Fixation Based Object Recognition in Autism Clinic Setting

Sheng Sun[1], Shuangmei Li[2], Wenbo Liu[3], Xiaobing Zou[4], and Ming Li[5]

[1] School of Electronics and Information Technology, Sun Yat-sen University, China,
[2] PuJiang Institute, Nanjing Tech University, China,
[3] Department of Electrical and Computer Engineering, Carnegie Mellon University,
[4] The Third Affiliated Hospital of Sun Yat-sen University, China,
[5] Data Science Research Center, Duke Kunshan University, China,
ming.li369@duke.edu,
WWW home page: https://scholars.duke.edu/person/MingLi

Abstract. With the increasing popularity of portable eye tracking devices, one can conveniently use them to find fixation points, i.e., the location and region one is attracted by and looking at. However, region of interest alone is not enough to fully support further behavior and psychological analysis since it ignores the abundant information of visual information one perceives. Rather than the raw coordinates, we are interested to know the visual content one is looking at. In this work, we first collect a video dataset using a wearable eye tracker in an autism screening room setting with 14 different commonly used assessment tools. We then propose an improved fixation identification algorithm to select stable and reliable fixation points. The fixation points are used to localize and select object proposals in combination with object proposal generation methods. Moreover, we propose a cropping generation algorithm to determine the optimal bounding boxes of viewing objects based on the input proposals and fixation points. The resulted cropped images form a dataset for the subsequent object recognition task. We adopt the AlexNet based convolutional neural network framework for object recognition. Our evaluation metrics include classification accuracy and intersection-over-union (IoU), and the proposed framework achieves 92.5% and 88.3% recognition accuracy on different testing sessions, respectively.

1 Introduction

Research on human eyes is becoming increasingly attractive in the past few decades [1, 2]. Researchers found that the trajectory movements of our eyes often contain some specific patterns [3, 4] and can be mainly described by fixations and saccades. When someone stares at a point, the gaze may not be strictly fixed, sometimes jitters within a very small region. On the other hand, saccades are quick movements of our gazes when we read texts or view scenes. Both fixations and saccades can vary rapidly along the time. It is therefore almost impossible to observe and accurately measure eye movement by naked eyes.

Eye research relies heavily on eye tracking devices which can generate information including front scene and fixation point, i.e., the point of gaze. There are mainly two types of eye tracking devices. The first type is table-mounted ones that contain an illuminator and a camera mounted above or below the computer screen. The second ones are head-mounted devices that typically contain an illuminator, a front view camera that records the scene one is looking at, and one or two cameras that capture the movements of the eyes. Table-mounted ones are suitable for tasks such as texts reading, advertisement viewing, or visual searching on computer screens, while head-mounted wearable ones are more portable so that the viewer can walk around and perform eye tracking experiments in real world environments. These wearable eye tracking devices provide great opportunities for research on psychological behavior analysis, advanced human-computer interaction, Augmented Reality (AR) applications, etc.

Recently, considerable progress on high-level computer vision, especially object recognition has been made with the advancement of deep representation learning. Traditionally, hand-crafted features such as scale-invariant feature transform (SIFT) [4] and histogram of oriented gradients [5] present the main visual representation methods as they are designed to be invariant to scale, orientation, affine distortion, and illumination changes. Later, convolutional Neural Networks (CNNs), which consider deeply layered nonlinear representations with neurons, pushed the boundaries of object recognition to new levels. Some large scale datasets such as ImageNet [6] are widely used to pretrain a network model from scratch, and then researchers can fine tune the model with domain specific data. There are several popular network structures, from the simple Alexnet [7], to GoogLeNet, VGGNet, and ResNet. CNNs require a large amount of training data and computing resources, so the introduction of large scale image databases and the advance of GPU computing pave the way for the rapid improvement in object recognition. Also, the emerging deep learning frameworks such as Caffe [8], Torch, and TensorFlow make it easier to implement the powerful CNNs. Similarly, performance of the object detection method has also been enhanced significantly due to the usage of CNNs. Histogram of Oriented Gradients (HOG) features and Deformable Part Models (DPM) with Support Vector Machine (SVM) have been used to detect objects, however, only moderate recognition accuracy was achieved. Currently, several novel CNN based models such as faster R-CNN and You Only Look Once (YOLO) [9] dramatically improve the accuracy on the PASCAL VOC and COCO dataset.

