Abstract – Object localisation and identification is an important crucial problem for advanced mobile service robots. We implemented a neurobiologically plausible system on a robot that localises and identifies objects using a colour-based visual attention control algorithm and a hierarchical neural network for object classification. Object localisation and classification are performed in two stages. First low-resolution features are used to determine windows of attention in the robot’s camera image. Then high-resolution visual features are used for object recognition. The approach is evaluated in a test scenario where a robot is located in front of a table carrying different objects. The robot has to identify and manipulate these objects. We evaluated the total object recognition performance and compared the effectiveness of different feature sets. The approach performed very well regarding object localisation and classification results in this scenario and meets real-time constraints.

Keywords – Visual attention, selective attention, object recognition, hierarchical neural networks.

I INTRODUCTION

Detecting and identifying objects, in particular the recognition of 3D objects are essential skills for robots performing non-trivial tasks in real world environments. In order to be able to solve object-related problems the robot has to localise objects of interest in a complex visual scene and has to identify or categorise certain task-relevant objects. The objects are usually recognised from 2D camera images. Both object localisation and classification should be robust and reliable.

We present a two-stage visual object recognition system that localises and identifies objects using a colour-based visual attention control algorithm and a hierarchical neural network for object classification. First a window of attention is determined by means of low resolution colour and shape information. If an interesting region is found, this part of the image is analysed in more detail. High resolution features such as edges, corners or T-junctions are extracted from the determined window and used in a trained neural network for object recognition. This two-stage process reflects some properties of human or monkey vision such as guidance of eye-movements by visual attention and high resolution processing in the fovea versus decreasing resolution towards the periphery. The visual scene is scanned for salient regions and the fovea is successively centred on these regions, where the information is processed in much more detail than in the vicinity. Furthermore, this two-stage process helps saving computational power and thus facilitates sophisticated object recognition in real-time. To evaluate the approach, a test scenario has been defined where a robot is located in front of a table with different kinds of fruit and other simple objects on it. The robot has to perform a set of tasks: Localising and identifying fruit-like objects followed by object-manipulating tasks such as grasping, showing and moving specified fruits.
II Visual Attention Control

When locating relevant objects in complex visual scenes it would be too time-consuming to simply look at all possible regions. Therefore a pre-processing is necessary to segment the image into interesting and non interesting regions. To reduce the time expense of the following process steps the number of regions that do not contain a relevant object should be minimised. It also should be ensured that all regions that contain a relevant object will be detected in this step. This pre-processing which separates meaningful objects from the background is called visual attention control which defines rectangular windows of attention. These windows are called regions of interest. A region of interest is defined herein as the smallest rectangle that contains one single object of interest. Figure 1 illustrates the process of placing the region of interest.

![Image](image_url)

**Fig. 1.** Meaningful objects are separated from the background and are marked as regions of interest. The object colour is used to identify objects of interest. This also allows to detect partially occluded objects.

Important features can be selected by a top-down guidance of the first (attention) process. These low-resolution colour and shape features are used to determine a window of attention on the camera image. Both task-independent features such as colour or edges (bottom-up approach) and task-specific features like width-to-height ratio of the region of interest or specified object colour (top-down approach) are used. Depending on the task ranges are specified for the top-down features. If the extracted features fall into these ranges the detected region is regarded as potentially interesting. The following calculations are performed regardless of the task at hand, hence they can be regarded as providing bottom-up information. Regions of similar colour are identified using a flood-fill algorithm. The smallest rectangle containing the object is determined and the degree to which the object fills this rectangle. As top-down or task-related information the average colour of the object, the size of the region of interest, the ratio of width and height of the region of interest as well as the ratio of number of pixels belonging to the object and the total number of pixels in the region of interest can be used. Thus it is possible, for example to localise only objects of a specified colour.

