ACTIVE OBJECT DETECTION

G. de Croon   E.O. Postma

MICC-IKAT, Universiteit Maastricht, P.O. Box 616, 6200 MD, Maastricht, The Netherlands [1]

Abstract

Most existing object detection methods passively scan images to find the target object. Passive scanning is computationally expensive and inefficient: at each sampling point feature extraction is performed, while the probability of detecting an object is very low. In this article we explore the viability of active scanning for object detection. In active scanning, each feature extraction is utilised to constrain the further scanning process and to detect objects. We present an active object detection method and identify two requirements for the successful application of active scanning for object detection: (1) local samples contain information on the location of the object, and (2) subsequent samples should decrease the distance to a target object. We show that both requirements are met in a license plate detection task. Our active scanning method attains a test performance of 91.75% on the license plate task. We conclude that active scanning provides a fast and efficient alternative to passive scanning.

1 Introduction

Object detection is the automatic determination of the image locations of objects that are instances of a predefined class. Numerous methods for object detection exist (e.g., [2, 4, 5, 7]), most of which scan a part of the image at some stage of the object detection process. The scanning for objects is performed in a passive manner: local features are extracted at all points of a sampling grid defined by regularly-spaced locations on the image. For example, in the object detection method of Viola and Jones [7] the entire image is scanned at all points of a grid. More efficient variants, such as the one in [4], employ global features to determine a region of interest. Subsequently, passive scanning is performed within the region of interest. Despite such pre-selection schemes, passive scanning remains computationally expensive and inefficient: at each sampling point computationally costly feature extraction is performed, while the probability of detecting an object is very low. In this article we explore the viability of active scanning for object detection. In the active scanning for objects, all feature extractions are utilised both to constrain the further scanning process and to detect the object. We present an active object detection method that maps local image samples to shifting vectors indicating the next sampling position. The method takes successive samples towards the expected object location, while skipping regions unlikely to contain the object.

We identify two requirements for the successful application of active object detection. Firstly, local samples should contain some information on the location of the target object. Secondly, subsequent samples should decrease the distance to the object. We evaluate our active object detection method by determining empirically to what extent both requirements are met and by assessing its detection performance on a license plate detection task.

The remainder of the paper is organized as follows. We present the active object detection method in Section 2 and discuss the experimental setup in Section 3. Then we test to what extent the two requirements are met in Section 4. Also, we determine the performance of the active detection method on the license plate task. The results obtained lead us to introduce and test an extension of the method in Section 5. Then, we discuss the implications of our results in Section 6. Finally, we draw our conclusions in Section 7.

[1] g.decroon@cs.unimaas.nl, postma@cs.unimaas.nl. This research is carried out within the ToKeN VindIT project (grant number 634.000.018) of the Netherlands Organisation for Scientific Research (NWO).

2 The Active Object Detection Method

The active object detection method consists of two parts: the first part extracts features from a local sampling window, called the fovea, in the image and the second part is a controller that computes the scanning shift vector from the extracted features. Figure 1 shows a schematic overview of the active object detection method and its two parts.

In the feature-extraction part of the method, a square window of size s^2 (with four inner quadrants) is extracted from the image and downsized by a factor r. The pixel gray values of each quadrant are concatenated to form a vector. We reduce the dimensionality of each vector using principal component analysis (PCA), yielding n components. The concatenation of the four vectors forms the input vector i (of length $4n$) to the second part of the method, the controller.

The controller is a feedforward multilayer neural network, as depicted in the right part of Figure 1. Boxes in the figure represent layers of neurons, lines indicate that the layers are completely connected. The controller has two output neurons (o_1 and o_2). The activities of these neurons specify the direction and magnitude of a shift of the sampling window in the image (dx, dy), with $dx = o_1 m$ and $dy = o_2 m$, where m is a scaling factor.

The active object detection method takes its initial sample from a random location and then proceeds by extracting the features from the window centered at that location, mapping these features on a vector towards a new location, moving towards the new location, and so forth. This cycle is repeated until a predefined maximum number of time steps has been reached.

