accurately and with a higher speed constructing the bounding box of the target object in image.

To achieve the purpose in the master's thesis following tasks were solved:

1. Collection and preliminary data analysis various classes using eye-tracking technology;

2. Rationale for use of data collection technology for model training;

3. Determination of relationship between fixations and spatial position of object, allowing annotating the object and constructing a bounding box around each object class in image;

4. Development of classification mathematical model;

5. Training the model and results correction.

In the course of master's thesis, were used: eye-tracking technology is a technique that tracks and fixes eye movement; identification of features was carried out through object segmentation and refinement of segmentation; was used soft segmentation technology for to extract a foreground object from an random image; work with super pixels was performed using the turbo pixel method; training and object class prediction was carried out through using the linear SVM method and the weakly supervised localization.

1 DESCRIPTION OF THE OBJECT AND OF RESEARCH METHOD

1.1 Description of the research object

This study focuses on improving computer vision algorithms using eye-tracking technology and visual significance. Recent advances in eye-tracker technology have resulted in a large data set containing images with fixations on object. Since when collecting data, the respondent's task was a visual search for the object and the class definition, then image data has valuable information about the location of the object in image.

Object of this research is process of object class defining in image and constructing a bounding box around target object.

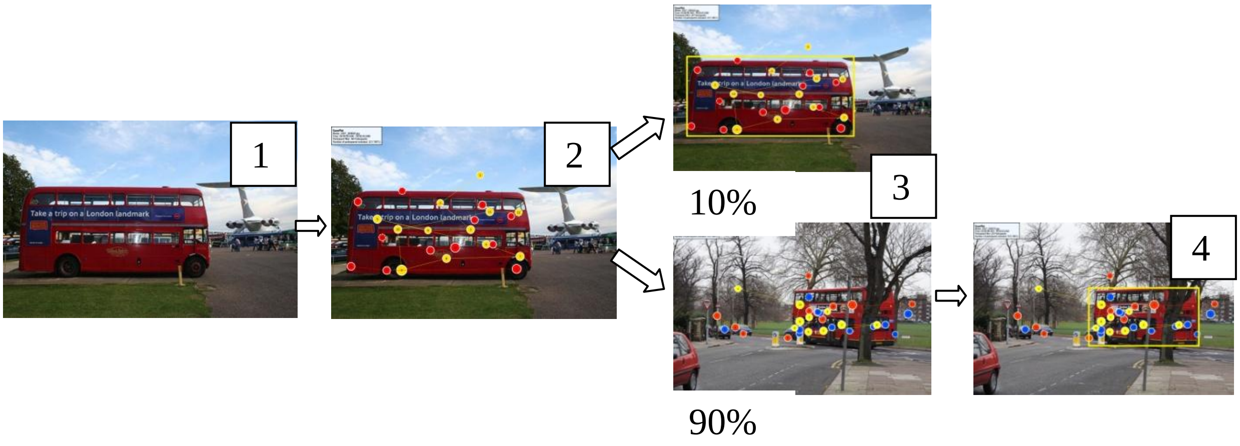


Figure 1 – The process of class defining and constructing the object's bounding box; 1– input images, 2 – marking of fixations eye-tracking on object, 3 – division of images into two blocks (10% – testing images with fixations and manually drawn bounding box, 90% – learning images with fixations), 4 – detection on learning data.

The object class detector is a predictor that annotates object and constructs a bounding box around each object class in image. In order to train object class detector required usually a large number of images, in which the bounding box are built manually. This is a rather time consuming and not effective process. In article [3] described that in order to draw a bounding box, on average, takes 26 seconds. Also requires detailed instructions on annotation, training based on these instructions and verification.

Weakly supervised methods are trained on block of images labeled only as containing a certain objects class, without annotating the location [29, 34]. These methods try to find an object by searching for appearance patterns that are repeated in all testing images. However, learning the detector without annotating the location is very difficult, and performance is much lower than learning methods with the teacher [29, 34, 35]. Also, other researchers use the face and text as a weak annotation [36]. We propose a method that additional foregoing and uses eye fixations on object instead of the usual observation. During eye-tracking research, we get important information about the position and size of object in image. Unlike manual annotation of bounding box, the eye-tracking task does not require annotation instructions and can serve as quick constructions of bounding framework in the future.

1.2 Description of eye-tracking technology

1.2.1 Definition of eye-tracking technology

Eye-tracking is a technology that allows tracking and capturing the user's eyes movement. The device for data recording is called eye-tracker, it consists of several built-in cameras and infrared lamps. Rays of infrared lamps are aimed at human eye and form a glare on the surface of cornea. The cameras focus on them, which fix the movement of view in screen (Fig. 2). Then the device calculates angle of view and records information received in computer [10]. This technology used in studies of visual system, psychology, cognitive linguistics, as well as recently for data collection. Several methods are used to track eyes. The most popular is the time-lapse video eye analysis, also used contact techniques, such as electrooculography [11].

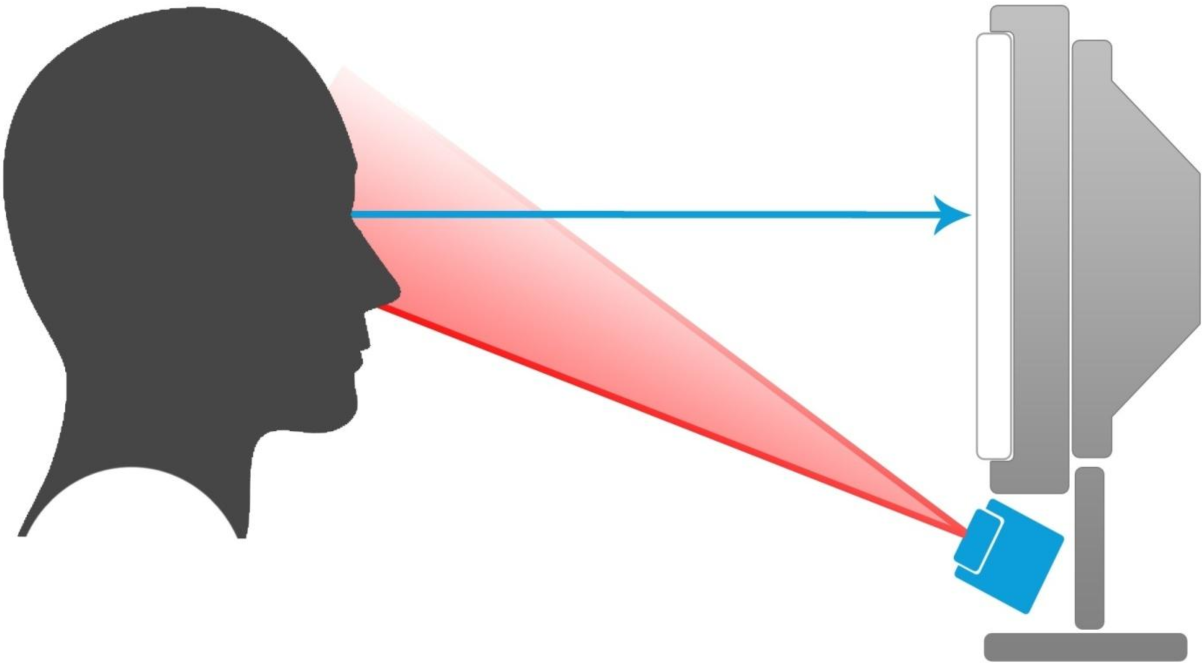


Figure 1 – Process of eye-tracking device

1.2.2 History of appearance of eye-tracking

In the XIX century, studies of eye movement were carried out only by observation method.

In 1879 in Paris, Louis Emil Javal revealed that in process of reading printed text, the eyes do not move monotonously, as they thought before. Instead, they produce short stops, which Javal gave the name of fixation, and sharp movements - saccades. This observation led to appearance of important questions about the essence of the reading process, which had solutions already in twentieth century: What words attract the most attention of a person? What is the duration of such fixations?

The first inventor of tracking device was Edmund Hugh. The device was like a contact lens with a hole for the pupil. The mechanism was connected with an aluminum pointer, which moved synchronously with eye pupil. Hugh used quantized regressions.

First non-invasive eye observer was developed by Guy Thomas Buswell in Chicago. Buswell used reflections of light rays from the eyeball on photosensitive film. In this way, he made research into the processes of reading and studying static images.

In the 1950s in Moscow, Russian scientist Alfred Yarbus conducted important research in field of eye tracking and his 1967 monograph was highly appreciated by the world scientific community. He showed that the formal task assigned to the respondent has a big impact on pupil tracking experiment result.

Yarbus also talked about relationship between eye fixation and the respondent's motivation. A number of experiments showed that experiment result is dependent not only on visual stimulus, but also on task posed, as well as on information that respondent is going to receive from the visual stimulus.

Records of experiments on eye movement showed that only a small part of image elements attract the attention of respondent and his eyes are fixed on these objects. The eye movement process reflects thinking a person process. The view, with some delay, follows point at which the respondent's attention is directed.

Thus, it is possible to identify which elements of image cause more respondent's attention, in what order and how often.

Often attention of the respondent was attracted by objects that cannot give important information, but in his personal opinion they can do it. Eye respondent is fixed on those objects that are unusual in this situation.

Moving from one fixation point to another, the human pupil often returns to some elements of image that he has already observed, that is, additional time is needed to view most important objects instead of viewing less important ones.

In the 1970s, eye tracking research accelerated dramatically, especially in field of reading theory. A qualitative survey of such studies was carried out by Rainer.

In 1980, Just and Carpenter expressed a hypothesis about relationship between the visual system and human consciousness. If this hypothesis is correct, then when a respondent looks at a word or an object, then he thinks about it, and this process is comparable in duration with recorded fixation duration. This hypothesis is often referred to by modern researchers in eye-tracking field.

In the 1980s, this hypothesis developed in the light of hidden attention problem. The hidden attention issue is deciphered in such a way that respondents do not always pay attention to what actually causes their interest. Hidden attention is found in glance movement recording, during which fixations often pass by elements that really attracted attention, and only occasionally show short-term fixations. From this it follows that not in all cases there is an unambiguous relationship between the results of eye-tracking experiment and cognitive process.

Based on Hoffmann's work, the point to which the respondent's attention is directed is always slightly (100-250 ms) ahead of eye movement. However, it is imperative that when attention point moves to a new position, the eyes will try to follow it.

Until now, it is impossible to establish cognitive processes work directly from the eye tracking experiments results. For example, fixing at a face or a picture cannot show that the person or picture likes or dislikes. Therefore, eye-tracking technology requires additional studies confirming dependence of test results and eye fixation [11].

