Object Recognition: Learned Hierarichal Networks
VIS4

Jens-Jakob Bentsen, Michael Kjær Schmidt & Mikael Westermann

Autumn 2014

Contents

1 Introduction **1**

2 Flat processing schemes vs. deep hierarchies **2**

3 Inspiration from the Primate Visual System **3**

4 Learned Hierarichal Networks **4**
4.1 Hierarchical compositionality 4
4.2 Validation 7

5 Conclusion **8**

6 References **9**

1 Introduction

To realize the vision of continuously expanding human abilities in robotics, surely the visual system is of great importance. Specifically the task of object recognition is essential to the humanoid robot.

The primate brain's ability to categorize and recognize objects could lead to the idea that the task in its general sense is easy and therefore also easily copied onto an artificial system ie. "Computer vision". However, it is not. Although the idea of implementing a computer vision system analogous to the biological visual system in the primate brain has existed since the 1970's there is still long way for such a system to be developed.

The theories proposed by neuroscientist David Marr had, until recently been found difficult to implement due to lack of computational power (Krüger et al., 2013). Hence the main focus in general computer vision has been on designing individual task oriented algorithms that perform analysis and recognition of such features as color, shape etc. The specific task of object recognition through pixel-information might seem easy as the human brain seems to do it instantly and effortlessly. However, the seemingly unlimited amount of different objects in addition to variation of size and posture in the 3D space is a great obstacle in the development of object recognition. Imagine the simple concept of a cup. As humans we instantly recognize the use of it even when we have never seen that specific type of cup, regardless of size, shape, handle/no handle, color etc.

Here, we give an overview of the results presented in Krüger et al. (2013) and Fidler et al. (2009) on the subjects of object recognition in computer vision systems using learned hierarichal networks.

2 Flat processing schemes vs. deep hierarchies

In the SIFT object recognition algorithm an object is identified by obtaining features known as keys, in an input image, followed by a search for these features in a database. When multiple statements point at a specific object, further effort will be made to specify it, and eventually an object will be identified, given that it is in the database (Wikipedia, 2014b). The complexity of this unbounded visual search (referred to as a flat processing scheme) is NP complete(Fidler et al., 2009).

In figure 2.1, on the right is shown a graphical representation of the flat processing scheme, where each task oriented algorithm computes with respect to a big database of known features. On the left is the representation of a hierarchal structure where common computations are made to support multiple tasks, hereby sharing information.

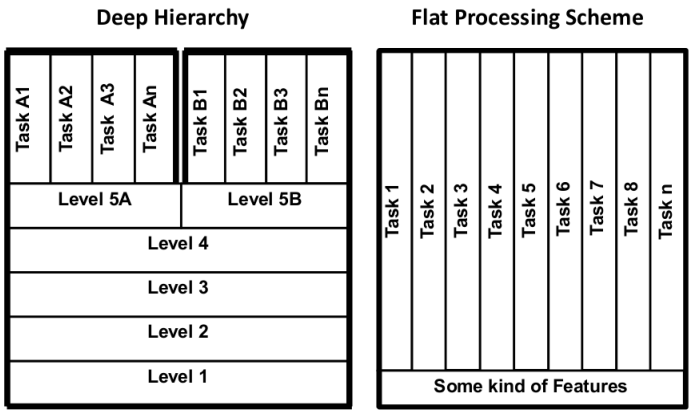

Figure 2.1: Deep hierarchies and flat processing schemes, borrowed from Krüger et al. (2013).

As will be discussed in section 3, the primate visual system is a deep hierarichal network, and this could mean that computer vision systems built using deep hierarichal networks could be a more optimized way to develop robust systems, than using flat processing schemes. As mentioned, the human brain seems to interpret visual stimuli with very little effort, and therefore trying to emulate the brain of primates might help in optimizing object recognition in computer vision. Especially the relatively recent development of multi-core systems in computer science greatly supports the idea of these structures as parallel processing is of fundamental character in the primate brain.(Fidler et al., 2009).

A hierarchical model consists of a number of layers placed "on top of each-other". Each item in a layer, representing more advanced features, is composed of items from the layers below, which represent simpler features, the smallest feature typically being an edge.