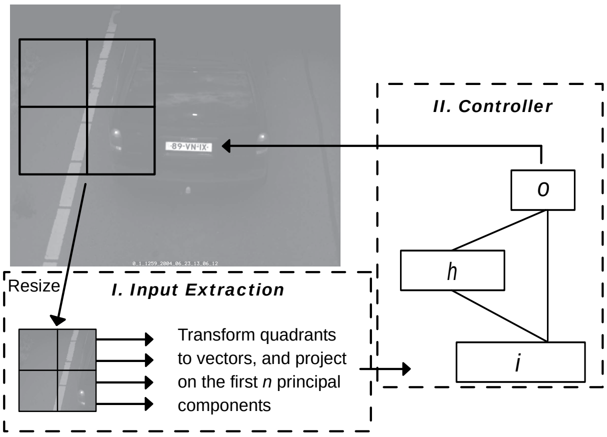
II. Controller
o
h
i
Resize
I. Input Extraction
Transform quadrants to vectors, and project on the first n principal components

Figure 1: Overview of the object detection method.

3 Experimental Setup

3.1 The License Plate Detection Task

To evaluate the active object detection method, we apply it to a license plate detection task. For this task, we use a private data set of the company Prime Vision BV (http://www.primevision.nl). The data set consists of 2930 labeled gray-scale images containing photos of cars, motor bikes, and trucks on the highway. All photos have been taken from behind and above vehicles on the highway, under different lighting conditions. Each image contains a white alphanumeric registration code at the bottom of the image. Figure 2 shows some example images from the image set, in which we changed some of the license plate letters and numbers for privacy reasons (all images in this paper are reproduced with permission of Prime Vision). Each image is 1300 pixels wide and 1030 pixels high. We do not preprocess the images before applying the active object detection method to it. In the experiments reported in this article, we divide the image set into a training set of 2640 images and a test set of 290 images.

3.2 Experimental Settings

In our experiments the width and height of the fovea (i.e., the sampling window) is set to one third of the image width, $s = 433$ pixels. The window is downsized by a factor $r = 0.4$, using

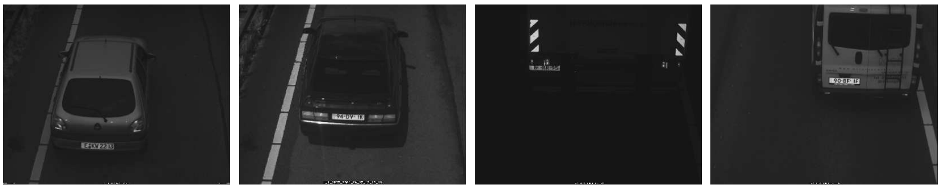

Figure 2: Example images from the data set.

bicubic resampling. To compute the principal components we apply iterative simple PCA [6] on a collection of uniformly sampled quadrants from the training set images and retain the first $n = 10$ principal components. The controller is defined as a neural network with hyperbolic tangent transfer functions in both the hidden and output layers. The output values are multiplied by the scaling factor m that is set to half the width of the image, $m = 650$ pixels. If the shift results in a location outside of the image, the location is reset to the closest location for which this is not the case. The maximum number of time steps T is four, which implies that the method performs three scan shifts.

3.3 Training the Controller

We train the controller to map a local sample directly to the center of the target object in the following manner. We sample the training set images at uniformly distributed image locations, taking into account that the sample window has to stay within the boundaries of the image. For each sample location we extract the input vector i as described in Section 2.1, and store it for training. In addition, we store the relative vector of the sample location towards the target location (this vector represents the ideal scanning shift). The extracted inputs and calculated shift vectors serve as inputs and targets for the neural network training procedure, respectively. We train the feedforward multilayer neural network with one-step secant backpropagation [1]. We used a learning rate of 0.01 and trained the network for maximally 300 epochs with the help of a validation set, consisting of one tenth of the gathered inputs and targets. Training stopped when the validation set error increased (i.e., early stopping) or if the maximum number of epochs was met.

4 Experimental Evaluation

In this section, we verify if the two requirements for the viability of the active object detection method are met. We first determine whether local samples of the image contain information on the object location. Then, we verify whether subsequent samples approach the location of the object. Finally, we determine the performance of the trained active object detection method on the license plate detection task.

4.1 Do Local Samples Contain Information on Object Location?

To determine whether local samples of the image contain information on the object's location, we investigate whether there are clusters of visual inputs that occur in a structural spatial relation to the center of the license plate. To this end, we first gathered 10 inputs per training image by sampling the image at uniformly distributed locations. On the basis of the resulting 26,400 inputs (our training set contains 2640 images), we determine k input clusters by using k-means clustering (see, e.g., [3]) with $k = 9$. To assess whether there is a structural spatial relation between the input clusters and the license plate location, we sample each test set image on all points of an evenly-spaced grid of 25 by 25 points, covering the whole image. For each input vector i we determine the nearest cluster center and store the x- and y-distance from the sample to the license plate, annotating it with the cluster number. In addition, we also store the absolute coordinates of the sample location, annotating it with the cluster number. Using these data, we are able to visualise the structural spatial relation of the input clusters relative to the target location and in absolute coordinates.