1.2.3 Using eye-tracking technology in object detection

At present, scientific and technological development contributes to formation of new computer vision technical systems, as one of essential directions of human-machine interaction. One of main these systems tasks is object recognition task. With successful solutions to detection tasks, technical production systems will develop that can intelligently recognize the external environment and perform some actions in it. A number of authors began to use data obtained from the eye-tracker device for computer vision problems [1, 2, 5]. Such articles described that using eye-tracking data in image segmentation increases detection performance and reduces computation time compared to using conventional object detection algorithms. There are also articles in which data on eyes movement are used to recognize text or persons. In this research, we'll look at how can use eye movement data to learning a model that defines a particular class object in image.

Use of fixation during tracking has been the subject of a large number of experiments and studies [4]. The results of such experiments show that participants often fix their views on the object. Therefore, the fixation data can be used in models that will automatically locate objects in image. However, the data give only an approximate indication of a certain object, people tend to look at object center, or on face [2].

1.3 Description of machine learning methods

1.3.1 Concept and tasks of machine learning

Machine Learning is a vast field of artificial intelligence that studies the technology of constructing algorithms learning capable. There are two types of training. Pre-school education, or inductive instruction, is based on identification of joint patterns on private empirical data. Deductive learning consists in formalizing the knowledge of experts and their transfer to the computer as a knowledge base. Deductive learning is ranked in field of expert systems, so terms machine learning and learning by precedent are considered synonymous.

Machine learning is at the intersection of mathematical statistics, optimization methods and classical mathematical disciplines, but it also has its own specificity associated with problems of calculation and retraining. Some methods of inductive learning were created as an alternative to modern statistical approaches. Other methods are closely associated with information mining and data mining (Data Mining).

Many abstract sections of machine learning are combined into a separate discipline, the theory of computational learning (Computational Learning Theory, COLT).

Machine learning is not only mathematical but also practical, engineering discipline. The study of theory, as a rule, does not immediately lead to the development of methods and algorithms used in practice. In order to make the algorithms work successfully, we have to invent additional heuristics that make up for the inconsistency of the assumptions made in conditions of specific tasks theory. Virtually no study in computer training has taken place without experiments on certain specific data confirming the practical working capacity of method.

The basic standard types of tasks [39]:

1. Training with teacher (supervised learning) – the most common case. Each precedent is a pair of "object, answer". It is required to find the functional dependence objects descriptions responses and to construct an algorithm that accepts at the input the description of the object and outputs the answer. The quality functional is usually defined as the average error of answers given by the algorithm for all sampling objects.

- Task of classification
- Regression tasks
- Ranking task
- Task of forecasting

2. Training without a teacher (unsupervised learning). In this case, no answers are given, and you need to look for relationships between objects.

- Clustering task
- Task of finding associative rules
- Task of filtering emissions
- Task of building a trust area
- Dimension reduction task

3. Partial training (semi-supervised learning) occupies an intermediate position between teaching with a teacher and without a teacher. Each precedent is a pair of "object, answer", but the answers are known only on a part of precedents. An example of an applied problem is automatic classification of a texts large number, provided that some of them have already been assigned to certain categories.

4. Transductive learning. The final training sample of precedents is given. It is required, according to these particular data, to make predictions with respect to other private data - test sample. Unlike standard setting, there is no need to reveal a general pattern, since it is known that there will be no new test precedents. On other hand, it becomes possible to improve the quality of predictions by analyzing the entire test sample as a whole, for example, by clustering it. In many applications, transductive learning is practically the same as partial learning.

5. Training with reinforcement learning. The role of objects is played by pairs "the situation, the decision taken", the answers are the values of the quality functional characterizing the correctness of the decisions taken (environment reaction). As in forecasting problems, time factor plays an important role here. Examples of applied problems: formation of investment strategies, automatic control of technological processes, self-training of robots, etc.

6. Dynamic learning (online learning) can be both teaching with a teacher, and without a teacher. Specificity is that precedents flow. It is required to immediately decide on each precedent and at the same time complete the model of

dependence, taking into account new precedents. As in forecasting problems, the time factor plays an important role here.

7. Active learning is different in that trainee has opportunity to independently designate following precedent, which will become known.

8. Meta-learning or learning-to-learn differs in that precedents are previously solved learning tasks. It is required to determine which of heuristics used in them work more efficiently. The ultimate goal is to ensure constant automatic improvement of learning algorithm over time.

9. Multi-task learning. A set of interrelated or similar learning tasks is solved simultaneously, using various learning algorithms that have a similar internal representation. Information about similarity of tasks between each other makes it possible to more effectively improve learning algorithm and improve solution quality of the main task.

10. Inductive transfer. The experience of solving individual particular problems of instruction by precedents is transferred to solution of subsequent private learning tasks. For the formalization and preservation of this experience, applied knowledge representation relational or hierarchical structures

1.3.2 Techniques used to learn object class detector

In this research used the linear SVM method [40] to learning super pixel classifier. Support vector machine is a set of identical learning algorithms with a teacher, used in this dissertation to solve the classification problem. Belongs to the family of linear classifiers. The main property of the support vector method is a constant decrease in the empirical classification error and an increase in distance between classes, so the method is also known as the classifier method with maximum distance.

The method main idea consists in placing initial vectors in a space of higher dimension and finding the separating hyperplane with the maximum distance in this space. Two parallel hyperplanes are based on both sides of hyperplane that separates the classes. The separating hyperplane is a hyperplane that maximizes distance to two parallel hyperplanes. The algorithm works on the hypothesis that greater the difference or the distance between these parallel hyperplanes, the smaller the average error of the classifier will be (Fig. 3).

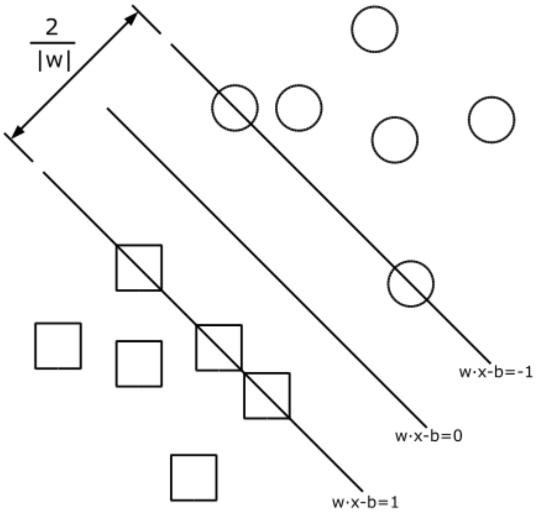


Figure 3 – support vector machines

Also in master's thesis was used Weakly Supervised Localization (WLC) [29,30,31]. This method shows the percentage of correctly localized target class objects in accordance with the Pascal criterion (Fig. 4). There are various works on the study of object detectors from images without location annotation. These methods usually try to approximately localize object instances when studying a class model [32, 33]. Researchers offer a method with a single window selection for each training image from a large set of options to maximize selected windows appearance similarity.

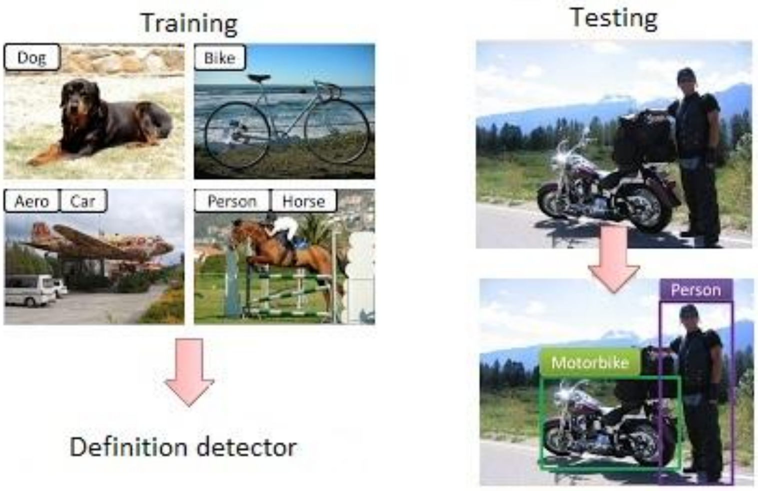


Figure 4 – Weakly supervised method

2 DATA COLLECTION AND PRELIMINARY ANALYSIS

2.1 Collection of eye-tracking data

2.1.1 Materials for study

Let's consider in more detail task of getting fixations in images. Images that contain objects of observation were taken from edition of Pascal VOC 2012 [9]. From this set of images selected part for learning the model, containing 10 different classes and divided in pairs: cat / dog, bus / train, bicycle / motorcycle, airplane / boat, cow / horse. In order to classify image data, the groups were composed in such a way that there were no images containing both classes (for example, the image contains either a cat or a dog). As a result, 5 groups were obtained, containing 1564 images for model training. Since images had a size not corresponding to eye-tracker screen resolution, images were pre-processed. All images were reduced to screen size of 1280x1024. Thus, it was possible to avoid results incorrectness. If the image were small, the observer would always look at center of this object, and eye-tracker would not capture data correctly. When the image size was changed, the objects shape did not change, so the result should be correct.

2.1.2. Data collection procedure

In the articles on object recognition [6, 7, 8] it is stated that free image viewing solves problem with a large error in data collection for learning model. This task, on contrary, increases the probability that respondent recognizes target object, and this makes it easier main task and makes results not correct. Such tasks require a large amount of data and tests. For example, if the main task is to find a cat in image, then the participant presses the "yes" button if the participant sees a cat in image and a "no" button if the cat is missing. Such a data set should consist of images half containing the cat, in order to reduce chance of accidental hitting. Such tracking data is used for a large number of images and cannot be correctly used in model training. In our case, used a task with forced selection of an object (for example, if image contains a cat, the respondent presses one button if image contains a dog, then another) (Fig. 5). Using this method, data is collected for two classes of the object, and this will allow us to reduce probability guessing.

Research begins with a standard procedure - screen calibration. Calibration setting the eye-tracker for each respondent in order to reduce device error.

