Adaptive Visual Attention Based Object Recognition

Rebecca Fay, Ulrich Kaufmann, Heiner Markert, Günther Palm
Department of Neural Information Processing
University of Ulm, Germany
{fay, kaufmann, markert, palm}@neuro.informatik.uni-ulm.de

Abstract – When performing tasks in complex environments robots are likely to encounter objects they have not seen before and consequently cannot identify. Thus the ability to learn novel objects during run time is an essential skill for advanced mobile service robots. Another helpful skill is the ability to track known and unknown objects since changes in the visual scene are very common due to motion of the robot and of possible objects of interest. Moreover, knowledge about the position of an already localised or classified object reduces the necessity of recalculation for every new image. We present a multi-stage visual object recognition system that localises and identifies objects using an adaptive colour-based visual attention control algorithm and hierarchical neural networks for object recognition and is able to track the localised objects as well as to learn novel objects during run-time. The approach is evaluated in a test scenario 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 localise and identify these objects as well as to perform a set of object manipulating tasks such as grasping, showing or moving specified objects. The experiments conducted showed encouraging results. New objects can be learnt with reasonable classification rates and in adequate time. The tracking of the objects allows for advanced object classification even on slower computers because classification is not exerted for every image.

Keywords: Selective visual attention, object tracking, object recognition, hierarchical neural networks, adaptive incremental online learning.

1 Introduction

Detecting and identifying objects, in particular the recognition of 3D objects, are important capabilities 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 complex visual scenes and has to identify or categorise certain task-relevant objects. When performing tasks in complex environments robots are likely to encounter objects they have not seen before and consequently cannot identify. Thus next to localising and recognising objects as well as processing language and planning actions, the ability to learn novel objects during run time is an essential skill for advanced mobile service robots. This enables the robot to incrementally learn new objects shown during run time and thereby increase knowledge of its environment and adapt to new situations. In real world environments which are fairly complex and subject to numerous changes only being capable of coping with previously learnt objects might not be sufficient. Another helpful skill is the ability to track known and unknown objects since changes in the visual scene are very common due to motion of the robot and of possible objects of interest. Moreover, knowledge about the position of an already localised or classified object reduces the necessity of recalculations for every new image.

We present a multi-stage visual object recognition system that localises and identifies objects using an adaptive colour-based visual attention control algorithm and hierarchical neural networks for object recognition. Initially 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 colour distribution are extracted from the determined window and used in a trained neural network for object classification. The visual scene is scanned for salient regions, where the information is processed in much more detail than in the vicinity.

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.

Once an object has been localised it is tracked. This tracking is a two-stage process. First the localised but not yet identified object is tracked without using model knowledge. Once the object is classified model knowledge is employed in order to facilitate the tracking of the object. Moreover the position of the object is memorised. Hence these two different tracking approaches can be interpreted as unconscious tracking and conscious
tracking with position memory.

The approach is evaluated in a test scenario 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 localise and identify these objects as well as to perform a set of object manipulating tasks such as grasping, showing or moving specified objects.

2 Attention Control and Object Tracking

An adaptive colour blob approach which segments the image on the basis of predefined colour ranges and evaluates the resulting binary images according to certain object specific characteristics such as width to height ratio is used to localise objects of interest within the robot’s camera image [6]. Afterwards the region of interest is determined which is the smallest rectangle containing the object. To identify possible regions of interest as well image-based as context-based information is used as shown in figure 1.

![Figure 1: Salient regions are localised within the camera image on the basis of low-level features. This image-based or bottom-up selection is facilitated by context-based or top-down information resulting in the selection of rectangular regions that are likely to contain objects of interest.](image)

After localising objects of interest these objects are tracked. Initially when the object is not yet classified it is tracked without utilising model knowledge. Therefore the detected regions of interest in one image are compared to the regions of interest of the previous image regarding size, width to height ratio and mean colour. If they correspond to a certain degree the object is marked on a simple map relative to the image centre together with its direction and velocity as well as the parameters used to compare the regions of interest. Only objects within the visual field are marked on the map. Objects no longer visible are deleted, objects still within the visual field are updated and newly occluded objects are added. If tracking is performed this way a necessary assumption is that the objects to be tracked are within the visual field.

In order to save computational time and to facilitate the tracking of objects, classification results are utilised. For each object within the visual field the classification result is calculated three times consecutively. The aggregated result is obtained by majority vote. More than one classification result is used in order to compensate misclassifications due to occlusion. If there is an unambiguous decision the resulting class is stored together with the other parameters. Every 30 images a new classification result is obtained. Only if the new classification result does not match the stored class again three consecutive classification results are requested.

Based on the knowledge about the object’s class objects can be tracked more easy and objects then do not necessarily need to be within the visual field. If an object is classified unambiguously and is present in five consecutive images it is marked on an egocentric map of the robot. The objects are then tracked with a Kalman filter [16] [1]. This yields less sensitivity against position noise and facilitates falsification in case of major divergences due to occlusion or re-appearance in the visual field.