Figure 3 shows for each input cluster where it occurs most often in the image relative to the license plate location (high light intensity represents high occurrence). The center of each image represents the location of the license plate (indicated with a white cross and black circle).

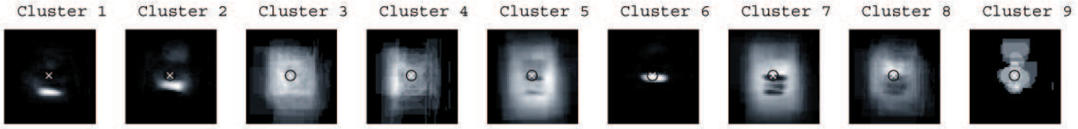

Figure 3: Distribution of the nine input clusters relative to the target location. The center of the license plate is in the center of each image and is indicated by a white cross surrounded by a black circle. For each coordinate, the light intensity is proportional to the frequency of occurrence of each cluster. These results were obtained with a sampling window size of one third of the image width.

Figure 3 reveals that some clusters have a clear spatial relation to the center of the license plate, whereas for other clusters the relation is less clear. Input clusters that occur within a confined region of the image in the figure contain information on the location of the license plate. In contrast, input clusters that have a uniform distribution in Figure 3, do not contain information on the location of the license plate. Since in Figure 3 none of the input clusters is uniformly distributed, all of them contain some information on the license plate location. For example, Figure 3 shows that input clusters 1, 2, and 6 are located at different distances to the bottom of the license plate, where cluster 6 is almost located on the license plate. The figure also shows that cluster 4 usually occurs to the left of the license plate. Furthermore, clusters 7 and 8 mostly occur above the license plate. Only 1% of the input vectors was mapped to cluster 9.

Figure 4 shows the occurrence of clusters in absolute coordinates. The figure shows that cluster 6 (and hence the license plate) can occur almost anywhere in the image, with a higher probability of occurring in the middle. We can also see that input cluster 4 usually occurs in the left part of the image, which sometimes contains the left line that marks the border of the road.

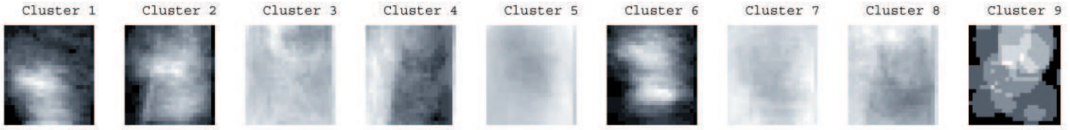

Figure 4: Occurrence of the nine input clusters in the test images in absolute coordinates. High light intensity represents high occurrence.

Our analysis suggests that the local samples taken from our image set contain information on their spatial relation to the license plate location. Since the sampling window is rather large in our experiments (one third of the width of the entire image), one could argue that the samples we take are not very 'local'. We assessed whether smaller windows can still contain information on the license plate location. Figure 5 shows the distributions of nine input clusters for a window size of one sixth of the image width. Clearly, there are still input clusters that are confined to small regions. However, the figure also shows that there is a relation between the window size and the distance at which local samples give information on the location of the license plate. This is reflected in the observation that confined clusters in Figure 5 are located closer to the license plate and are confined to smaller regions than those shown in Figure 3. The smaller window size seems to result in more information on the license plate location at a shorter distance and less information at a larger distance. These observations suggest an approach where the size of the sampling window depends on the distance to the object. We will adopt such an approach in our extended active detection method in Section 5.