This form of representation gives some main advantages, taking the aspect of computational efficiency and storage space into account. As stated, layers of more advanced features are composed from elements of the layer below, hence only the fundamental building blocks are directly stored, along with information regarding connections of compositions. In addition, this method takes advantage of the fact that some lower level items will be reused in many of the items in layers above, which minimizes the need of duplicates. With this form of representation, results presented in Fidler et al. (2009) indicate that only very few actual descriptors are needed: only six are used in the lowest layer, and a few hundred in the other layers. In fact, the best "flat processing schemes" of today must store millions of small images with 25 x 25 pixels as descriptors in order to get satisfying results, which is in great contrast to hierarchies (Fidler

et al., 2009).

Another advantage of the deep hierarchy is what is refereed to as generalization. Generalization means that, like in the primal brain, we can make common calculations applicable to several individual tasks related to computer vision such as object recognition and categorization, grasping, manipulation, path planning, etc. (Krüger et al., 2013).

3 Inspiration from the Primate Visual System

Krüger et al. (2013) argues that deep hierarchies are beneficial to computer vision systems, and that several design principles of the primate visual system can be applied to computer vision systems and learning of deep hierarchies. These include hierarichal processing, separation of information channels, feedback and balancing coded structure with learning. Figure 3.1 roughly illustrates the principles of hierarichal processing and separation of information channels in the object recognition pathway, which includes the retina, LGN, V1 through V4, TE and TEO. V3 is omitted from the figure, as not much is known about it. However, information from V2 passes through V3 into V4. Note that the strict hierarichal structure as illustrated, is not an exact representation of the primate visual system, but rather shows some of the shortcuts between the levels of the hierarchy (Krüger et al., 2013). Note also that the feedback connections in the visual cortex are not visible in the figure.

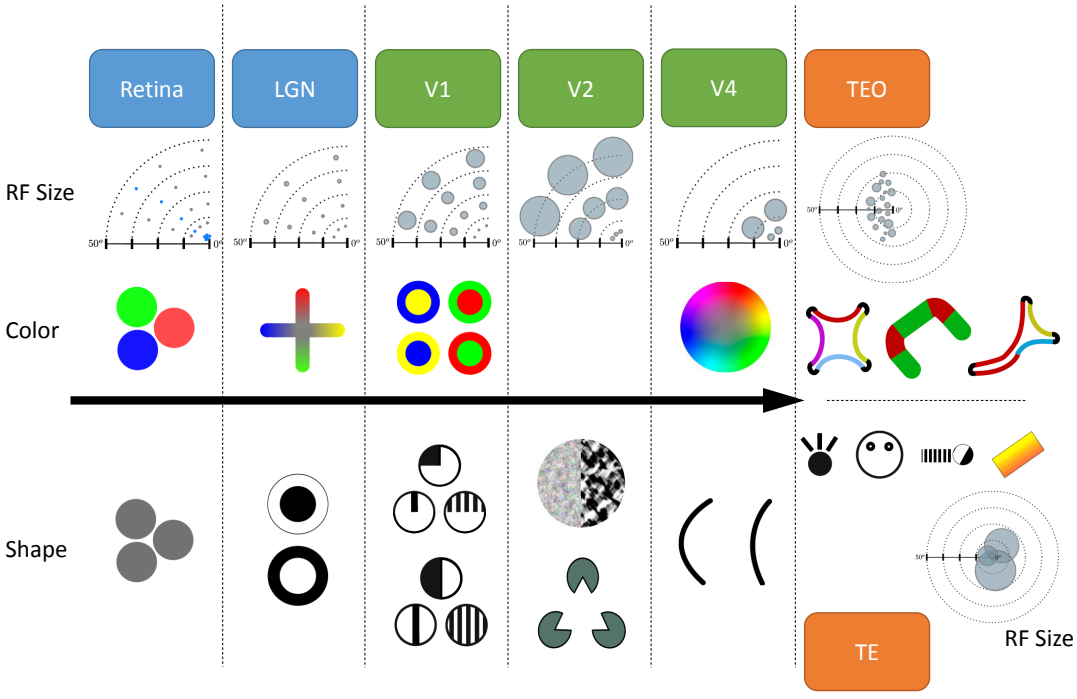

Figure 3.1: Simplified sketch of the hierarichal structure of the object recognition pathway in the primate visual system. The vertical lines separate layers from the low layers on the left, to the higher, more complex layers, on the right. Up until layers TEO and TE, the compositions can be split into two categories: color and shape. Also illustrated are the Receptive Field sizes (RF Size) of the individual layers. Adapted from Krüger et al. (2013).