The attention control algorithm consists of six consecutive steps. Camera images in the RGB format constitute the starting point of the attention control algorithm. At first the original image is smoothed using a $5 \times 5$ Gaussian filter. This also reduces the variance of colours within the image. The smoothed image is then converted to HSV colour space, where colour information can be observed independent of the influence of brightness and intensity. In the following mainly the hue components are considered. The next step is an adaptive colour clustering algorithm. For this purpose the colour of the object of interest and a tolerance are specified. These values can be set at run time by means of an application programme. The HSV image is searched for colours falling into this predefined range. Thereafter the mean hue value of the identified pixels is calculated. This mean value together with a predefined tolerance is now used to once again search for pixels with colours in this new range. The result is a binary image where all pixels that fall into this colour range are set to black and all others are set to white. Owing to this adaptivity object detection is more robust against varying lighting conditions. The results are further improved using the morphological closing operation. The closing operation has the effect of filling-in holes and closing gaps. In a last step a flood-fill algorithm is applied to identify coherent regions.

The region of interest is then determined by the smallest rectangle that completely encloses the region found. Simple heuristics are used to decide whether a such detected region of interest contains an object or not. For the problem at hand the width-to-height ratio of the region of interest as well as the the ratio of pixels belonging to the object to pixels belonging to the background in the region of interest are determined. If these ratios fall into certain predefined ranges depending on the objects to be detected the region is regarded to contain an object.

This method allows for detecting several objects in one scene and can handle partial occlusion to a certain degree. In the case of detecting partially occluded objects it is necessary that the objects have different colours.

In order to reduce the effort of the subsequent classification task and hence helps to save computation time the localised objects are tracked. The detected regions of interest are therefore stored and compared to the regions of interest identified in the following images based on the mean colours of the object, the size of the region of interest and the number of pixels belonging to the object. If these values match and the position of the region has only slightly changed the object is interpreted as the same object as in the previous
image. As images are available with a frame rate of 15 - 30 images per second the position of a non-moving object in the robot’s field of vision will only change slightly from image to image. Hence, on the basis of the previous images, expectations about the future position of the regions are calculated. These are also used for matching. If one object is regarded as the same in several subsequent images it is only classified once. New classification only takes place if the detected region differs too much from the previous one. This may have been caused by a new object or by an unexpected change in position of the object tracked.

In our system the purpose of attention control is the localisation of all objects of interest in a scene in real-time. Thus it is important to use features that can be extracted fast. The usage of simple features allows for fast feature extraction.

III Object Recognition

Once the objects are localised high-resolution visual features are extracted from each region of interest. This yields independence of the extracted features of scale and position of the localised object. The selection of the features depends on the objects to be classified. For the distinction of different fruits, appropriate features are those representing colour and the overall form. For the application at hand we use the mean colour values of the HSV representation of the detected region of interest as well as orientation histograms [4] summarizing all orientations (directions of edges represented by the gradient) within the region of interest. These features proved suitable for the task at hand. These characteristic feature vectors are classified by a trained neural network.

To determine the mean colour values the camera image is converted from RGB colour space to HSV colour space [5]. For each colour channel the mean value of the pixel belonging to the localised object is calculated. Colour information is helpful to distinguish e.g. between green and red apples. Advantages of colour information are its robustness to partial occlusion as well as its rotation and scale invariance. Furthermore it can be effectively calculated.

If neural networks are used for object recognition an object is represented by a number of features, which form a d dimensional feature vector \( x \) within the feature space \( X \subseteq \mathbb{R}^d \). A classifier therefore realises a mapping from feature space \( X \) to a finite set of classes \( C = \{1, 2, \ldots, l\} \). A neural network is trained to perform a classification task using supervised learning algorithms. A set of training examples \( S := \{(x^\mu, t^\mu), \mu = 1, \ldots, M\} \) is presented to the network. The training set consists of \( M \) feature vectors \( x^\mu \in \mathbb{R}^d \) each labelled with a class membership \( t^\mu \in C \). During the training phase the network parameters are adapted to approximate this mapping as accurately as possible. In the classification phase unlabelled data \( x^c \in \mathbb{R}^d \) are presented to the trained network. The network output \( c \in C \) is an estimation of the class corresponding to the input vector \( x \).