4.2 Do Subsequent Samples Reduce the Distance to the Object?

Having established that local samples can contain information on the target location, we now turn to the second requirement for active object detection. In order to determine whether, in active

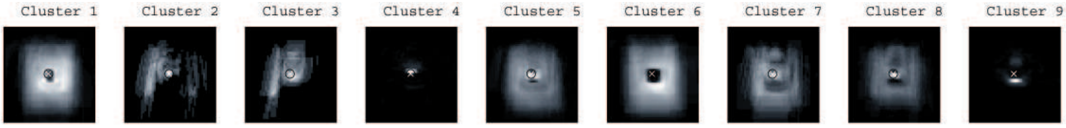

Figure 5: Distribution of the nine input clusters relative to the target location. The center of the license plate is in the center of each image and is indicated by a white cross surrounded by a black circle. For each coordinate, the light intensity is proportional to the frequency of occurrence of each cluster. These results were obtained with a sampling window size of one sixth of the image width.

scanning, subsequent samples reduce the distance to the object, we train the controller of our method on samples from the training set. Then, we apply the trained neural network to samples in the test set. For each sample we determine the distance to the center of the license plate before and after performing the shift as determined by the controller. Figure 6 shows a plot of the distance before and the average distance after performing a shift (dashed line). The standard deviation is represented by the gray, dotted lines. The solid line represents the case in which the distances before and after the shift were to be equal. The figure shows that the trained controller is unable to locate the license plate in one time step. If that were the case, the dashed line should correspond to the horizontal axis. However, it also shows that, for distances larger than 110 pixels, the average distance after the shift is smaller than the distance before the shift. Therefore, the distance to the center of the license plate becomes smaller if we take multiple samples in sequence. The fact that the dashed and solid line in Figure 6 intersect at 110 pixels indicates that on average, the method will not converge to the license plate location. This limitation will be discussed in Section 5. Figure 7 illustrates the shifts generated by the trained neural network for 10 × 10 initial points on a regular sampling grid superimposed on the central region of the image. The direction and length of each arrow emanating from these points represent the shift towards the next sampling location. The figure shows that most arrows point inwards towards the license plate, whereas a few arrows point towards the 'wrong' direction, e.g., the arrows at the bottom and the top left. These errors are due to the local nature of the sampling window. For instance, the wrongly-directed arrows at the bottom are due to the registration code at the bottom of the image.

These results indicate that in most cases, taking subsequent samples reduces the distance to the license plate location.

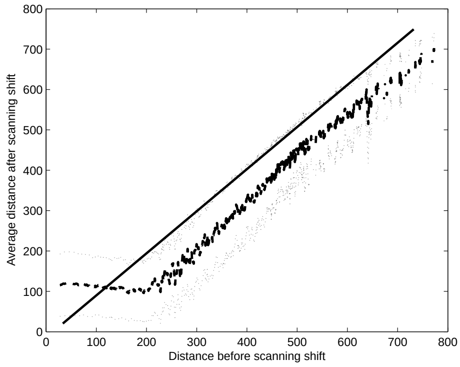

Figure 6: The dashed line is a plot of the distance before a shift and average distance after a shift in the image. The gray, dotted lines indicate the standard deviation. The solid line represents the case in which the distances before and after the shift were to be equal. For distances larger than 110 pixels, the average distance after the shift is smaller than the distance before the shift.

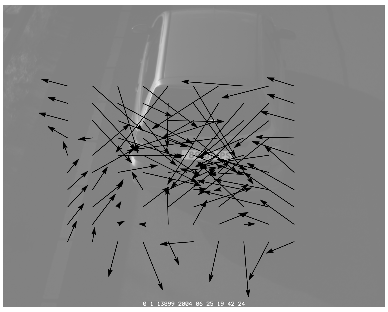

Figure 7: Arrows represent the magnitudes and directions of the shifts generated by the controller for 10 × 10 initial points on a regular sampling grid superimposed on the central region of the image. Most arrows point inwards towards the license plate.

4.3 Application of the Active Object Detection Method

Having established that the two requirements for active object detection are met, we now assess the performance of our method on the license plate task. Figure 8 shows four consecutive time steps of 10 different runs of the active object detection method. Per run, the center of the fovea is indicated with a white cross. The figure illustrates that the method approaches the license plate, but does not always reach the center of it. Testing 10 runs of the model on each image in the test set shows that the initial average distance to the center of the license plate is 323 pixels. After 3 scanning shifts, this distance is reduced to 179 pixels. This result agrees with the analysis performed in Section 4.1 and 4.2. Apparently, our method is capable to approach the target object, but it fails to reach it. In an attempt to understand what is causing this failure, we performed an analysis of the individual runs.