As these fields develop rapidly, we come up with an idea of combining them and trying to identify what we are looking at instead of simply where. This is of great importance because when we want to analyze a video clip with fixation point included, we first need to label the frame of interest and objects manually, which is a time-consuming task. If we can utilize the computer vision technology to automatically label the object, we would save plenty of time and money for in-depth analysis of the data. What's more, when we want to know the conspicuity of the objects or the histogram of eye fixations on different objects, the proposed method can generate the results more efficiently and objectively.

In our study, eye tracking and object recognition are standard tasks individually, but when combined together, multiple new challenges emerge. Firstly, since there is few related work, the open databases are scarce. Therefore, we have to collect a database to perform experiments. Secondly, the eye tracking devices use fisheye lenses, which produce strong distortion and negatively affect the recognition accuracy when we use the pre-trained models. Thirdly, we seldom look at the center of an object, so the fixation generated by the eye tracking device is usually at the edge or even a few pixels away, making it difficult to determine whether the viewer is actually watching the object or just looking at another place that is close to the object. Fourthly, the resolution of the captured videos is relatively low, making small and far away objects more difficult to be distinguished. Last but not the least, it is both labor and cost expensive to collect sufficient amount of training data for large-scaled network training.

In this paper, we first introduce a fixation based object recognition dataset. The setting is in an Autism Diagnostic Observation Schedule (ADOS) screening room where the subject looks at those assessment tools. Knowing what the children look at and how their eye movements behave when stimuli appear will benefit the diagnosis of Autism and other psychological experiments. In this scenario, we select 14 commonly used ADOS assessment tools, which are shown in Fig.1. There are two main reasons for this choice. First, these 14 assessment tools are standard and widely used in different ADOS screenings. Second, they are big enough for the eye tracking device to clearly capture.

Fig. 1. Sample images of the selected 14 assessment tools and their labels.

After collecting the data, we preprocess images by identifying fixations. During the testing phase, we first determine the region based on the fixation point

and the bounding boxes generated by the object detection module, then we predict the class label using our object recognition system.

The remainder of the paper is organized as follows. In Section 2, we present the related works. In Section 3, we introduce the data collection procedure and in Section 4, we describe the methods in details. Experimental results are provided in Section 5 followed by the conclusions in Section 6.

2 Related works

With the increasing popularity of eye tracking devices, more and more research on eye-gaze pattern analysis appears. Portable and wearable eye tracking devises are more convenient for real world eye tracking experiments.

German Research Center for Artificial Intelligence (DFKI) has been working on this topic [10–12]. The goal of their research is to develop an AR human computer interaction application, namely, Museum Guide 2.0. It can provide tourists with personal guide in the museum. The tourist wears the eye tracking device while walking around. If he looks at an exhibit item, the application can detect the gaze and present relevant information such as audio descriptions so that the tourist would have a better understating of the exhibition. The original implementation of their method is shown as follows: 1. Creation of exhibits database: take images from different angles with eye tracking device, then extract SIFT features and label them. 2. Object recognition: crop a region with fixed size around the fixation point and extract SIFT features. Compare it with every sample in the database and find the nearest one using Euclidean distance.

Later Shdaifat Mustafa et al. [12] proposed a segmentation-based method to generate dynamic region size instead of the fixed region size method in step 2. They first conduct a series of image pre-processing steps, including Canny edge detection, morphology operations, etc. Then, the proper boundary and bounding box was selected based on the position of the fixation point.