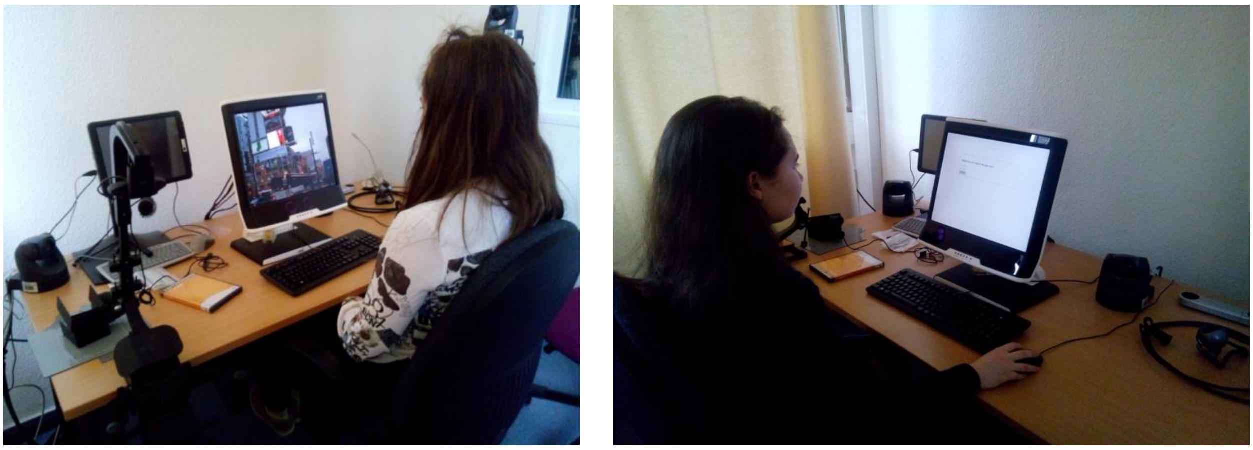


Figure 5 – Respondents undergoing data collection procedure

Participants are invited to view from 3 to 5 blocks on average each of 50 images submitted in a random order. Between blocks, the observer has the opportunity to take a break. The participant's task is to view the image for 3 seconds, and then press one of the two buttons to answer (for example, you saw a cat or dog on the image). This is necessary to determine the class to which objects belong. The procedure of re-calibration should be carried out as necessary. On average, one block is viewed for 5 minutes.

2.1.3 Devices

This experiment is conducted in a special soundproof laboratory. Observers are seated at a distance of 60 cm from the LCD screen, eye movements are recorded with the Eye-tracker Tobii T120 (Fig. 6).

Figure 6 – Devices for data collection Eye-tracker Tobii T120

Pressing buttons takes place due to the module Tobii Studio, which provides high accuracy and generates additional data for training.

2.1.4 Participants

Five observers (3 male and 2 female respondents) participated in the data collection, all students of HS Anhalt. They gave informed consent to participate in the experiment on a voluntary basis. For testing, 10 blocks were prepared, each containing 50 images. Each participant viewed from 3 to 5 blocks with images. Thus, in each of the 10 blocks, several participants were recorded. This will allow us to objectively evaluate the fixations that fall on target object.

2.1.5 Results of data generation

During research, about 12,300 records were collected, on average each participant left 7 fixations in image. Thus, in each image there are from 2 to 4 observations (Fig. 7). As research shows, depending on the task, a large number of fixes are collected on target object, which is confirmed by our guesses about use of eye-tracking for learning object class detector.

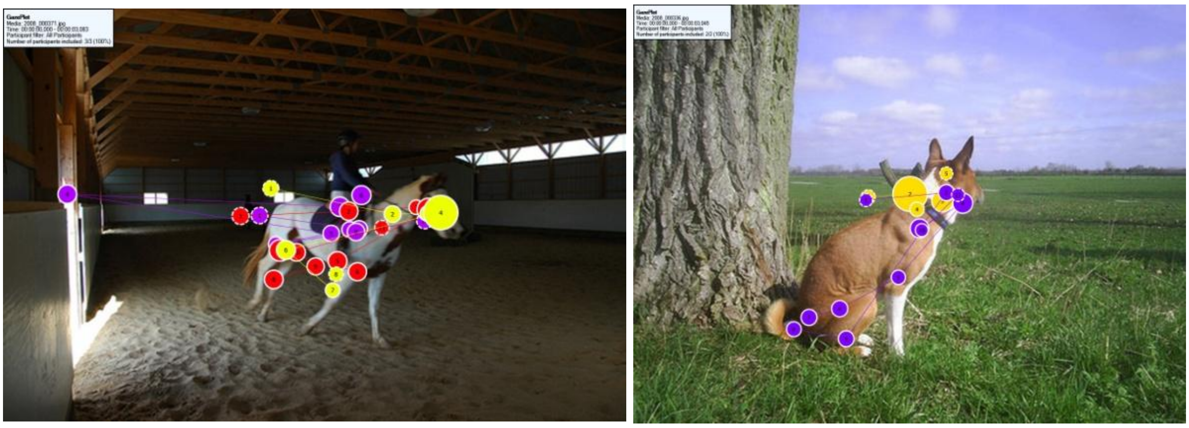


Figure 7 – Fixation of participants on objects in image

The response time of the respondent after a three-second viewing of image averaged 2.35 seconds, therefore, it can be effectively used in constructing task bounding box, compared with the time of constructing bounding box manually (26 seconds) [3].

Comparing bounding box position and the eye fixation, it is revealed that about 85% of fixations are in bounding box. Hence, our guesses confirmed that fixations are useful for localizing an object.

2.2 Preliminary processing of data

Data with fixations got as a result of the experiment will be divided into two blocks: the main block 90% and the small block 10%. The main block will be used as input data in model predicting object position and constructing the bounding box. The bounding rectangle valuation is selected as the problem of segmenting a curved surface. Since the relationship between fixations and the bounding box can be ambiguous, a small block of 10% of the images with fixations is used for this. Preliminarily, a small block is annotated by the bounding boxes manually (Fig.). After that, it is planned to train the model on this data block to get a bounding box for large set. In the future this model can serve for the training of any standard object class detector.

3 DEVELOPMENT OF DETECTING METHOD OBJECT CLASS AND TRAINING THE CLASSIFICATION MODEL

3.1 Machine learning

At this stage it is proposed to consider how the model will be built and learn from the data. This problem will be solved, as the image segmentation based on received fixation diagram. At the entrance, the classification model takes eye-tracking data in the form of human eye fixes and outputs the spatial support of the object, placing each pixel as an object or background. The method consists of two parts: object segmentation and refinement of segmentation [4].

1. At initial stage, predicted object location estimation, designating each super-pixel separately. Superpixel (*C*) is a segment consisting of a pixels set. This parameter determines relationship between human eye fixations and spatial object position. The prognostic parameter is trained on a small set of images, which is annotated both by fixations and manually by bounding frames. Next, we apply trained classifier to a large data set of 90%, resulting in a soft segmentation mask. At the output of this stage, the values for each pixel are formed, which correspond to probability of being on object.

2. At the second stage of segment acquisition, the soft segmentation output M is refined, taking into account the paired dependencies between neighboring super pixels and improving the appearance models.

3.1.1 Segmentation of objects

As it was said earlier, the predictor is trained on a small block of data labeled manually with fixations and bounding box. After training the classifier on a small data set, it was revealed that there is a connection between fixation functions and the fact that the super-pixel is located on the target object or not. In the next step, we apply the trained classifier to a large data set of 90%, resulting in a soft segmentation mask. This process means extracting a foreground object from an arbitrary direct image. Each pixel value in the mask corresponds to estimated probability that pixel is on object.

When segmenting an image, the following features should be noted [23]. In the process of segmentation, each image becomes a superpixel, using the turbopixel method [22]. The turbopixel method is often used for initial segmentation. It forms superpixels about the same size. It uses the approach of level lines for segmentation. The basic idea is the following: for this image, NS starting points (NS - the number of superpixels) are uniformly distributed over image. Of these, contour outgrowth corresponding to the superpixel begins. The contour speed depends on the gradient and proximity to the assumed boundary. Due to this, the received superpixels recover their image growth and divide it into fragments of same size. At the algorithm output, we obtain the matrix $M_{h \times w}$, whose elements are the labels p_i, which indicate the super pixel's belonging to pixel. This algorithm consists in superpixels calculation. It produces segments that, on the one hand, belong to the local boundaries of the image, and on the other hand, it limits the lack of compactness (Fig. 9a). Depending on this method, the computational complexity of the technique is simplified. Unlike the N-cut algorithm [21], this algorithm offers significant acceleration, which ensures compliance with compactness. Figure 8 shows the application of turbopixel method in image.

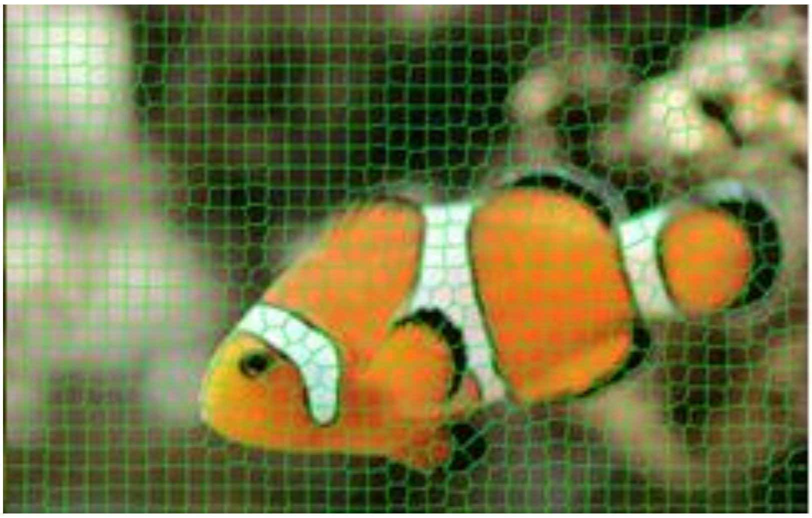


Figure 8 – Turbopixel method in image

Another feature is fixations position. Each fixation is determined by four values: fixation coordinates (x; y), fixation time and rank in chronological order.

Tables of values obtained from device eye-tracker are presented in Appendix 3. In this work, eye-tracking data is used instead of usual view, in order to increase fixes number target object falling into object zone. Thus, fixations position determines object position in image. This allows asserting that super-pixel is on object relative to fixation position (Fig. 9b). Defined features list of superpixel connection and fixation:

• Average distance: the distance between superpixel center and average position of all fixations in image;

• Nearest distance: the distance between superpixel center and fixation position closest to superpixel;

• Average offset: the vertical and horizontal difference between superpixel center and average position of all fixations in image;

• Nearest offset: the vertical and horizontal difference between superpixel center and fixation position closest to the superpixel.

In addition to the each eye-tracking fixation position also provides information about time, such as duration and rank of each fixation. These properties carry valuable information: longer fixation lasts, more significant it is. In addition, in many images, first few fixations do not reach the target object, while later fixations, with a higher probability, will be on object. The list of time features is as follows:

• Time: the fixation duration closest to superpixel

• Rank: the fixation rank closest to superpixel.

Also note an important feature – the fixation appearance. This function supports learning relationship between superpixels on object and fixations. For example, it is possible to find out that superpixels that are at closest fixing distance are on object.

However, in some objects, spatial ratio may vary from image to image, and if learning is done only on coordinates, there will be a large error. In this case, for example, animals can appear in a wide points range view and deformations. This complicates guessing of their entire body, based solely on fixation position.

Next, consider another family of functions, based on superpixels appearance. Main idea is that several superpixels that fall into fixation predetermine background of other superpixels. For example, we notice that dog is black and background is green, then, applying several fixations and this knowledge, we can reveal the dog completely as an object.

Note that this idea operates independently of spatial relationship between shapes and object size in image and fixation location. This method effectively creates a mapping from commit to segmentation, which adapts to contents of target image. To be more precise, an estimate is given for two Gaussian mixture models (GMM), one for object and one for background.