3 Object Recognition

Hierarchically organized RBF networks [6] are used to classify the detected objects of interest. The hierarchy consists in case of a binary tree architecture of \( l - 1 \) internal nodes which represent a neural classifier, where \( l \) is the number of classes to be classified and \( l \) leaf nodes which represent the classes to be classified. The hierarchy is generated utilising unsupervised k-means clustering and conducts a successive partitioning of the primary set of classes into disjoint subsets with coarse discrimination at higher levels of the hierarchy and step-wise refined discrimination with increasing depth of the hierarchy. The classification result is obtained similar to the retrieval process in a decision tree yielding a specific path through the hierarchy.

The incremental learning of new objects is performed in two stages. In the first stage fast but less sophisticated methods are used to obtain initial results, i.e. the novel objects are learnt but the recognition rate might be weak. In this first stage the recognition rate can be improved by using a similar method to retrain the new object with additional data. In a second stage more complex algorithms are used to adapt the system and
to further improve the classification results.

Given the situation that an object is in the robot’s visual field and it is told the name of the new object by an instructor, the robot acquires sample data by looking at the object from different points of view. The number of samples gained for this new object is considerably lower than the number of samples used for learning the known classes.

Once the new data is obtained it is necessary to identify whether the presented object is already known or whether it is a new object. This is accomplished by presenting the new data to the trained classifier and taking the strength of the classifier response into account. Thereby a strong response is considered as an unambiguous decision and weak responses indicate a dubious decision which could be evoked by unknown classes or if the object to classify bears resemblance to more than one class. The thresholds for this are derived from the classifier responses when testing the known data. If a significant majority of the new data is unambiguously classified as a certain class the object is considered as familiar. Otherwise the object is regarded as a hitherto unidentified object. If an object is identified as unfamiliar it is learnt by fitting it into the hierarchy and if necessary retraining the affected nodes.

The new class associated with the unknown object is inserted into the hierarchy as a new leaf. The position of the new leaf is determined by classifying all new data and evaluating the corresponding paths through the hierarchy. The leaf will be inserted where the paths start to diverge. As complete identicalness for all data cannot be presumed even at the root node since the network has not been trained with this data a certain variance needs to be considered. Otherwise the new leaf would always almost be added at the root node. Thus the common path is successively determined by looking at each traversed node to which successor node the majority of the new samples is assigned by this classifier. This successor node is then added to the common path and the classification results for this node are obtained. If the classification result of one classifier is not completely consistent, i.e. not all new samples were assigned to the same successor node, this classifier is retrained. If there is no significant majority of samples assigned to the same successor node or if an end node is reached the new class is inserted as a new leaf.

As a consequence of the hierarchy composition only parts of the hierarchy need to be amended while the rest of the hierarchy remains unchanged. Figure 2 shows how a new class is inserted as a new leaf at the last node of the common path which the new samples take when being classified by the original hierarchy.

The retraining or incremental training of the classifiers is conducted by adding a new neuron to the hidden layer and then retraining the output weights with the joint sample set of old and new samples. The centre of the new prototype is determined by the mean of all new samples. The width of the corresponding gaussian function is set to the mean distance of the new samples to the centre and the output weights are learnt by calculating the pseudo-inverse [10] which is fast method already yielding good classification results.

The incremental training of the classifiers by adding a single prototype works well for data not spread widely. In order to account for scattered data the usage of more prototypes could yield better results. Hence two other ways of rapidly adding initial prototypes have been exploited, namely using all new samples as prototypes and performing a k-means clustering with \( k = N/2 \) clusters where \( N \) is the number of new samples. For each classifier to be retrained it must then be decided which method to use. Initially the method of adding one prototype is used. If optimal results are achieved with this method, nothing more is done. Otherwise the k-means method and the method of adding all samples as prototypes are also performed and the one achieving the best classification results is chosen.

A similar mechanism is applied when retraining already learnt classes. The only differences are that no additional leaf is added and that the path through the hierarchy is predetermined. The single classifiers on this path are retrained if there are any ambiguous or incorrect decisions. This retraining can be used in order to further improve the classification results of the new classes learnt online that are only represented by few samples.

An alternative approach for adding new classes to the hierarchy more resembling the hierarchy generation process is not to simply add the new classes as new leaf, but to add new nodes to the hierarchy by utilising unsu-
Figure 3: Example of the incremental learning of a new class I by adding new nodes. At first the position where to insert the new class is determined by identifying the common path which all new samples take through the hierarchy. Then the new class groupings are calculated by means of unsupervised k-means clustering, new nodes are added accordingly and the affected nodes are retrained. Node 4 is rebuilt and a new node 7 is inserted. The classifiers 0 and 1 are retrained.