Looking at figure 3.1, it is seen that the receptive field size varies throughout the layers. This indicates a spatial distribution of the computations. The increase in complexity of the color and

space information thoughout the layers indicates a sequential distribution of the computations as well. In summary, the object recognition pathway is parallelized and pipelined, and, as argued in Krüger et al. (2013), this might be a real plus in eg. future GPUs, taking advantage of parallel computations.

The separation of information channels provides a certain robustness to the visual system: If some visual cues are not available, for example, in the absence of color information, the structure allows the higher layers of the visual system to trust the more reliable cues, in this case, shape information. Another advantage of the separation of information channels, as explained in Krüger et al. (2013), is the efficiency of the information representations: Four colors and four shapes represented separately allows a more efficient representation of the 16 possible objects than if each object needed to be coded as a unique color/shape combination. The separation of channels is a more scalable approach.

With respect to learning frameworks in computer vision, it is difficult to directly apply the layers and streams of the primate visual system to computers, as these are not simply trained bottom-up or uniformly, which those of computer systems would typically be. However, the structure could serve as a guidance when designing a hierarichal learning framework for computer systems, as the learning problem is decomposed into sequences of simpler problems (Krüger et al., 2013). Furthermore, the principles of how the systems, biological and artificial, are built through learning, are already similar: There is some structural bias, which is genetically coded/manually designed by the programmer, and some exposure to visual signals/images, serving as training data sets.

4 Learned Hierarichal Networks

This section aims to explain the work presented in Fidler et al. (2009). There it is proposed to use hierarchical compositionality (explained in section 4.1) for object recognition.

4.1 Hierarchical compositionality

In essence, the term compositionality refers to complex parts made from combining other parts (Fidler and Leonardis, 2007, p. 2). The term is used within mathematics and languages as well, but in this paper, only its use with respect to extracting visual features is discussed.

An example of combining multiple parts into a more complex part is shown in figure 4.1. The letter "T" is formed from or *composed* by two simple line fragments. How to determine which parts to combine into a more complex part will be discussed below.

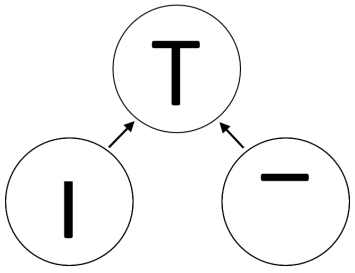

Figure 4.1: Combining two simple features into a more complex feature/composition.

In Fidler et al. (2009), a hierarchy is built, combining features in a lower layer into more complex

features in the layer above it. The motivation for using a hierarchy has already been discussed in section 2. Three different types of layers are used:

The bottom layer extracts simple features (edges) from the image. It uses a filter bank[1] consisting of 6 different Gabor filters of different orientation.[2] The filter bank is shown in figure 4.2. This filter bank is applied directly to the image and the most prominent features are extracted. This is done by comparing the response of the filters with a threshold value, and thereby the sensitivity and number of extracted features can be controlled. This is repeated for different image scales, to make the system robust towards the objects being of different sizes (scale-invariance).

The feature extraction/detection done at this layer is similar to area V1 of the primate visual cortex, see section 3.

Figure 4.2: Filter bank used for feature extraction at layer 1 (Fidler et al., 2009, fig. 7).

Category independent layers are located just above the bottom layer, and are learned without supervision. In Fidler et al. (2009), layers 2 and 3 are category independent layers. The features of a layer are learned by combining parts from the layer below it (ie. layer 2 is learned from layer 1 features, layer 3 from layer 2 features etc.). This is described in section 4.1.1. Examples of of layer 2 and 3 features are shown in figure 4.3.

The learning of these layers is unsupervised - this means that no input from humans is necessary when the layers are learned, and that the layers are learned from a set of unlabeled images containing a variety of objects.