The GMM algorithm is simple enough:

1. Initialize the cluster centers $\mu = (\mu_1 \dots \mu_k)$
2. Calculate hidden variables expectation

$$f\left(z_{ij}\right) = \frac{p(x = x_i|\mu = \mu_j)}{\sum_{j'} p(x = x_i|\mu = \mu_{j'})}$$

3. Recalculate cluster centers $\mu_j = \frac{1}{N}\sum_{i=1}^{N} f\left(z_{ij}\right) x_i$
4. Repeat steps 2-3 until convergence.

i – original centers index, *j* – changed cluster centers index, *p* – function of cluster center and element position, *N* – number of clusters, and x_i – model element.

Each GMM has 5 components, each of which is a Gaussian distribution over RGB color space. The GMM object (obj) is estimated from all the pixels within all superpixels that fall under any fixation, since they are highly likely to be on object (Fig. 9c). The pixels choice located on background is more difficult, because opposite ratio is not realized: the fact that superpixel is not located under any fixation does not determine superpixel position on background or object. Therefore, superpixels are selected according to three criteria that lead to three different GMM background (bg) models, and leave their teaching model in order to decide how best to weigh them:

1. The sample is proportional to distance to middle fixation, as a result of which many samples are far from middle;

2. The sample is proportional to distance to nearest fixation.

3. The sample is inversely proportional to object probability

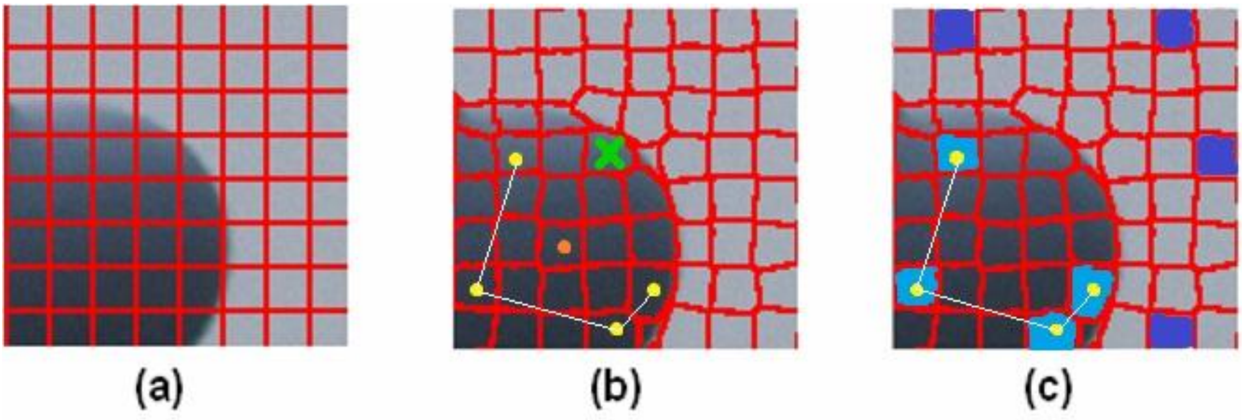


Figure 9 – Fixations position using the turbopixel method

After evaluating models of appearance obj and bg, they are used to estimate each superpixel C in image (1), which results in perpixel object / background probabilities.

$$f(C|obj) = Obj(C), f(C|bg) = Bg(C) \quad (1)$$

The probability combination occurs in a posteriori probability with respect to object according to Bayesian formula (2) with the uniform order:

$$f(obj|C) = \frac{f(C|obj)}{f(C|obj)+f(C|bg)} \quad (2)$$

Three different a posteriori values are calculated using each of three background models in turn, resulting in three appearance functions for each superpixel.

The superpixel classifier training takes place separately for each object class, because connection between fixations and objects can be non-uniform (Fig. 10). The training sample will consist of vector attributes of all superpixels from a small data set. Each superpixel is labeled according to whether it is inside the bounding box or not [23]. After the selection of traits, a linear SVM with high performance is trained on a random set of training data [38]. The regularization parameter is set by checking for 10% of data, and then reconfiguring SVM to 90% of data. To obtain a smooth probabilistic result, scaling is used [37]. The output of the classifier must be calibrated with a posteriori probability for possibility of subsequent processing. Standard SVMs do not provide such probabilities. One of creating probabilities methods is the immediate preparation of a kernel classifier with a logging function and a regularized maximum likelihood indicator. Logit is inverse sigmoidal logistic function or logical transformation used in mathematics, especially in statistics. Thus, training with maximum likelihood indicator will produce non-sparse kernels. Instead, teach SVM, and then train additional sigmoid function parameters to map SVM outputs to probabilities and place the sigmoid on output of SVM on training data 10%.

After, as indicated, classifier training on a small data block (Fig. 10), it is revealed that there is a connection between fixation diagram and fact that superpixel is located on the target object or not. Next, we apply the trained detector to a large data set of 90%, resulting in a soft segmentation mask (Fig. 11).

This process means extracting foreground object from an arbitrary direct image. Each pixel value in the mask corresponds to estimated probability that pixel is on object.

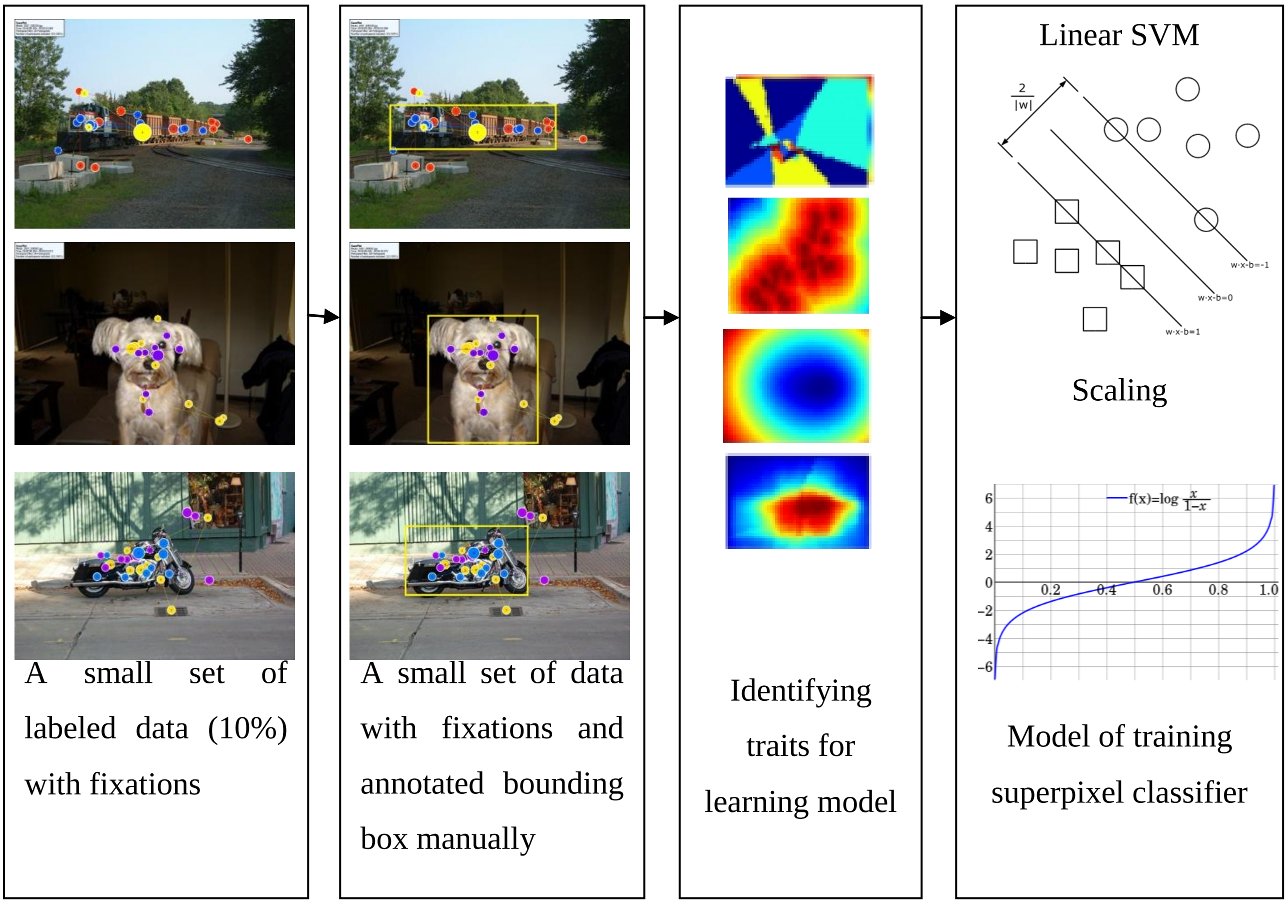

A small set of labeled data (10%) with fixations
A small set of data with fixations and annotated bounding box manually
Identifying traits for learning model
Linear SVM
2/|w|
w·x−b=−1
w·x−b=0
w·x−b=1
Scaling
f(x)=log x/(1−x)
6
4
2
0
−2
−4
−6
0.2
0.4
0.6
0.8
1.0
Model of training superpixel classifier

Figure 10 – Training superpixel classifier on a small block of images (10%)