However, the aforementioned methods have certain limitations. First, during testing, the SIFT descriptors of the test image needs to be compared with all the samples in the database, which is not time efficient. Second, SIFT feature is not robust enough to achieve highly accurate recognition performance. Third, the segmentation based bounding box detection method is not robust against fixation deviation and offset. We believe that, using the state-of-the-art YOLO framework with out proposed selection method for bounding box generation and the Alexnet model for object recognition would significantly enhance the overall accuracy and efficiency.

3 Data Collection

3.1 Eye Tracking Device

The eye tracking device used in this study is the UltraFlex headgear designed by Positive Science [13]. The head-mounted eye tracker includes eye/scene cameras,

audio, and infrared illumination. The scene camera faces front and the eye camera capture the right eye. Fig.2 shows the eye tracking device and one sample frame [14]. The IRLED illuminates the eye, then the device and corresponding software estimate the gaze location by center of the pupil and corneal reflection and then project the coordinate onto the video captured by the scene camera. The device operates at 28 frames per second with a resolution of 640 x 480 for the scene view and 320 x 240 for the eye view.

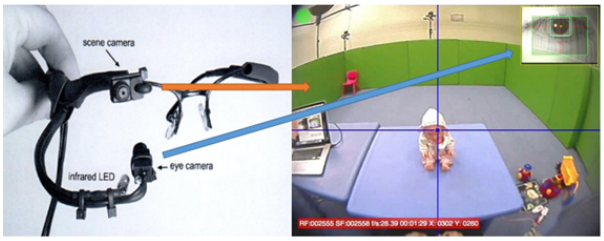

Fig. 2. The three main components of the DB9-KHG-1 Child Headgear eye tracking device and the views of the scene camera and the eye camera with detailed information.

3.2 Training Data Collection

In this work, we proposed a new object recognition dataset with a wearable eye tracking device. This is a close-set datset in which the objects in the testing set is a subset of the ones in the training set.

In the collection of training data, we collected the images of each object from different views. The object was put on the table, and the subject wearing the wearable eye tracking device looks at the object from different angles. If the object has more than one formats, we treat them as different classes. For instance, the baby toy can be converted into two different shapes, sitting and lying. We name them as sitting baby and lying baby separately. Fig.3(a) illustrates these two classes of baby. After collecting the training data, we extract frames from the scene video every 10 frames and use the open source tool named LabelImg to annotate the bounding box of the object of interest. Note that we also create a special class named "Others", which represents the objects that do not belong to the pre-defined 14 classes, including the table, the floor, and the wall in the background, etc. Some examples of class "Others" were shown in Fig.3(b).

3.3 Testing Data Collection

The testing data was collected in two different sessions with objects in short and long distances.

In session 1, the objects are laid on the table the same way as the training data collection. In order to efficiently collect the testing data in a real autism

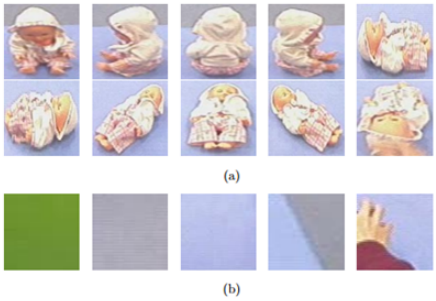

Fig. 3. Sample training images for the class Baby (a) and Others (b).

clinic setting, we group the objects into seven categories, which is shown in Table. 1. During each round, all objects from one category are put on the table. The participant is asked to look at the objects from random positions and angles. Furthermore, the locations of objects were changed several times during the testing data collection to enhance the generalization. Fig. 4 shows the experiment setup for this scenario.

In session 2, the subject sits on the ground with most of the objects surrounding him in a circle as shown in Fig. 5. This setting is different from the one in our training data and session 1. We want to evaluate the robustness of our model against the recording environment changes.