The online learning phase is followed by an offline learning phase where more sophisticated learning algorithms such as three phase learning are used which will further improve the classification performance. All nodes to which the novel classes have been assigned are newly trained. These algorithms would be too time consuming for usage during run time.

4 Results

By means of classification experiments the suitability of hierarchical neural networks for extension was examined. It could be verified that new classes which are only represented by a few samples can be learnt sufficiently well in moderate time and that it is possible to learn new classes without negatively affecting the classification performance of already learnt classes.

The adaptive incremental learning approach was evaluated against a complete redesign of the classifiers. Both approaches showed essentially the same classification quality with average classification rates of 95.63 ± 2.49% and 95.54 ± 2.56% respectively where the redesign approach features longer training time as a complete rebuild of the hierarchy and training of all nodes within the hierarchy are required. The results are visualised in figure 5.

The confusion matrix for the incremental learning experiments displayed in figure 6 shows that although being represented by a significantly lower number of samples the classification rates of the new classes is equal to the classification rates of the primarily learnt classes.

The described approach was implemented on a Peo-
Figure 6: Confusion matrix for the experiments utilising incremental learning.

Figure 7: Adding of new classes to nodes. For each of the 10 × 10 cross-validation experiments conducted for each of the new classes 10 to 19 a leaf was added to different nodes of the classification hierarchy. For each new class is shown in percent to which node the corresponding leaf was added.

pleBot base by ActivMedia. The processor is a Pentium III 850 MHz with 128 MB RAM. The images have 384 × 288 resolution. On this robot the locating and simple tracking of an object took 62 ms on average. The time needed for localising and tracking an object together with the feature extraction averaged out at 102 ms. For the object recognition the mean time to learn a new object was 362 ms on a laptop with a Pentium 4 Mobile 1 processor 6GHz with 256 MB RAM. These results clearly show that the approach is able to meet the real-time requirements the robotic context gives rise to.

5 Related Work

In current literature there is much evidence for the usefulness of combining attention and object recognition.

In [14] [8] biologically motivated models of attentional selection of objects for recognition are described which combine spatial attention and object recognition in order to improve the recognition performance. The object recognition component is a hierarchical model inspired by the Neocognitron which utilises hierarchical combinations of bar-like features. Thus good recognition results could be achieved on artificial images. Furthermore the calculations are time-consuming and consequently not applicable for robotic applications where time is a critical factor. Employing another object recognition component that uses local scale-invariant features and is suitable for general object recognition yield good recognition results on cluttered real-world images [12] [15].

As the decomposition of problems into simpler sub-problems features advantages such as effectiveness and efficiency in learning and interpretability modular learning has attracted much interest recently. There are various ways of dividing a problem into less complex sub-problems. One possible way is a partitioning of the output space. In [7] a hierarchical decomposition of a multi-class problem into several two-class problems is performed utilising Fisher discriminant analysis in combination with a deterministic annealing process. The grouping of the classes is based on the class distributions resulting in a binary tree architecture. Simple Bayesian classifiers are used to solve the sub-problems. The approach is applied to the problem of categorising landcover using hyperspectral data. Instead of Bayesian classifiers support vector machines are used in [11]. The approach has been evaluated on several pattern recognition problems. An alternative method for the decomposition of the output space is applied in [3]. A max-cut algorithm is successively applied in order to find those class partitions that have a maximal distance. AS classifiers support vector machines are used. Another approach for building a hierarchical binary tree classifier architecture is proposed in [4] where a self-organising map is trained in the kernel space where classification by the deployed support vector machines takes place. On the basis of the trained self-organising map the class grouping is determined by identifying the grouping that shows maximises the inter-group distance while minimising the intra-group variance. In this architecture no disjoint partitioning of the classes is forced, but overlaps are allowed and are shown to improve the performance.

An example for an incremental learning approaches are ART networks. ART networks [5] [2] allow for online learning of evolving data sets. If a presented sample is similar enough to an learnt prototype this prototype is adjusted to the sample, otherwise a new prototype is defined by the sample. In the ARAM model [13] new classes can be learnt while preserving previously learnt classes and so the stability-plasticity dilemma is regarded. However, ART networks are non-hierarchical networks and they consider new samples one at a time.

6 Conclusions

The proposed approached has proven functional. Although the networks were trained with only a few sam-
amples of the new classes they were able to classify the new class and no considerable deterioration of the classification results of the former classes could be observed.

The experiments conducted showed encouraging results in particular that hierarchical neural networks are suitable for incremental adaption. New objects can be learnt with reasonable classification rates and in adequate time.

The proposed approach enables the robot to deal with varying object categories in addition to the predefined categories. New objects can be learnt rather quickly with satisfactory quality. The quality can even be increased by retraining with additional samples and in an offline phase more sophisticated learning algorithms can be used to further improve the classification quality.

The tracking of the objects allows for advanced object classification even on slower computers because classification is not exerted for every image.

References