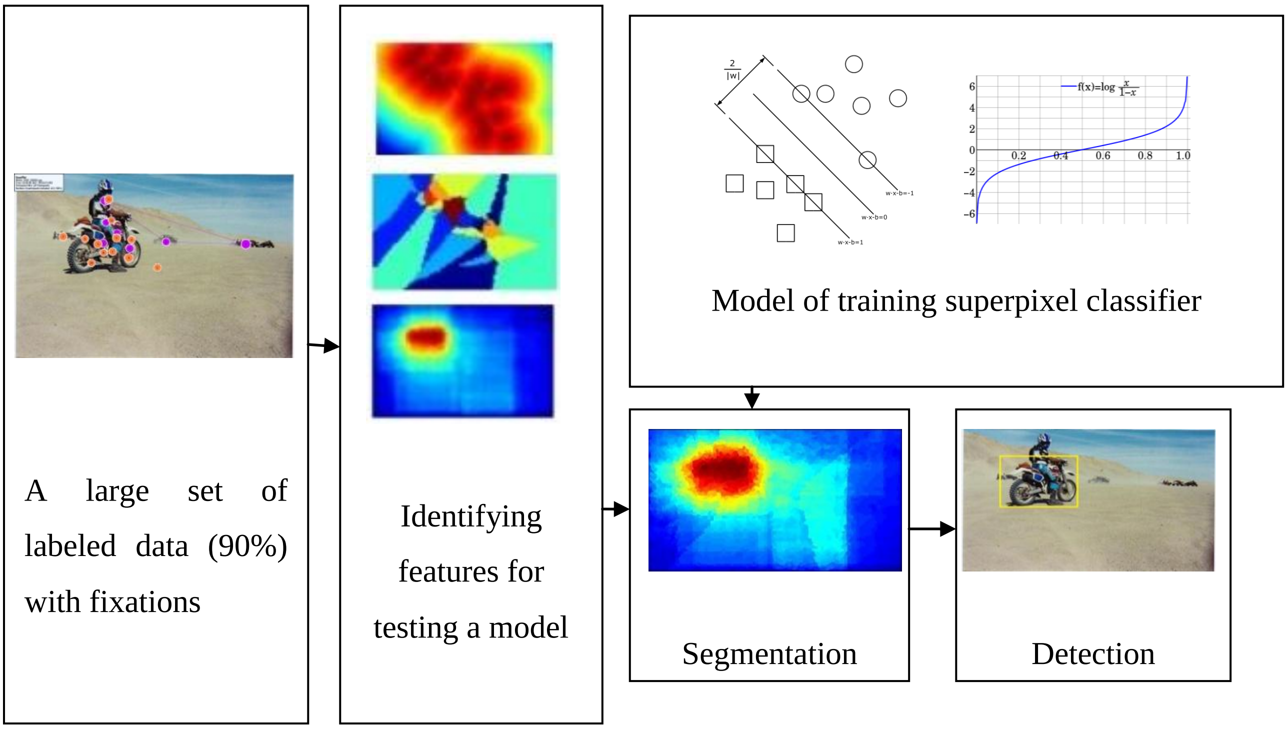

A large set of labeled data (90%) with fixations
Identifying features for testing a model
2/|w|
w·x−b=−1
w·x−b=0
w·x−b=1
f(x)=log x/(1−x)
6
4
2
0
−2
−4
−6
0.2
0.4
0.6
0.8
1.0
Model of training superpixel classifier
Segmentation
Detection

Figure 11 – Testing trained model on a large block of images (90%)

3.1.2 Refinement of segmentation

At the second stage of segment acquisition, soft segmentation output *M* is refined, taking into account paired dependencies between adjacent superpixels and improving appearance models (Fig. 12).

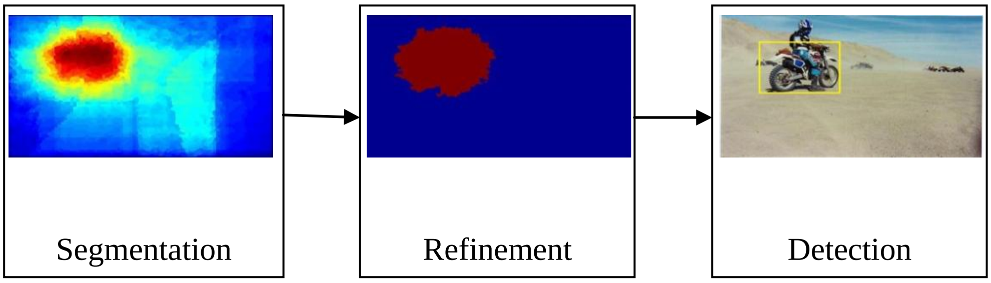


Figure 12 – Refinement of segmentation when an object is detected

Let $I_c \in \{0, 1\}$ – label for superpixels *C*, a *L* – label all I_c in image. We use the binary function pairwise energy *E*, defined over superpixels (3).

$$E(L) = \sum_{\mathrm{c}} M_{\mathrm{c}}\,(I_{\mathrm{c}}) + \sum_{\mathrm{c}} A_{\mathrm{c}}\,(I_{\mathrm{c}}) + \sum_{\mathrm{c},r} V\,(I_{\mathrm{c}}, I_r) \qquad (3)$$

$I_c, I_r -$ labels for superpixels of different classes,

M_c – soft segment mask value in superpixel,

A_c – unary potential value,

V – pairwise potential, encouraging smoothness.

As in [24, 25], the pair potential V encourages smoothness, penalizing neighboring pixels with different labels. The penalty depends on color contrast between pixels, which is smaller in areas with high contrast (the image edges). Summation over (*c*, *r*) is defined on an eight-connected grid of pixels.

Because of the soft segment mask M probabilistic nature, one can use:

$$M_c(l_c) = M(C)^{l_{\mathrm{c}}}(1 - M(C))^{1-l_{\mathrm{c}}} \qquad (4)$$

As a unary potential (with M_c mask value in superpixel *C*). Since Mc evaluates probability that superpixel *C* is on object, this potential stimulates final segmentation to be close to *M* (see [23]). This method binds segmentation to areas of image that can contain target object, allowing second step to refine its exact distinction.

The second unary potential A_c evaluates probability that superpixel to catch label l_s in accordance with appearance models of the object and background, as in classical GrabCut method [25], external model consists of two GMMs, one for object (used for l_c = 1) And one for background (used when l_c = 0).

$$A(I_c|l_c = 1) = A_1(I_c) \qquad (5)$$
$$A(I_c|l_c = 0) = A_0(I_c)$$

Each GMM consists of five components, each of which is a complete Gaussian covariance over RGB color space.

In a traditional paper using similar energy models [25, 26], appearance models requires evaluation user to interact with image area containing object (usually manually drawn by the bounding box). Recently [27] proposed to automatically evaluate appearance models from soft segmentation mask obtained by transferring segmentation from images manually annotated in the training set.

After this initial evaluation, as in [25], alternating between searching for optimal segmentation L, taking into account appearance models and updated appearance models taking into account segmentation alternates. The first step is solved globally by the optimal minimization method using Graph-cuts method [5], since the pairwise potentials are sub modular.

The general idea of Graph-cuts method is as follows: image is represented as a weighted graph, with vertices at the image points. The graph edge weight reflects points similarity in a certain sense (distance between points over some metric). Image partitioning is modeled by graph sections.

The second step corresponds to GMM for labeled superpixels. Energy determination over superpixels pixels instead brings great memory savings and reduces the cost of optimization compared to L. As shown in [28], superpixel model accuracy is almost identical to corresponding pixel model. The final result of our method is bounding box that encompasses largest connected component in segmentation.

3.2 Learning the model

Experiment is carried out on data taken from the Pascal VOC 2012. First, an assessment is made of proposed method ability to construct a bounding box from fixations on a small 10% learning block. Secondly, object class detector is trained, based on bounding box, after which trained classifier is used on a large data block. To teach class detector, images using eye-tracking technology are viewed, data obtained were divided into 2 blocks (10% and 90%). The complete technique for detecting an object class is shown in Appendix 2.

Divided each class into two subsets, both fixation and bounding box constructed by hand are used to prepare segmentation model. This subset consists of labeled data 10%. This block was chosen so that each class had more than 25 images. On average, each class contains 45 images. After training on this set, a trained detector is used on a large set of 90%, which are annotated only with fixations. Next, Weakly Supervised Localization (WLC) [29, 30, 31] will be used. This method shows percentage of correctly localized target class objects in accordance with the Pascal criterion.

There are various works on study of object detectors from images without location annotation. These methods usually try to approximately localize object instances when studying a class model [32, 33]. Researchers offer a method with a single window selection for each training image from a large options set to maximize similarity of selected windows appearance.

For evaluation, 5 baselines were taken:

1. Image Center: a window in image center with an area set in middle of object's bounding boxes. This baseline indicates data set complexity

2. All fixations: bounding box around all the fixings

3. Objectivity: window with highest probability of location on object

4. Model of deformable parts DPM: A sliding window is used, while for each candidate:

$$score(p_0, \dots p_n) = \sum_{i=0}^{n} A_i - \sum_{i=1}^{n} d_i \cdot def\,(\Delta x_i, \Delta y_i) + b \cdot def(\Delta x, \Delta y)$$

$$= (\Delta x, \Delta y, \Delta x^2, \Delta y^2$$

d_i – adjustable weights, for example (0, 0, 1, 1). i – number of candidates, $\Delta x, \Delta y$ – deformation by coordinates. Selecting most appropriate representation uses maximum selection.

5. Regression: linear regression from the mean value of fixations to bounding box.

$$y_i = a + bx_i + e_i$$

y_i – dependent observable variable,

a – free equation term,

b – argument coefficient (independent variable),

x_i – independent variable,

e_i – residual effect.

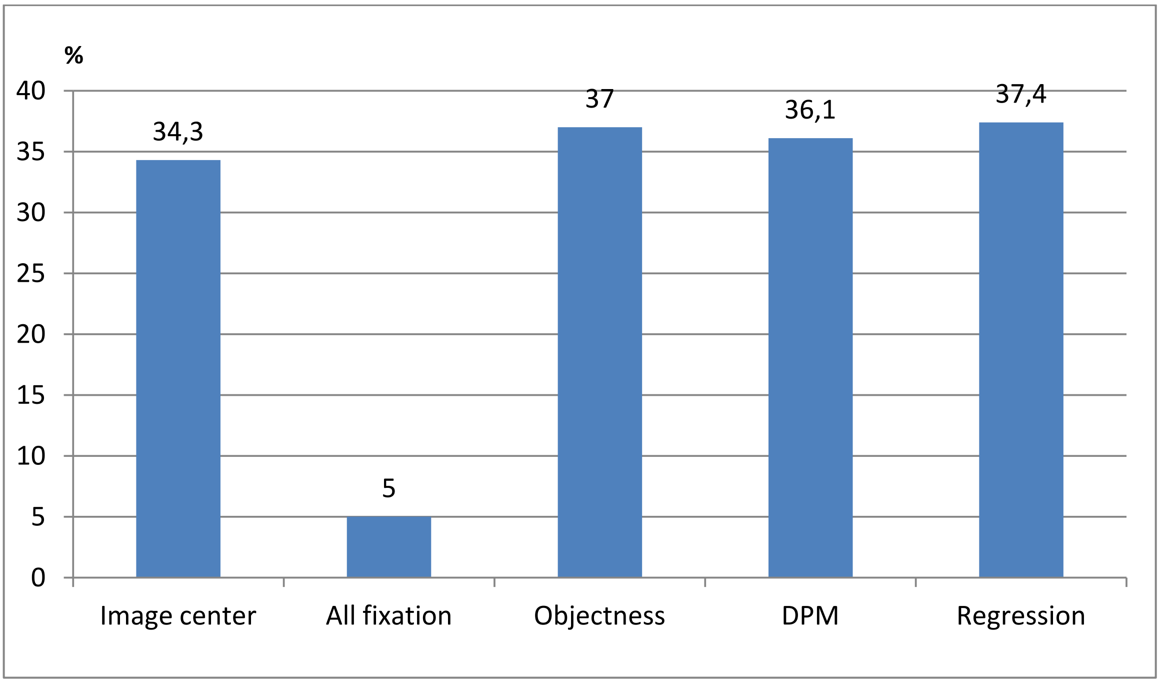


Figure 13 – Baseline Performance

This approach models relationship between fixation diagram and target object. As shown in Figure 13, image center reaches a performance of 34.3%, confirming fact that data block contains images with objects that are not centered. In addition, all fixations do not work completely, demonstrating that task of obtaining bounding box from fixations is far from trivial. Regression shows result better by finding object in images 37.4%. It is important to note need to study relationship between fixations and bounding box. Note that objectivity also works quite well (37.0%). This function can detect some objects when used alone. Since regression and objectivity contain elements of a complete algorithm, they set standards for learning. Finally, the DPM base line reaches only 36.1%, indicating that problem cannot be solved if object class detector is trained on a small fully annotated set.