Table 1. The seven categories of the ADOS assessment tools

Category	Assessment tools
1	Baby, Rabbit
2	RedToy, BlueToy, ToyGuy
3	FireEngine, Truck
4	Book, Plate
5	ChestBox, FoodContainer
6	Ball
7	Cup, Jar

Fig. 4. Data collection setup for session 1 testing data.

Fig. 5. Data collection setup for session 2 testing data.

3.4 Fixation Identification

The eye movements include two types: saccade and fixation. Saccades are the type of eye movement used to move rapidly from one point to another, while for fixation, the eye movements are relatively stable in a certain duration. We are more interested in conducting recognition in fixation points because fixation is a natural description of the observed eye movement behaviors.

In order to differentiate the fixations from saccades, we use modified Dispersion-Threshold Identification (I-DT) algorithm similar with the Dispersion-Threshold algorithm(DT) proposed by Dario D. Salvucci et al. [15].

When there is a fixation, the gaze points tend to cluster together in a specific timing interval.We utilize the sliding window technique to identify the fixation points. Two parameters are important in defining window size: the duration and the dispersion threshold . As mentioned in [16], the duration of a fixation varies from task to task. Specifically, in the scene viewing task, the duration is usually set to be at least 200ms. In our experiment, the duration is set at 350ms which is around 10 frames. Dispersion threshold emphasizes the dispersion (i.e.,spread distance) of a fixation point. In terms of the dispersion, we simply compute the dispersion in each window as follows:

$$Dispersion = max(x) - min(x) + max(y) - min(y) \tag{1}$$

We set the maximum dispersion as 20 pixel, which could generates a reasonable amount of fixations [17].

We ignore the noisy samples in the following two conditions. First, the eye tracking device fails to detect the corneal reflection. Second, the point is beyond the image boundary. We perform window sliding on the time axis, and obtain the fixation points based on both duration and dispersion threshold. The pseudo code for this algorithm is shown in Algorithm1.

This modified Dispersion-Threshold Identification (I-DT) algorithm is more computational efficient than the original DT algorithm because we directly set the new start to the frame that incurs termination or the frame next to it instead of shifting the window by one frame.

Algorithm 1: Fixation identification algorithm

```
Function Modified I-DT (duration, dispersion threshold) Result: Fixations
Initialize the start of the window to the first point while there are still points do
    Inspect current point i if the point is invalid then
        if n ≥ duration then
            n is the number of points in the window The fixation is the middle point of the window
        else
            Abandon the current window
        end
        Set the new start to i + 1
    else
        Update the current dispersion with point i if dispersion ≥ dispersionthreshold then
            if n ≥ duration then
                The fixation is the middle point of the window
            else
                Abandon the current window
            end
            Set the new start to i
        else
            Add point i to the window
        end
        Move to the next point i + 1
    end
end
if n ≥ duration then
    The fixation is the middle point of the window
end
return fixation
```

4 Methods

4.1 Data Augmentation

After collecting the training data, we split it into the training set and the validation set. For each object, we first count the number of images and randomly choose 75% of the images as the training set and the remaining 25% images as the validation set.

We first augment the training set by altering the color, contrast, and brightness of the original images [18]. And then, we generate five more crops, namely, top-left, top-right, bottom-left, bottom-right, and central by using two-thirds of the region along each axis. This augmentation strengthens our model because test crops may not be perfect and some of them contain only a portion of the object. With partial images included in our training set, our model can have better generalization.

We added nine times more images to the training set using the aforementioned two augmentation methods, which benefit the training of our CNN models.

4.2 Segmentation and Bounding Box Selection

For each video frame, the first step is to find the optimal bounding box around the fixation point. We adopt two different bounding box generation methods.

The first one is the fixed size method. We use multiple fixed cropping sizes from 64 x 64, 96 x 96, to 128 x 128 around the fixation and group them together for testing.