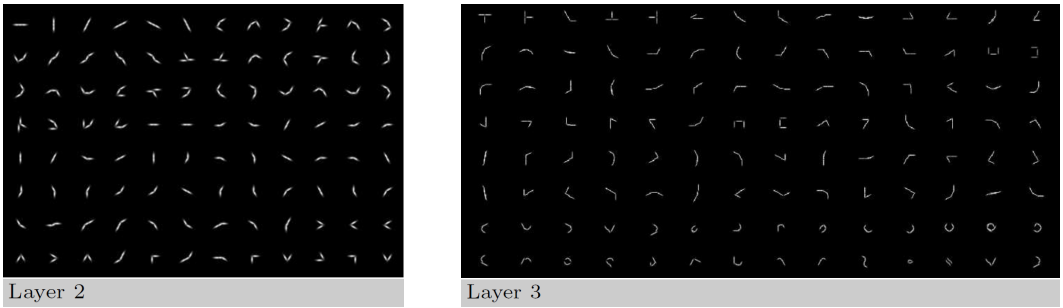

Figure 4.3: Learned features in layers 2 and 3 (Fidler et al., 2009, fig. 7).

Category specific layers are the highest layers (4 and 5) of the hierarchy. These layers are learned from images of specific categories. The top-most layer combines the parts through the center of the object, to form the final object.

[1] A filter bank is an array of filters that split the signal into different components (Wikipedia, 2014a).

[2] A Gabor filter is a combination of a Sine and a Gaussian distribution that is often used within computer vision for edge detection (Kämäräinen, 2012).

4.1.1 Composition of parts

Every part $\mathcal{P}_\ell^n$ in $\mathcal{L}_n$ (layer n) is composed of parts in $\mathcal{L}_{n-1}$. This means that all parts in $\mathcal{L}_5$ are compositions of parts in $\mathcal{L}_4$ and so forth. The composite part is made of a central part and its neighbours in the layer below. This list of sub-parts is shown in equation 4.1.

$$\mathcal{P}_\ell^n = \left(\mathcal{P}_{central}^{n-1}, \left\{ \left(\mathcal{P}_j^{n-1}, \mu_j, \Sigma_j \right) \right\}_j \right) \tag{4.1}$$

where $\mu = (x_j, y_j)$ is the relative position of sub-part $\mathcal{P}_j^{n-1}$. Σ_j is the max allowed variance of its position around (x_j, y_j). An example of a composite part is shown in figure 4.4 below.

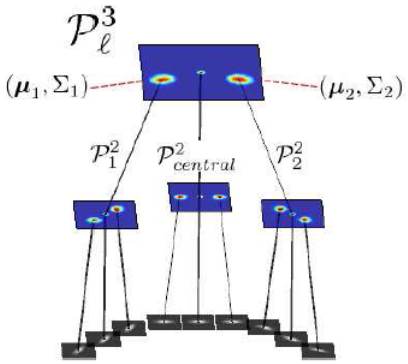

Figure 4.4: Example of a Layer 3 part, made from three Layer 2 parts (Fidler et al., 2009, fig. 2).

To avoid (nearly) identical parts being learned multiple times in different compositions, the learned compositions are grouped, based on similarity and co-occurrence. Compositions made up from the same sub-parts, but with different central parts, are grouped together. Parts often occurring together are also grouped. This grouping of parts lowers the number of parts in each layer, and makes the hierarchical representation more efficient.

4.1.2 Indexing and matching

When the hierarchy is built, recognition of the items is done using an indexing and matching scheme. Recognizing objects in an image is done in the following steps:

1. Extract $\mathcal{L}_1$ features by applying the filter bank. A list of filters producing the maximum response and their position is saved as $\mathcal{L}_1$ parts.
2. Every found feature is checked against the parts found in the layer above. Every part in $\mathcal{L}_N$ with part $\mathcal{P}_{\ell_k}^{n-1}$ as the central part is *indexed*, yielding constant lookup time.
3. The parts $\mathcal{P}_\ell^n$ having part $\mathcal{P}_{\ell_k}^{n-1}$ as center are then *matched*. In practice, this is done by checking if all subparts are present in layer $\mathcal{L}_{N-1}$.
4. Repeat step 2 + 3 for all remaining layers.

The process is also shown in figure 4.5.