3.3 Learning model results

Figure 13 shows result achieved by standard baselines. The influence of many different functions and distinction between segmentation (Section 2.2.1) and segmentation refinement (Section 2.2.2) was considered. To quantify segmentation stage performance, a soft segmentation mask is created, and we have a bounding box around most significant segment. Threshold optimization occurs on a training small data set. It is interesting that results became much more accurate.

1. All types of characteristics – features that are expected in section 2.2.1 lead to an improvement in overall model performance.

2. The final model significantly exceeds baseline, including regression and objectivity. This means that model is better able to learn on more complex relationships between fixations and spatial object position.

3. The refinement of segmentation also improves overall performance from 4% to 7%, depending on functions applied at segmentation stage.

Our results show that to detect an object and draw a bounding box it takes 3 seconds, which is significantly less than the 26 seconds required for building bounding box in comparison with the existing method [3].

In our study, only two respondents are recorded, which indicates a research economical version. However, since a fixation was collected from five respondents, a column with a performance of 5 participants is added in Figure 14. As can be seen in Figure 7, number of respondents does not significantly improve the overall model performance, hence, it is proposed to use only fixations of two respondents, which will take less time and will be an effective data collection scenario. The determining objects class results in image are presented in Appendix 1.

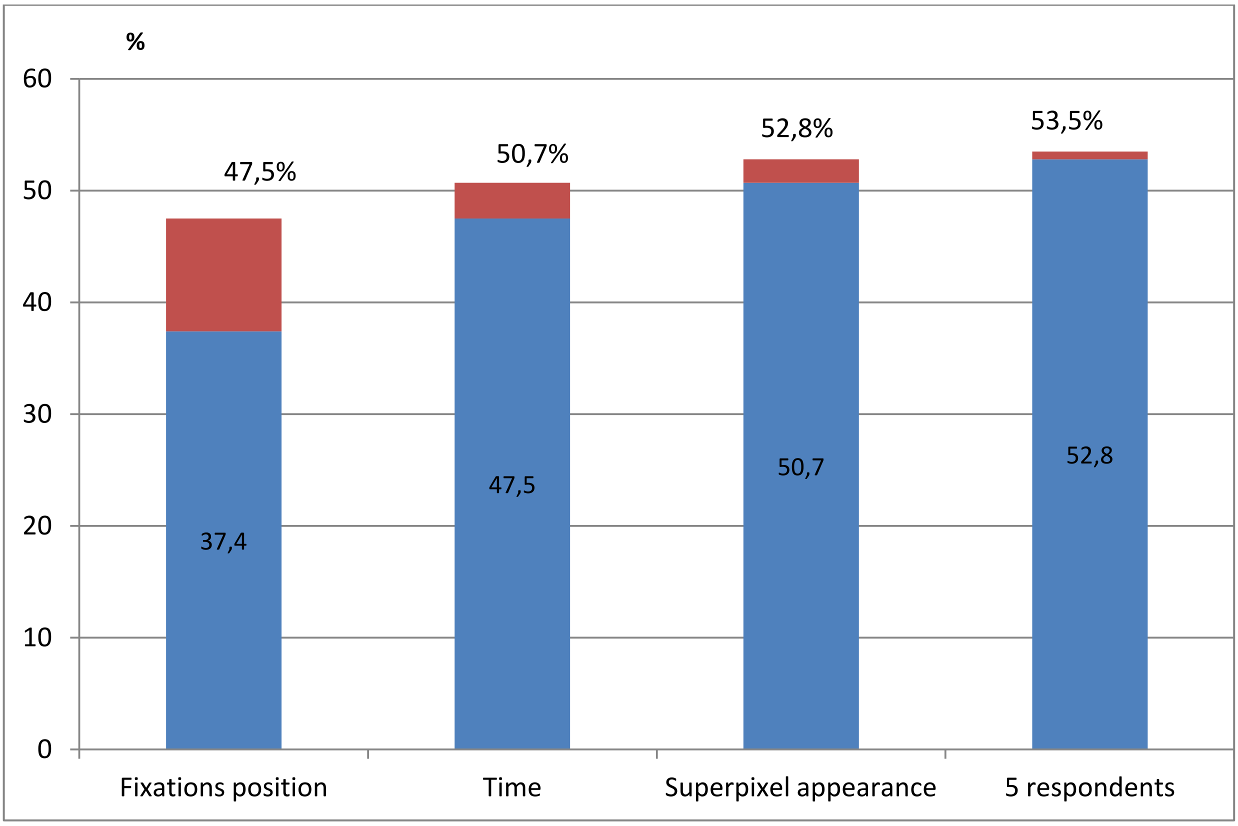


Figure 14 – Model performance using features

After testing, some model tinctures were made. We have taught the model to automatically construct bounding box from fixations for a large set of images from the Pascal VOC 2012 edition [1]. Next, can use results of the predicted bounding box together with 10% images from a small image set with bounding box to learn DPM detector for each class [42]. After training, detectors can be applied to the Pascal VOC 2012 set (10,991 images).

Comparing standard detectors results, annotation by limiting frames of which was made manually by 12.5% and detectors obtained by our method is 16.1%. It can be concluded that this is an encouraging result, considering that scenario of our study (3 sec) allows us to train detectors 8.6 times faster than total annotation time of bounding box (26 sec) in all images. Also it is necessary to take into account all relevant factors, such as: using fixations possibility of two respondents; Time required for setting and calibrating eye-tracker, a break between viewing image blocks; As well as time for drawing bounding box on 10% images in a small data set. To improve the results in future, it is possible to consider additional training methods, as well as to identify new signs of eye-tracking fixation and object spatial location.

CONCLUSION

In research eye-tracking technology was studied, with the help of which it was possible to improve target object annotation results. When collecting data, it was revealed that 85% of fixations were located on target object, this confirms using eye-tracking fixings assumption for learning detector objects class.

Also, relationship features between fixations and object spatial position were investigated. As features, such components as: fixations position, fixations time, superpixels appearance and fixations use of 5 respondents were obtained. It is proven that eye-tracking fixes use can improve standard methods of object detection.

To simplify research in image annotation, improve object position predictability and reduce time it takes to draw the bounding box manually, eye-tracking fixations were used. Taking into account fact that scenario of our research allows us to train detectors in 8.6 times faster than total annotation time of bounding box in all images. And also considering all positive factors described in previous chapter, it is worth concluding that an encouraging result has been obtained that can be used in subsequent studies.

In third part of master thesis, existing detecting methods an object class was investigated, on basis of which a custom modified method was developed, which allows successfully detecting different classes objects in image. Based on obtained method, a mathematical classification model was developed using machine learning methods, such as, Linear Support Vector Machine and Weakly Supervised Object Localization.

In research, predictor was trained on a small block of images (10%), annotated, like eye-tracking fixations, and manually bounding box. Then, using this detector in classification model, a large data set (90%) is tested from the original 1564 images.

This model is able to receive images annotated with eye-tracker fixations and output object spatial support by allocating target object to bounding box with an estimated productivity of 50%.

In the final part, the results were adjusted and retrained in all images (small block 10% and large block 90%). As a result, trained predictor can be used to detect objects on the Pascal VOC 2012 set (10,991 images).

List of sources

1. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascalnetwork.org/challenges/VOC/voc2012/workshop/index.html
2. Harel, J., Koch, C., Perona, P.: Graph-based visual saliency. In: NIPS, 2007
3. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. on PAMI 20(11), 1254–1259, 1998
4. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to predict where humans look. In:IEEE International Conference on Computer Vision (ICCV), 2009
5. Mishra, A., Aloimonos, Y., Fah, C.L.: Active segmentation with fixation. In: ICCV, 2009
6. Ramanathan, S., Katti, H., Sebe, N., Kankanhalli, M., Chua, T.S.: An eye fixation database for saliency detection in images. In: ECCV, 2010
7. Walber, T., Scherp, A., Staab, S.: Can you see it? two novel eye-tracking-based measures for assigning tags to image regions. In: MMM, 2013
8. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: interactive foreground extraction using iterated graph cuts. SIGGRAPH, 2004
9. Levinshtein, A., Stere, A., Kutulakos, K., Fleed, D., Dickinson, S.: Turbopixels: Fast superpixels using geometric flows. In: IEEE Trans. on PAMI (2009)
10. Spirin I.A. Research and application of eye-tracking technology on a person, journal Young Scientist №106, 2016, p. 227-230
11. Eye-tracking [Electronic source]. – URL: https://ru.wikipedia.org/wiki/
12. Dalal, N., Triggs, B.: Histogram of Oriented Gradients for human detection. In: CVPR, 2005

13. Levinshtein, A., Stere, A., Kutulakos, K., Fleed, D., Dickinson, S.: Turbopixels: Fast superpixels using geometric flows. In: IEEE Trans. on PAMI, 2009

14. Simakin I.S. Object position determination in the image along the fragments of the boundary: Yaroslavl, Russia, 2010.– 8 p.

15. Soifer, V.A. Methods of computer image processing / V.A. Soifer, - M.: Fizmatlit, 2001.- 784 p.

16. Branson, S.; Perona, P.; and Belongie, S. 2011. Strong supervision from weak annotation: Interactive training of deformable part models. In ICCV, 1832–1839.

17. Golubev M.N. Development and analysis of algorithms for detecting and classifying objects on machine learning methods basis / thesis – 2012.

18. Search and analysis of optimal color space for building out objects on images given class [Electronic source]. – URL: https://habrahabr.ru/post/229757

19. Spirin I.A., Khoroshev N.I. Methods of processing information using eye-tracking technology / Collection of materials of the XII international school-conference of students, graduate students and young scientists – 2016, p. 172-176

20. Spirin I.A. Development of models for detecting process an object on an image, j. Young Scientist № 138, 2017. – 57 c.