The analysis revealed that, generally, a few of the ten runs end up in a region far from the license plate. This is illustrated in Figure 8. At $t = 4$, two outliers are clearly visible at the top left. To compensate for these outliers, we compute the median of the ten locations. The median is represented by a circle in Figure 8. As can be seen, the median is rather close to the license plate. For testing our method on the test set, we use the median of 30 independent runs to indicate the location of the license plate. With 30 runs, at $t = 4$, the average distance of the median to the center of the license plate is 82 pixels, and only in 49% of the test images the median is located on the license plate. Since the detection performance is low, we propose an extension to the method to improve the performance in the next section.

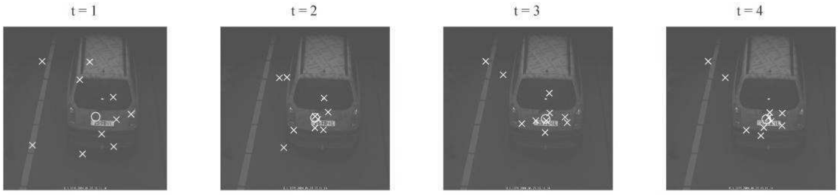

Figure 8: Four successive time steps of the active object detection method. At $t = 1$, the fovea is located at a random location. At $t = 4$, the method has performed three successive shifts in the image. The figure shows ten independent runs of the method, where the center of the fovea of each run is indicated with a white cross. The white circle represents the median coordinates of the fovea centers of all ten runs.

5 The Extended Active Object Detection Method

In subsection 4.3 we showed that on average our method fails in approaching the license plate sufficiently close to detect it. We discern at least two possible causes for this failure. Firstly, the neural network and the principal components are tuned on samples at various distances to the license plate, whereby the samples in the vicinity of the license plate are under-represented. As a result, the controller may perform worse near to the license plate. Secondly, the method may get stuck in a local optimum, even though the sample at that location does not resemble a license plate. These local optima affect the object detection performance. In this section, we first extend the active object detection method, so that it does not suffer anymore from the first cause of failure. Then, we further extend the method, so that it becomes less prone to the second cause of failure. We evaluate the impacts of the extensions, by determining the performance of the extended method on the license plate detection task. As in Section 4.3, we employ 30 independent runs and take the median of the locations reached at the final time step to indicate the license plate location.

The first extension entails employing a sequence of three neural networks, each of which is trained on samples at different distances from the license plate. The first and second neural network employ a sampling window size of one third of the image width, while the third network employs a sampling window size of one fourth of the image width. The first neural network is trained at uniformly distributed sample locations, as described in Section 3.3. Then, we determine the average distance to the license plate after applying this first network to the training images. We use this average distance to determine new principal components of samples extracted at or around this distance and to train the second neural network. More specifically, we now sample (for both PCA and network training) at normally distributed locations centered at the license plate location. The standard deviation of the sampling is equal to the average distance to the target location of samples after the application of the first neural network. The third neural network is trained in the same manner, but tuned to the sample locations reached by the second neural network. Figure 9 shows the application of the extended active object detection method for six time steps, again showing a cross per run and a circle to indicate the median coordinate. The first extension results in a performance of 85% on the test set. Clearly, this extension yields a considerable improvement over the performance of 49% as achieved in the original method.

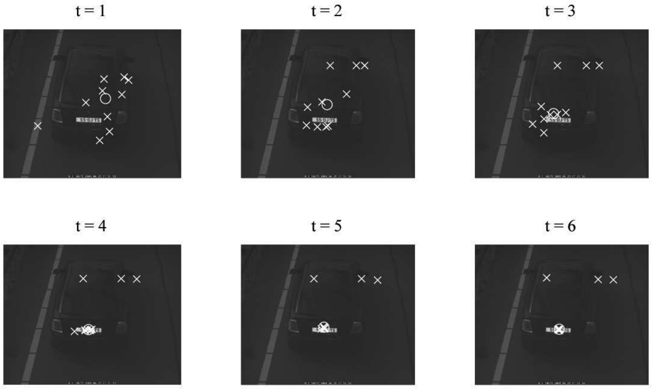

Figure 9: Example of the extended active object detection method. The first network takes one action, while the second and third network take two actions. The foveas of 10 example runs are indicated by white crosses, while the circle represents the median location of all 10 runs.