The object classification is performed by a hierarchical neural network [1]. It consists of several simple neural networks that are combined in a tree, i.e. the nodes within the hierarchy represent individual neural classifiers. For the application at hand RBF networks were used as neural classifiers. The basic idea of hierarchical neural networks is the hierarchical decomposition of a complex classification problem into several less complex classification problems. The hierarchy emerges from recursive partitioning of the original set of classes \( C \) into several disjoint subsets \( C_i \) until subsets consisting of single classes result. \( C_i \) is the subset of classes to be classified by node \( i \), where \( i \) is a recursively composed index reflecting the path from the root node to node \( i \). The subset \( C_i \) of node \( i \) is decomposed into \( s \) disjoint subsets \( C_{i,j} \), where \( C_{i,j} \subseteq C_i \). The total set of classes \( C_0 = C \) is assigned to the root node. Thus nodes at higher levels of the hierarchy classify between fewer but larger subsets of classes whereas nodes at the lowest level discriminate between single classes. This divide-and-conquer strategy yields several simple classifiers, that are more easily manageable, instead of one extensive classifier. These simple classifiers can be amended much more easily to the decomposed simple classification tasks than one classifier could be adapted to the original complex classification task. Furthermore different feature types \( X_i \) are used within the hierarchy. For each classification task the feature type that allows for the best discrimination is chosen. An example of such a hierarchy is shown in figure 2.

The hierarchy is generated by unsupervised k-means clustering. In order to decompose the set of classes \( C_i \) assigned to one node \( i \) into \( s \) disjoint subsets a k-means clustering is performed with all data points \( \{ x^\mu \in X_i | t^\mu \in C_i \} \) belonging to these classes. Depending on the distribution of the classes across the k-means clusters \( s \) disjoint subsets \( C_{i,j} \) are formed. One successor node \( j \) corresponds to each subset. For each successor node \( j \) again a k-means clustering is performed to further decompose the corresponding subset \( C_{i,j} \). The k-means clustering is performed for each feature type. The different clusterings are evaluated and the clusterings which group data according to their class labels are preferred. Since the k-means algorithm depends on the initialisation of the clus-
The classification result is obtained similar to the retrieval process in a decision tree [6]. Starting with the root node the respective feature vector of the object to be classified is presented to the trained classifier. By means of the classification output the next classifier to categorise the data point is determined, i.e. the classifier corresponding to the highest output value is chosen. Thus a path through the hierarchy from the root node to an end node is obtained which not only represents the class of the object but also the subsets of classes to which the object belongs. Hence the data point is not presented to all classifiers within the hierarchy. If only intermediate results are of interest it is not necessary to evaluate the complete path.

The hierarchical decomposition of the classification problem yields additional intermediate information. In order to solve a task it might be sufficient to know whether the object to be recognised belongs to a set of classes and the knowledge of the specific category of the object might not add any value. If the task for example is to grasp an apple, it is not necessary to distinguish between red and green apples. The hierarchy also facilitates a link between symbolic information and sub-symbolic information. The classification itself is performed using feature vectors which represent sub-symbolic information, whereas symbolic knowledge can be provided concomitantly via the information about the affiliation to certain subsets of classes. The usage of neural networks allows the representation of uncertainty of the membership to these classes since the original output of the neurons is not discrete but continuous. Moreover a distributed representation can easily be generated from the neural hierarchy. Since the hierarchy is generated using features which are based on the appearance of the objects such as orientation or colour information it primarily reflects visual similarity. Thus it allows the generation of a sparse similarity preserving distributed representation of the objects. A straight-forward approach is the usage of binary vectors of length corresponding to the total number of neurons in the output layer of all networks in the hierarchy. The representation is created identifying the strongest activated output neurons for each node. The corresponding elements of the code vector are then set to 1, the remaining elements are set to 0. These properties are extremely useful in the field of neuro-symbolic integration [7] [8] [9]. For limited task of object localisation and classification a distributed representation may not be relevant, but in the overall system in the MirrorBot project this is an important aspect [10].