21. Normalized cuts and image segmentation [Electronic source]. – Режим доступа: http://www.cis.upenn.edu/~jshi/papers/pami_ncut.pdf

22. A. Levinshtein, A. Stere, K. N. KutulakosTurboPixels: Fast Superpixels Using Geometric Flows, IEEE Transactions on Pattern Analysis and Machine Intelligence, p.31, Issue: 12, Dec. 2009

23. D. P. Papadopoulos, A. D. F. Clarke, F. Keller and V. Ferrari Training object class detectors from eye tracking data European Conference on Computer Vision (ECCV), Zurich, Switzerland, 2014

24. Kuettel, D., Ferrari, V.: Figure-ground segmentation by transferring window masks. In: CVPR, 2012

25. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: interactive foreground extraction using iterated graph cuts. SIGGRAPH, 2004

26. Wang, J., Cohen, M.: An iterative optimization approach for unified image segmentation and matting. In: ICCV, 2005

27. Kuettel, D., Ferrari, V.: Figure-ground segmentation by transferring window masks. In: CVPR, 2012

28. Guillaumin, M., Kuettel, D., Ferrari, V.: ImageNet Auto-annotation with Segmentation Propagation. Tech. rep., ETH Zurich, 2013

29. Deselaers, T., Alexe, B., Ferrari, V.: Weakly supervised localization and learning with generic knowledge. IJCV, 2012

30. Pandey, M., Lazebnik, S.: Scene recognition and weakly supervised object localization with deformable part-based models. In: ICCV, 2011

31. Prest, A., Leistner, C., Civera, J., Schmid, C., Ferrari, V.: Learning object class detectors from weakly annotated video. In: CVPR, 2012

32. H. Arora, N. Loeff, D. Forsyth, and N. Ahuja. Unsupervised segmentation of objects using efficient learning. In CVPR, 2007.

33. M. Blaschko, A. Vedaldi, and A. Zisserman. Simulatenous object detection and ranking with weak supervision. In NIPS, 2010.

34. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scaleinvariant learning. In: CVPR, 2003

35. Siva, P., Russell, C., Xiang, T., Agapito, L.: Looking beyond the image: Unsupervised learning for object saliency and detection. In: CVPR, 2013

36. Karthikeyan S.V.: Modeling eye tracking data with application to object detection, University of California, 2014

37. Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in large margin classifiers, 1999

38. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms, 2008

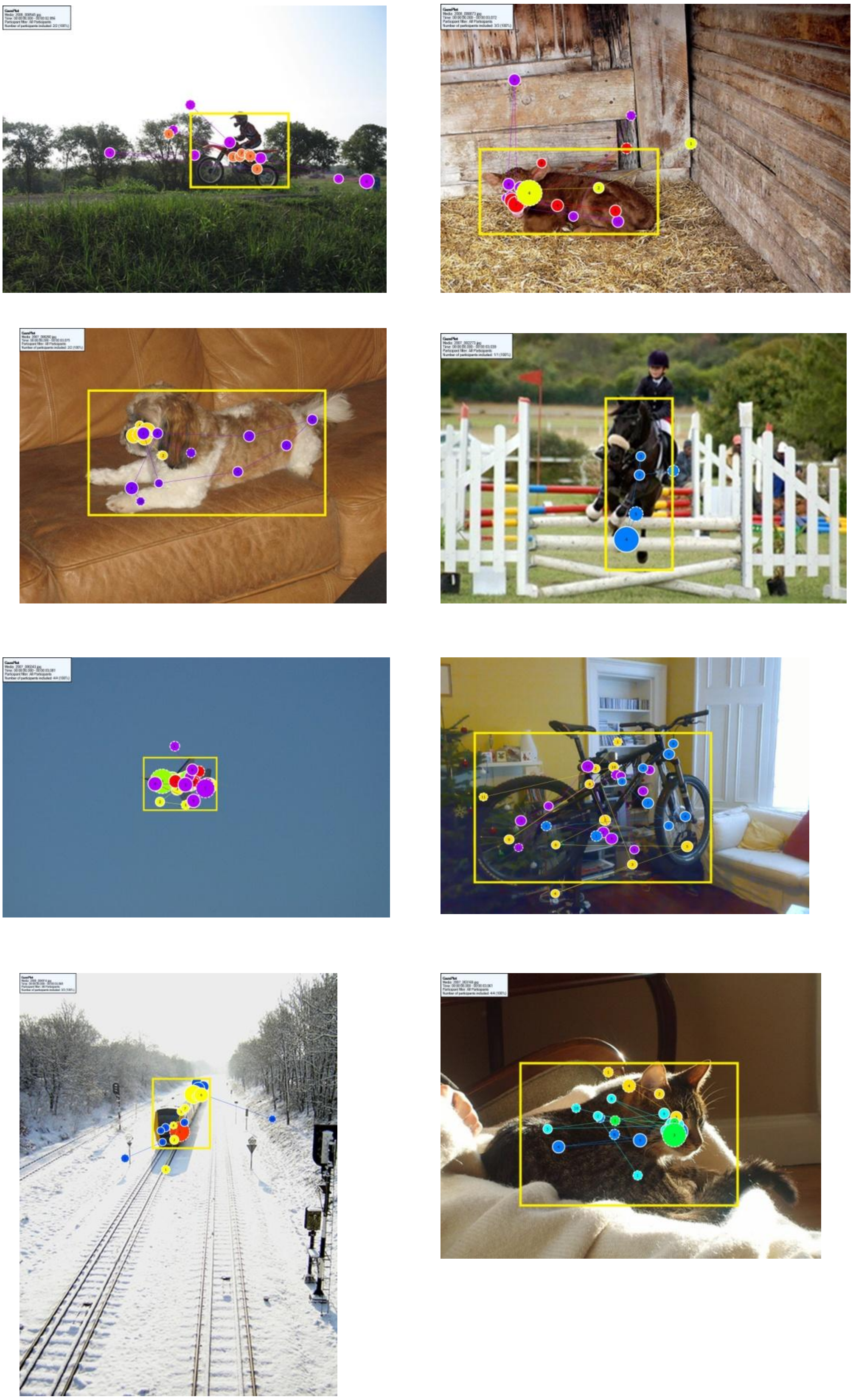
39. Machine lerning [Electronic source]. – URL: http://www.machinelearning.ru/wiki/

40. Support vector machine [Electronic source]. – URL: https://ru.wikipedia.org/wiki

41. Dong Li, Jia-Bin Huang, Yali Li, Shengjin Wang, and Ming-Hsuan Yang Tsinghua University, University of Illinois, Urbana-Champaign, University of California, Merced: Weakly Supervised Object Localization with Progressive Domain Adaptation. In: CVPR, 2016

42. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Trans. on PAMI 32(9), 2010

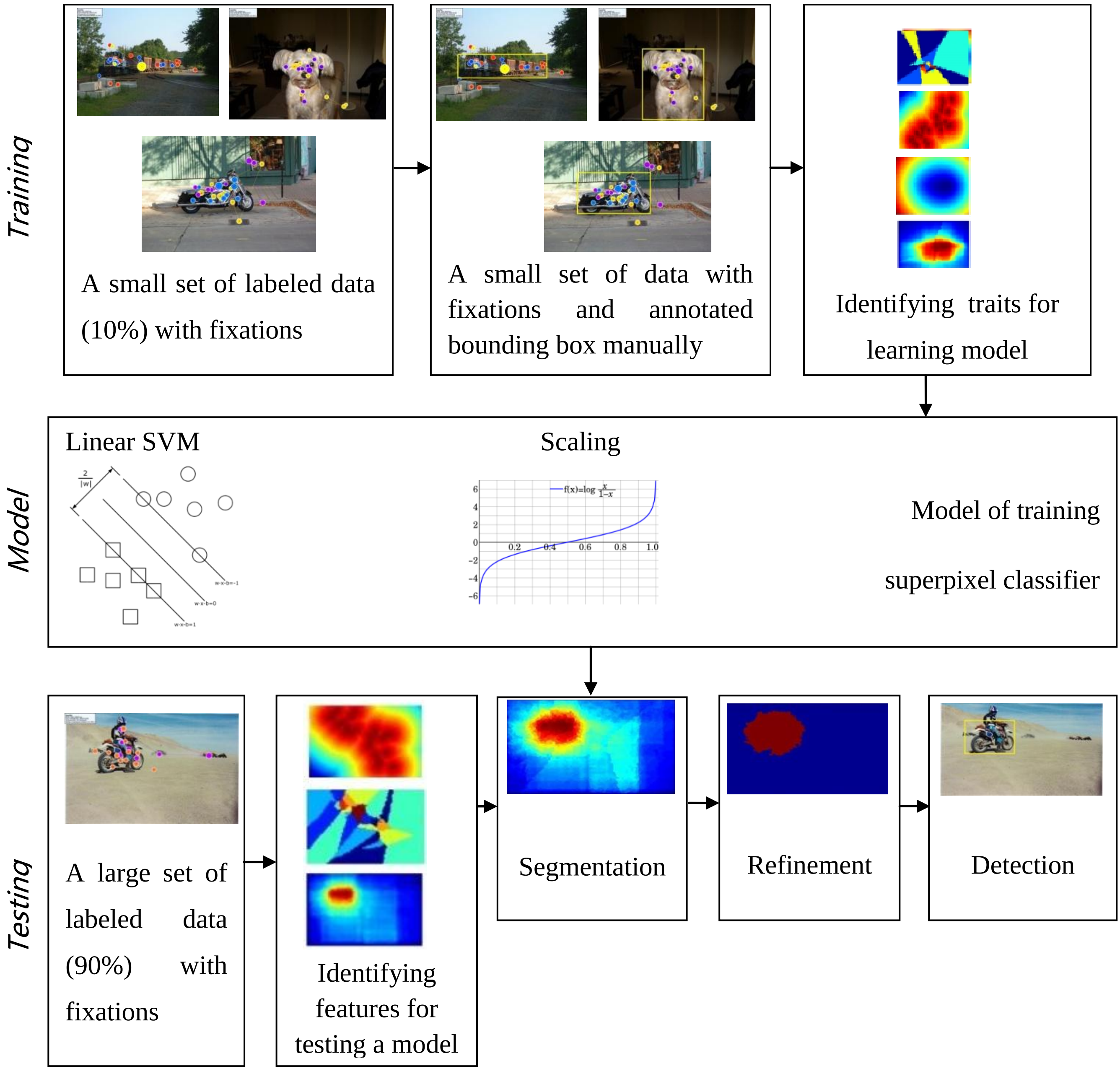
Appendix 1. The classification model results of object detection