The second one is the deep learning based object detection method. We use YOLOv2 model pre-trained by the VOC 2007 and VOC 2012 datasets. We feed testing images into the network and generate detected bounding boxes. Every input image has multiple detected bounding boxes. So we need to filter out some nonrelevant detections. First of all, we filter out some severely occluded cropping boxes based on the RGB threshold. For instance, when the subject rotates the object, the captured image is very likely to suffer from occlusion with the subject' hands. Fig.6 illustrates this situation. Secondly, we select the bounding boxes based on their sizes. As YOLOv2 is a general object detection model, the system tends to output some unrelated objects such as table, bed, and person, which are relatively bigger than the objects we are interested in. So we set another size threshold to further filter out bounding boxes with large sizes.

After bounding box selection, there are still multiple bonding boxes on each single image. We need to choose correct box around the fixation point. For each fixation point, we set the fist bounding box contains the fixation as the initial crop. If there is another bounding box also containing the same fixation point, we deploy a special selection process to select the final bounding box. It works as follows: if there is an encompassment relationship of two boxes, we choose the smaller one as the new crop. Otherwise, we calculate the distance from the two centers to the fixation and choose the one with smaller distance. In an extremely rare case when the distance is equal, the two bounding boxes are merged. Fig.7 illustrate the final bounding box generated by the proposed algorithm.

This bounding box selection method is better than the simple union and intersection. As for the union, the cropping box is likely to grow very big and contains many unnecessary objects when multiple detection boxes contain the fixation point. On the contrary side, the intersection of the boxes cannot fully include the object of interest, and the partial image fails to represent the characteristic of the whole object.

Note that there are some special cases when the system fails to detect any object or none of the bounding box contains the fixation; we simply use the default 96 x 96 region around the fixation. We also consider the case when the fixation slightly deviates from the object, as shown in Fig. 8, so we try a 16-pixels soft boundary in all four directions.

4.3 Object Recognition

After augmentation, we have 9000 cropped images for training and another 297 images for validation. Due to the relatively small scale of our training data, we use the Alexnet structure for CNN modeling. The model is trained from scratch with a batch size of 128, base learning rate of 0.01, momentum of 0.9, and weight decay of 0.005. We adopt the step policy for learning rate decrement, reducing it by a factor of 10 for 1000 iterations. After 2500 iterations, the network is fully trained, which takes 64 minutes on a desktop with NVIDIA GeForce GTX 745.

Fig. 6. Some example images when the subject's hands occlude with the object of interest.

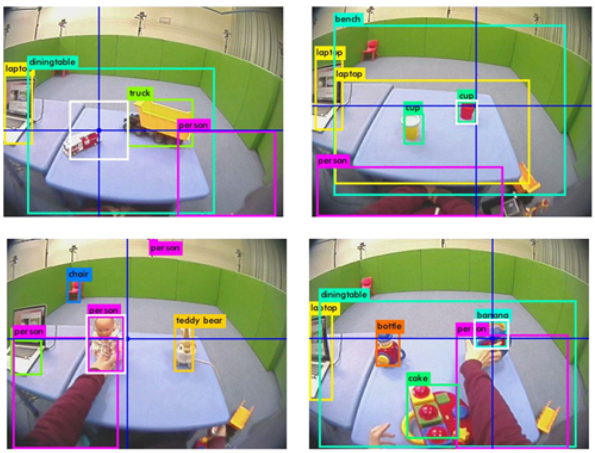

Fig. 7. Four final bounding boxes generated by our algorithm are marked as white rectangles. The rectangles with labels are the raw bounding boxes generated by the YOLOv2 detection system. The blue points are the fixation outputs from the wearable eye tracker.

Fig. 8. Sample images when the fixation point slightly deviates from the object.