As stated before, our method can get stuck in local optima, even though the samples associated with these optima do not resemble the object at all. Therefore, we extend the method with a dedicated neural network that is specifically trained to estimate the distance to the target location. After training, the network maps the last sample of a run to an estimate of the distance d to the license plate. The feedforward multilayer neural network has 30 hidden neurons and one output neuron with hyperbolic tangent transfer function. The neural network is trained on the training set to output values $e \in [0, 1]$, according to the following function: $e = d/\theta$, if $d < \theta$, and $e = 1$ otherwise. We set θ, the threshold at which we cut off the distances, to one third of the width of the image. The effect of the second extension can be explained using Figure 9. The last image of Figure 9 shows that three out of ten runs end up far away from the license plate. The dark areas where they end up, clearly do not contain a license plate. If, at the end of multiple runs, we remove all samples for which $e > 0.10$, then these three runs are discarded. In this way we assure that we only accept sample locations very close to the center of the license plate, since $e = 0.10$ roughly corresponds to one fifth of the average license plate width. With the addition of the distance estimation network and by estimating the license plate location with the median of all runs with $e \leq 0.10$, the method attains a performance of 91.75% on the test set[2].

6 Discussion

The results in Section 3 and 4 show that active scanning can restrain the scanning process in a sensible manner. Moreover, our active scanning method has been shown to achieve a reasonable performance on a license plate detection task. There are two main issues that are of interest for the application of the active object detection method.

The first issue is the generality of the approach. In this paper, we have applied the method to a task of license plate detection. To determine the generality of the approach, it will be necessary to apply the active object detection method to other tasks. However, as mentioned in [4], most objects are situated in a more or less fixed context. This observation leads us to believe that our active object detection method can also be successfully applied to other types of problems.

[2] Note that the addition of the distance estimation network is necessary for tasks in which there can also be none or multiple objects in an image.

The second issue concerns the main advantage of the active object detection method with respect to passive object detection methods: the computational efficiency. As stated in the introduction, the passive scanning of traditional methods is computationally demanding, which may hamper application to large images and large data sets. Even though the active object method might need local samples with quite a large spatial extent and may also perform multiple runs to overcome local optima, the saving in computation time might be significant. In what follows, we provide a tentative indication of the computational costs of our active method as compared to (partially) passive methods. A traditional method that scans the entire image at all points of an $N_1 \times N_2$ grid performs $N_1 N_2$ feature extractions. In contrast, our active method performs $M(T-1)$ features extractions, where M is the number of method runs, and T the number of time steps. If we take a separation of sampling points of 10 pixels for the passive scanning, then N_1 and N_2 are both approximately equal to 100. This leads to about 10,000 feature extractions, whereas our active scanning method with $M = 30$ and $T = 6$ performs 150 feature extractions. Active scanning requires roughly 66 times less computational effort under these conditions. Our tentative indication of the computational advantage of active scanning over its passive counterpart is only intended to give an idea of the computational demands of both methods. A more detailed comparison of both types of methods on a variety of tasks is needed to firmly establish the relative performance and computational characteristics of the active scanning approach.

7 Conclusions

Our analyses and experimental results show that active scanning is a viable approach to object detection. The active object detection method meets the two requirements for successful application to object detection. Firstly, local samples of the image contain information on their relation to the object's location. Secondly, consecutive local samples reduce the distance to the object's location. We introduced an extended active object detection method that employs three cascaded controllers for determining shifts and that estimates the distance from the last sample to the center of the license plate. The extended method generally shifts its sampling window to the center of the license plate: it detects 91.75% of the license plates in our test set. We conclude that active scanning is a viable and computationally efficient approach to object detection. Our future research aims at further establishing the advantages and disadvantages of active scanning compared to passive scanning in a variety of tasks. In addition, we envisage refining the implementation of the active object detection method by employing different feature extraction techniques and controllers.

References

[1] R. Battiti. First and second order methods for learning: Between steepest descent and newton's method. *Neural Computation*, 4(2):141–166, 1992.

[2] N.H. Bergboer, E.O. Postma, and H.J. van den Herik. A context-based model of attention. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI), Valencia, Spain*, pages 927–931, 2004.

[3] A.K. Jain, M.N. Murthy, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3), 1999.

[4] K. Murphy, A. Torralba, D. Eaton, and W.T. Freeman. Object detection and localization using local and global features. In *Sicily workshop on object recognition.* Lecture Notes in Computer Science, 2005.

[5] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38:15–33, 2000.

[6] M. Partridge and R.A. Calvo. Fast dimensionality reduction and simple pca. *Intelligent Data Analysis*, 2:203–214, 1998.

[7] P. Viola and M. J. Jones. Robust real-time object detection. *Cambridge Research Laboratory, Technical Report Series*, 2001.