The yellow rectangle is bounding box on object

Color circles - this fixes respondents eye

Appendix 2. The detecting method an object class in image



Training

A small set of labeled data (10%) with fixations

A small set of data with fixations and annotated bounding box manually

Identifying traits for learning model

Model

Linear SVM

Scaling

Model of training superpixel classifier

Testing

A large set of labeled data (90%) with fixations

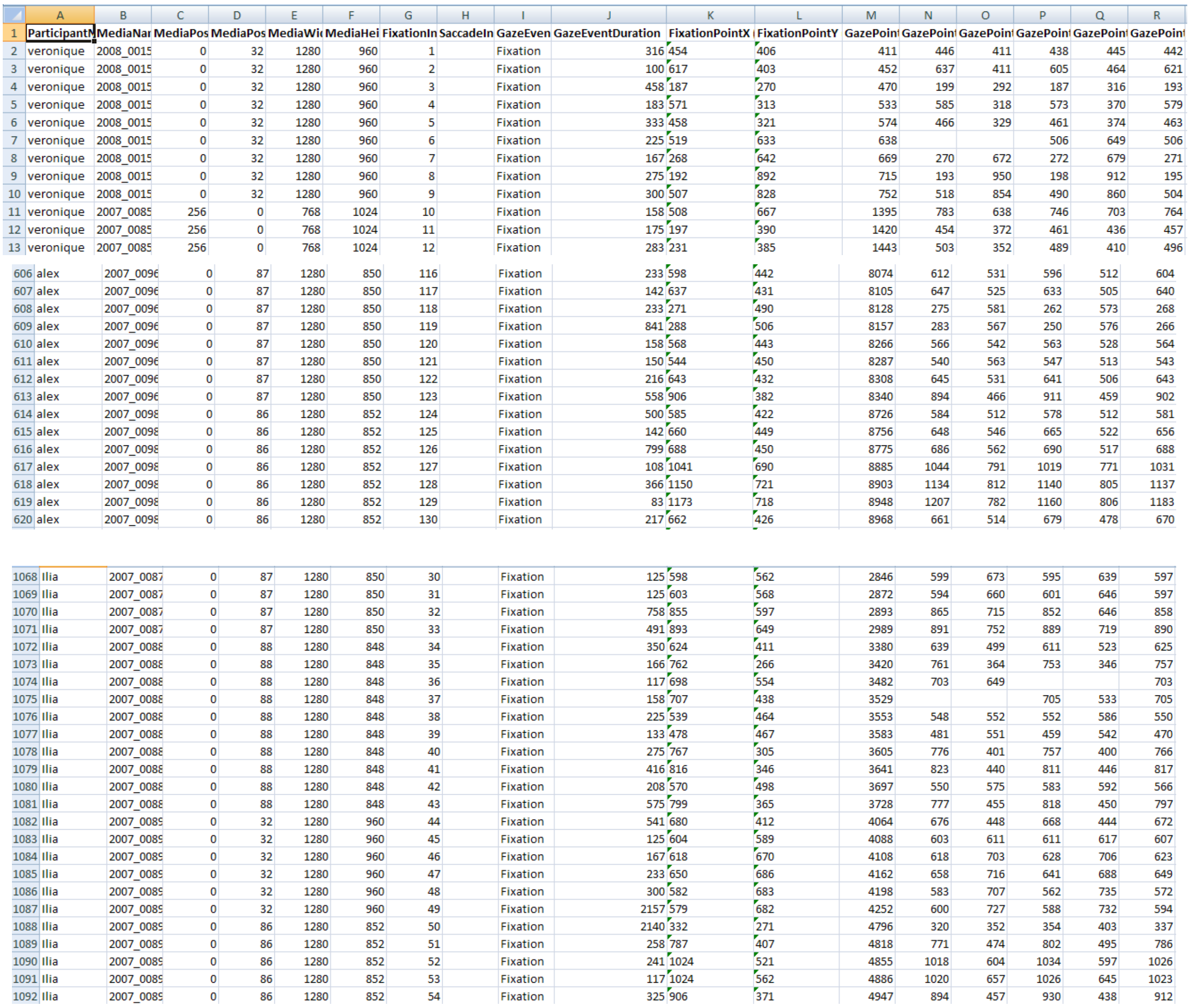
Identifying features for testing a model

Segmentation

Refinement

Detection

Appendix 3. Tabular data taken from eye-tracker



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	ParticipantN	MediaNar	MediaPos	MediaPos	MediaWi	MediaHei	FixationIn	SaccadeIn	GazeEven	GazeEventDuration	FixationPointX (	FixationPointY	GazePoin	GazePoin	GazePoin	GazePoin	GazePoin	GazePoin
2	veronique	2008_0015	0	32	1280	960	1		Fixation	316	454	406	411	446	411	438	445	442
3	veronique	2008_0015	0	32	1280	960	2		Fixation	100	617	403	452	637	411	605	464	621
4	veronique	2008_0015	0	32	1280	960	3		Fixation	458	187	270	470	199	292	187	316	193
5	veronique	2008_0015	0	32	1280	960	4		Fixation	183	571	313	533	585	318	573	370	579
6	veronique	2008_0015	0	32	1280	960	5		Fixation	333	458	321	574	466	329	461	374	463
7	veronique	2008_0015	0	32	1280	960	6		Fixation	225	519	633	638			506	649	506
8	veronique	2008_0015	0	32	1280	960	7		Fixation	167	268	642	669	270	672	272	679	271
9	veronique	2008_0015	0	32	1280	960	8		Fixation	275	192	892	715	193	950	198	912	195
10	veronique	2008_0015	0	32	1280	960	9		Fixation	300	507	828	752	518	854	490	860	504
11	veronique	2007_0085	256	0	768	1024	10		Fixation	158	508	667	1395	783	638	746	703	764
12	veronique	2007_0085	256	0	768	1024	11		Fixation	175	197	390	1420	454	372	461	436	457
13	veronique	2007_0085	256	0	768	1024	12		Fixation	283	231	385	1443	503	352	489	410	496
606	alex	2007_0096	0	87	1280	850	116		Fixation	233	598	442	8074	612	531	596	512	604
607	alex	2007_0096	0	87	1280	850	117		Fixation	142	637	431	8105	647	525	633	505	640
608	alex	2007_0096	0	87	1280	850	118		Fixation	233	271	490	8128	275	581	262	573	268
609	alex	2007_0096	0	87	1280	850	119		Fixation	841	288	506	8157	283	567	250	576	266
610	alex	2007_0096	0	87	1280	850	120		Fixation	158	568	443	8266	566	542	563	528	564
611	alex	2007_0096	0	87	1280	850	121		Fixation	150	544	450	8287	540	563	547	513	543
612	alex	2007_0096	0	87	1280	850	122		Fixation	216	643	432	8308	645	531	641	506	643
613	alex	2007_0096	0	87	1280	850	123		Fixation	558	906	382	8340	894	466	911	459	902
614	alex	2007_0098	0	86	1280	852	124		Fixation	500	585	422	8726	584	512	578	512	581
615	alex	2007_0098	0	86	1280	852	125		Fixation	142	660	449	8756	648	546	665	522	656
616	alex	2007_0098	0	86	1280	852	126		Fixation	799	688	450	8775	686	562	690	517	688
617	alex	2007_0098	0	86	1280	852	127		Fixation	108	1041	690	8885	1044	791	1019	771	1031
618	alex	2007_0098	0	86	1280	852	128		Fixation	366	1150	721	8903	1134	812	1140	805	1137
619	alex	2007_0098	0	86	1280	852	129		Fixation	83	1173	718	8948	1207	782	1160	806	1183
620	alex	2007_0098	0	86	1280	852	130		Fixation	217	662	426	8968	661	514	679	478	670
1068	Ilia	2007_0087	0	87	1280	850	30		Fixation	125	598	562	2846	599	673	595	639	597
1069	Ilia	2007_0087	0	87	1280	850	31		Fixation	125	603	568	2872	594	660	601	646	597
1070	Ilia	2007_0087	0	87	1280	850	32		Fixation	758	855	597	2893	865	715	852	646	858
1071	Ilia	2007_0087	0	87	1280	850	33		Fixation	491	893	649	2989	891	752	889	719	890
1072	Ilia	2007_0088	0	88	1280	848	34		Fixation	350	624	411	3380	639	499	611	523	625
1073	Ilia	2007_0088	0	88	1280	848	35		Fixation	166	762	266	3420	761	364	753	346	757
1074	Ilia	2007_0088	0	88	1280	848	36		Fixation	117	698	554	3482	703	649			703
1075	Ilia	2007_0088	0	88	1280	848	37		Fixation	158	707	438	3529			705	533	705
1076	Ilia	2007_0088	0	88	1280	848	38		Fixation	225	539	464	3553	548	552	552	586	550
1077	Ilia	2007_0088	0	88	1280	848	39		Fixation	133	478	467	3583	481	551	459	542	470
1078	Ilia	2007_0088	0	88	1280	848	40		Fixation	275	767	305	3605	776	401	757	400	766
1079	Ilia	2007_0088	0	88	1280	848	41		Fixation	416	816	346	3641	823	440	811	446	817
1080	Ilia	2007_0088	0	88	1280	848	42		Fixation	208	570	498	3697	550	575	583	592	566
1081	Ilia	2007_0088	0	88	1280	848	43		Fixation	575	799	365	3728	777	455	818	450	797
1082	Ilia	2007_0089	0	32	1280	960	44		Fixation	541	680	412	4064	676	448	668	444	672
1083	Ilia	2007_0089	0	32	1280	960	45		Fixation	125	604	589	4088	603	611	611	617	607
1084	Ilia	2007_0089	0	32	1280	960	46		Fixation	167	618	670	4108	618	703	628	706	623
1085	Ilia	2007_0089	0	32	1280	960	47		Fixation	233	650	686	4162	658	716	641	688	649
1086	Ilia	2007_0089	0	32	1280	960	48		Fixation	300	582	683	4198	583	707	562	735	572
1087	Ilia	2007_0089	0	32	1280	960	49		Fixation	2157	579	682	4252	600	727	588	732	594
1088	Ilia	2007_0089	0	86	1280	852	50		Fixation	2140	332	271	4796	320	352	354	403	337
1089	Ilia	2007_0089	0	86	1280	852	51		Fixation	258	787	407	4818	771	474	802	495	786
1090	Ilia	2007_0089	0	86	1280	852	52		Fixation	241	1024	521	4855	1018	604	1034	597	1026
1091	Ilia	2007_0089	0	86	1280	852	53		Fixation	117	1024	562	4886	1020	657	1026	645	1023
1092	Ilia	2007_0089	0	86	1280	852	54		Fixation	325	906	371	4947	894	457	930	438	912

Hochschule Anhalt
Fachbereich Informatik und Sprachen
Master of Science *Informationsmanagement*

Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass diese Arbeit mit dem Titel

Object class definition in image using eye-tracking technology and machine learning methods

von mir selbständig verfasst wurde und in gleicher oder ähnlicher Fassung noch nicht in einem anderen Studiengang als Prüfungsleistung vorgelegt wurde. Ich habe keine anderen als die angegebenen Hilfsmittel und Quellen, einschließlich der angegebenen oder beschriebenen Software, verwendet.

Ort, Datum: _____20.06.2017_ Unterschrift: _Spirin I.A.__________

This statutory declaration is to be included at the end of the Master Thesis. Please use the German version above. The translation is as follows:

Statutory declaration

I herewith declare that I have completed this thesis independently making use only of the specified aids and sources, including the specified or described software. This Thesis (title) in this form or in any other form has not been submitted to any other examination body.

Date:_________20.06.2017____Signature:____Spirin.I.A.________