IV Results

We evaluated the object localisation approach using a test data set which consisted of 1044 images of seven different fruit objects. The objects were recorded under varying lighting conditions. The images contained a single object in front of a unicoloured background. On this data set all 1044 objects were correctly localised by the attention control algorithm. No false-negative decisions were made, i.e. if there was an object in the scene it has been localised, and only 23 decisions were false-positive, i.e. regions of interest were marked although they did not contain an object. In order to handle these cases appropriately the classifier should be able to recognise unknown objects. This could be achieved by adding an additional class for unknown objects and training the classifier with examples of unknown objects or by evaluating the classifier output to identify weak outputs which are likely to be caused by unknown objects. The algorithm was also tested on images that contained more than one object. It was found that the number of objects within the image did not have an impact on the performance of the algorithm as long as the objects are not occluded. Likewise different background colours and textures did not impact the performance provided that the background colour is different from the object’s colour.
Figure 3 shows examples of the images used to evaluate the approach. They vary in type and number of fruits present as well as in background colour and structure. The hierarchical classifier was evaluated using a subset of the data. This subset consists of 840 images, i.e. 120 images per object showing different views of the object. Figure 2 depicts the hierarchy generated by the above described algorithm for the classification of seven fruit objects. Using 10-times 5-fold cross-validation experiments have been conducted on a set of recorded camera images in order to evaluate our approach.

We compared the performance of non-hierarchical RBF networks utilising orientation histograms or colour information respectively as feature type to the hierarchical RBF classifier architecture presented above. The classification rates are displayed in the box and whiskers plot in figure 5. Compared to simple neural networks that only used one feature type the performance of a hierarchical neural network is significantly better. The average accuracy rate of the hierarchical classifier was 94.05±1.57% on the test data set and 97.92±0.68% on the training data set. The confusion matrix in figure 4 shows that principally green apples were confused with red apples and yellow plums. Further confusions could be observed between red plums and red apples.

The approach was also evaluated in a test scenario where a robot is situated in front of a table [11]. Different objects lie on this table. Moving and looking around the robot has to identify and manipulate specified objects. Figure 6 shows this scenario. The approach is also successfully applied in soccer-playing robots of our RoboCup team. This proved that the robot is able to localise and identify fruits not only on recorded test data sets but also in a real-life situation. A video of this can be found in [12].

V CONCLUSIONS

This paper addresses the problem of object identification from 2D images. We presented an iterative object recognition approach comprising a colour-based visual attention control algorithm for object localisation and a hierarchical neural network for object classification. The approach features fast and robust object detection and recognition. The experiments carried out showed very encouraging results. On the image data set used the object localisation algorithm detected all objects of interest and only few regions that did not contain an object were selected. Also multiple objects per image as well as different background colours and
In the test scenario the robot is situated in front of a table. Different objects are laying on this table. The robot has to grasp or point to specified objects. Textures can be handled. The features used for object recognition proved suitable for classification of fruits. The approach meets real-time constraints since it is very fast. The classification accuracy of the hierarchical classifier could be significantly improved compared to non-hierarchical classifiers. Moreover the hierarchical organisation of the neural networks allows for a sparse coding of visual similarity as well as a linking between sub-symbolic and symbolic information. Hence the proposed approach forms a good foundation for further research in pattern recognition and neuro-symbolic integration.

The importance of top-down selection of saliency-features such as colours in practical applications is quite evident. It will be further evaluated in our next experiments. We will also evaluate the suitability of additional features such as colour histograms, form information and wavelets. In order to amplify the adaptivity of the approach we will enhance the approach to be able to learn new objects and thus adapt to new situations. Furthermore the object tracking will be enhanced in order to be able not only to track objects that stay within the field of vision but also objects that are momentarily out of sight.

VI ACKNOWLEDGEMENT

This research was supported in part by the European Union award #IST-2001-35282 of the MirrorBot project.

References