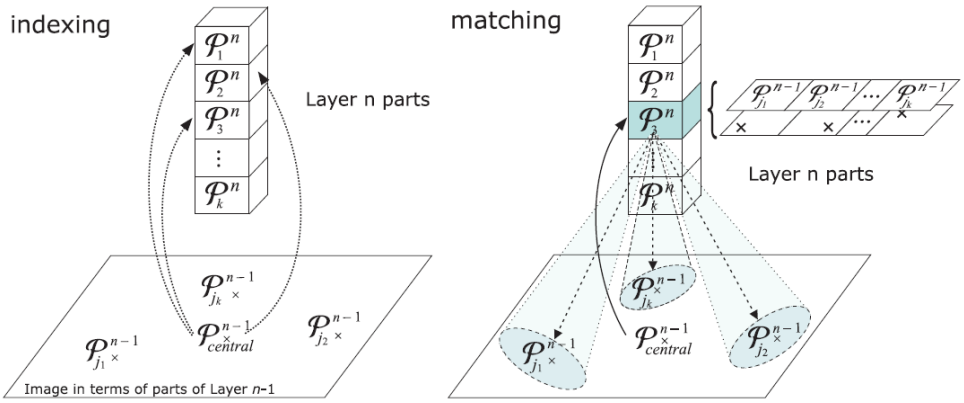

Figure 4.5: The left side shows how a found feature $\mathcal{P}^{n-1}$ is checked against parts in $\mathcal{L}_{N-1}$, where it is the central part. On the right side, the found part $\mathcal{P}_3^n$ is matched against the neighbour features in $\mathcal{L}_{N-1}$. If all subparts are found, the composite part is confirmed and added to the list of found $\mathcal{L}_N$ features.
(Fidler et al., 2009, fig. 3).

4.2 Validation

To validate the approach presented, learning of the hierarchy was tested by Fidler et al. (2009), using a set of 1500 images. The unsupervised part of the learned hierarchy consisted of 160 parts in Layer 2 and 553 parts in Layer 3. Some of the learned features are shown in figure 4.3.

The category specific layers were also tested. Multiple categories were learned, one of them being faces. The faces category was learned by using 20 images containing faces. Parts in $\mathcal{L}_4$ were then used to learn $\mathcal{L}_5$ by looking at the parts relative to the center of the face. The learned features for faces are shown in figure 4.6.

Figure 4.6: Learned parts in $\mathcal{L}_4$ and $\mathcal{L}_5$ for recognizing faces (Fidler et al., 2009, fig. 10).

5 Conclusion

Deep hierarichal networks can be used for object recognition in computer vision systems, and it is possible to build these hierarchies through unsupervised learning. The use of hierarichal compositional structures allow for efficient object recognition, as it inherently reduces the problem-space, both in terms of the number of objects to match an image against, and in the internal space consumption of the implementation. In addition to this, the hierarichal structure is well suited for parallel processing, and allows for a generalized computer vision system.

6 References

Sanja Fidler and Aleš Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

Sanja Fidler, Marko Boben, and Aleš Leonardis. Learning hierarchical compositional representations of object structure. *S. Dickinson, A. Leonardis, B. Schiele, and TM, editors, Object Categorization: Computer and Human Vision Perspectives*, pages 196–215, 2009.

Joni-Kristian Kämäräinen. Gabor features in image analysis. In *Image Processing Theory, Tools and Applications (IPTA), 2012 3rd International Conference on*, pages 13–14. IEEE, 2012.

Norbert Krüger, Peter Janssen, Sinan Kalkan, Markus Lappe, Aleš Leonardis, Justus Piater, Antonio Jose Rodríguez-Sánchez, and Laurenz Wiskott. Deep hierarchies in the primate visual cortex: What can we learn for computer vision? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1847–1871, 2013.

Wikipedia. Filter bank — wikipedia, the free encyclopedia, 2014a. URL `http://en.wikipedia.org/w/index.php?title=Filter_bank&oldid=638782102`. [Online; accessed 4-January-2015].

Wikipedia. Scale-invariant feature transform — wikipedia, the free encyclopedia, 2014b. URL `http://en.wikipedia.org/w/index.php?title=Scale-invariant_feature_transform&oldid=630596816`. [Online; accessed 4-January-2015].