4.4 Evaluation

In the testing step, we first obtain cropped images, then feed them into the network to receive output scores. We use classification accuracy as an evaluation metric for the recognition system and intersection-over-union (IoU) for the detection system. The algorithm works as follows: we go through all the ground truth bounding boxes of an image. If the fixation point is in a bounding box (with some deviation toleration), we first calculate the IoU between the detection output and this bounding box, and then evaluate the classification accuracy. If none of the ground truth bounding boxes contains the fixation point, we believe that the subject is looking at some objects not belong to our pre-defined class. In this case, we need to verify whether the proposed system successfully predicts it as class "Others". Algorithm 2 shows the pseudo code for this algorithm.

Algorithm 2: Test result evaluation algrithm

```
Function Evaluation()
for all gt ∈ groundtruths(GT) do
    if the bounding box of GT contains fixation then
        Calculate IoU
        Compare predication class of GT
    end
end
if None of the GT bounding box tontains fixation then
    Compare prediction with class label "Others"
end
```

5 Experimental Results

5.1 Cropped Image Generation

In our experiment, we find that using the general YOLOv2 detection system combined with our customized bounding box selection method improves the overall fixation aware object recognition performance by 9% absolutely compared to the fixed size cropping baseline as shown in Table2.

As for IoU, the proposed approach outperforms the baseline by 2%. Since we only calculate IoU when the final object recognition prediction is correct, the improvement on IoU is not that high compared to the accuracy. The main reason might be that when the prediction is correct, the region covered by the baseline fixed size cropping is very similar to our selected bounding box.

5.2 Object Recognition

Table3 shows the recognition accuracy for each of the selected 14 commonly used assessment tools. Our system achieves 100% recognition accuracy in the flowing classes: Baby, RedToy, Truck, ChestBox, FoodContainer, and Jar. It works

Table 2. Comparison of two bounding box selection methods

type	session 1 testing data	session 2 testing data	IoU
fixed size bounding box	83.3%	79.9%	56.9%
our proposed method	92.5%	88.2%	58.3%

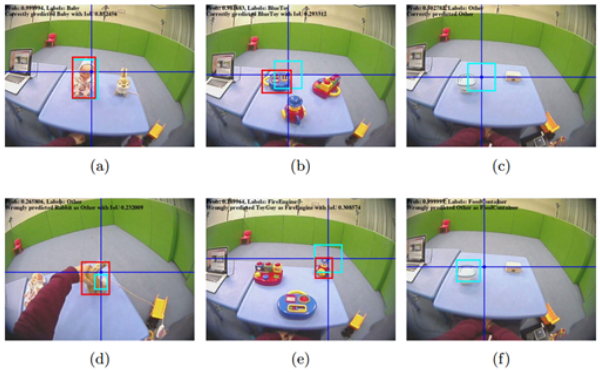

Fig. 9. Sample predictions of our system. (a)-(c) and (f) are the correct samples while (d)-(e) are the misclassified samples. The cyan rectangles are the bounding boxes generated by Section IV 4.2 while the red rectangles are the human annotated ground truths. The blue points are the fixation points.

fairly well in recognizing Rabbit, BlueToy, FireEngine, Book, Ball, and Others. However, the performance in ToyGuy, Plate, and Cup is not very satisfying. In Fig.9(a), the system detects the baby and the fixation point is in the region. However in Fig.9(b), the system fails to detect the blue toy, so we use a default region of 96 x 96 for test. Fig.9(c) is similar to Fig.9(b), except that there is no ground truth bounding box containing the fixation, so we classify it to class "Others". In Fig.9(d), the detection system outputs a small part of the rabbit, and coincidently the fixation is also in that region, which leads to the wrong prediction as class "Others". In Fig.9(e), we use a default region, but since it is quite larger than the toy guy, our system makes a wrong prediction that it is a blue toy. The example shown in Fig.9(f) is a soft boundary sample. The fixation point is within 16 pixels from the detection bounding box, and the system recognizes it as a food container.

As shown in Table 3, our proposed system achieves 92.5% and 88.2% accuracy for the overall fixation aware object recognition task for testing session 1 and 2, respectively. Because the data collection setup for the training data collection and testing session 1 are the same, so the recognition accuracy on testing session 1 is higher than testing session 2. The performance can be further enhanced if the outputs from the wearable eye tracking devices are more accurate, robust and with high resulution.

Table 3. Performance of our proposed method on the session 1 testing data, including the number of correct predictions, total number of test images, and the recognition accuracy.

Class	Correct	Total	Accuracy
Baby	23	23	100.0%
Rabbit	19	20	95.0%
RedToy	17	17	100.0%
BlueToy	24	25	96.0%
ToyGuy	13	18	72.2%
FireEngine	11	12	91.7%
Truck	10	10	100.0%
Book	8	9	88.9%
Plate	4	5	80.0%
ChestBox	3	3	100.0%
FoodContainer	12	12	100.0%
Ball	18	20	90.0%
Cup	11	13	84.6%
Jar	8	8	100.0%
others	30	33	90.9%
all	211	228	92.5%

6 Conclusions

In this paper, we introduce a new wearable eye tracking video dataset captured in a real autism clinic setting. We try to develop an algorithm that combines the eye tracking and computer vision technologies together to recognize the object tag when one is looking at. With the usage of deep learning object detection and recognition network, the proposed system could enable researchers to easily analyze and label the data. We believe that the proposed method shows great potential to extend our application beyond the computer screen setting to more general indoor settings, such as home, office, hospital, supermarket, etc. We can also consider outdoor activities, such as driving and campus walking. Future works include collecting more data, using more advanced network models and utilizing unsupervised clustering methods for better accuracy and efficiency.

Acknowledgement

This research was funded in part by the National Natural Science Foundation of China (61773413), Natural Science Foundation of Guangzhou City (201707010363), Six talent peaks project in Jiangsu Province (JY-074), and Science and Technology Program of Guangzhou City(201903010040).

References

1. A. L. Yarbus, *Eye Movements and Vision.* Springer US, 1967.

2. M. Hayhoe and D. Ballard, "Eye movements in natural behavior." *Trends in Cognitive Sciences*, vol. 9, no. 4, pp. 188–194, 2005.
3. K. Rayner, "Eye movements and attention in reading, scene perception, and visual search." *Quarterly Journal of Experimental Psychology*, vol. 62, no. 8, pp. 1457–1506, 2009.
4. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
5. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.
6. J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F. F. Li, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 248–255.
7. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *International Conference on Neural Information Processing Systems*, 2012, pp. 1097–1105.
8. Jia, Yangqing, Shelhamer, Evan, Donahue, Jeff, Karayev, Sergey, Long, and Jonathan, "Caffe: Convolutional architecture for fast feature embedding," pp. 675–678, 2014.
9. J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," pp. 6517–6525, 2016.
10. T. Toyama, *Object Recognition System Guided by Gaze of the User with a Wearable Eye Tracker*. Springer Berlin Heidelberg, 2011.
11. T. Toyama, T. Kieninger, F. Shafait, and A. Dengel, "Gaze guided object recognition using a head-mounted eye tracker," in *Etra 2012, Biennial Symposium on Eye Tracking Research Applications*, 2012, pp. 91–98.
12. M. Shdaifat, S. S. Bukhari, T. Toyama, and A. Dengel, "Robust object recognition in wearable eye tracking system," in *Pattern Recognition*, 2016, pp. 650–654.
13. "Positivescience eye tracker," http://positivescience.com.
14. L. Positive Science, "Yarbus eye-tracking software user guide," 2014.
15. D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," 2000, pp. 71–78.
16. K. Rayner and M. S. Castelhano, "Eye movements during reading, scene perception, visual search, and while looking at print advertisements." *Visual Advertising Hillsdale*, 2008.
17. P. Blignaut, "Fixation identification: The optimum threshold for a dispersion algorithm," *Attention Perception Psychophysics*, vol. 71, no. 4, p. 881, 2009.
18. A. G. Howard, "Some improvements on deep convolutional neural network based image classification," *Computer Science*, 2013.