Final Write Up

Bhargavi Madhunala

12/10/2016

Introduction

Humans make object recognition look trivial. We can easily identify objects in our surroundings, regardless of their circumstances, whether they are upside down, different of color or texture, partly occluded, etc. Even objects that appear in many different forms, like vases, or objects that are subject to considerable shape deviations, such as trees, can easily be generalized by our brain to one kind of object.

Object recognition is one of the fundamental challenges in computer vision. The main ideology of my project is to Recognize an object. Object detection involves verifying the presence of an object in image sequences and possibly locating it precisely for recognition. Object tracking is to monitor an object's spatial and temporal changes during a sequence, including its presence, position, size, shape, etc. This is done by solving the temporal correspondence problem, the problem of matching the target region in successive frames of a sequence of images taken at closely-spaced time intervals. These two processes are closely related because tracking usually starts with detecting objects, while detecting an object repeatedly in subsequent image sequence is often necessary to help and verify tracking. Object detecting and tracking has a wide variety of applications in computer vision such as video compression, video surveillance, vision-based control, human-computer interfaces, medical imaging, augmented reality, and robotics. Additionally, it provides input to higher level vision tasks, such as 3D reconstruction and 3D representation. It also plays an important role in video database such as content-based indexing and retrieval.

The main reason for our interest in object recognition stems from the belief that generalization is one of the most challenging, but also most valuable skills a computer can have. Since the area of vision probably depends on generalization more than any other area, this makes it a challenging endeavor.

Below we have listed the general characteristics of the research done for this thesis

• The ultimate goal of object recognition is to be able to recognize an object no matter what the circumstances (background, lighting, occlusion, etc.) As this is not trivial to achieve, certainly not without making any reservations, we will try a step by step approach, moving from simple shape recognition to more complex object recognition.

• As object recognition involves a lot more than just building a neural system other techniques are also discussed in this document. Since these other techniques are not always at the center of this research, these discussions will not always have the same depth.

• The created adaptive systems are tested as thoroughly as possible. Due to constraints of computational power and time sometimes tests have been less extensive in order to make way for testing more variations.

• The system will be tested using different kinds of tests. Simple tests are introduced to check parts of the system. In the end a final test with pictures of actual objects will show the workings of the complete system.

• All of the work described has been implemented and tested in Java.

Difficulties

The difficulties of object recognition are extensive. To avoid too general a discussion we will mainly look at them here in a light that makes sense when working with neural systems. Shapes can differ in appearance for several reasons. The most important reason is the difference in perspective we can have on a shape, i.e. shapes can be viewed from different angles and positions possibly making the shape appear bigger, upside down, tilted etc. But we also have to deal with shapes that, even though they might represent the same concept, actually are different. No two trees will ever look exactly the same, yet in the process of recognition we do want to label them using a single classifier.

A basic problem in recognizing a shape is caused by the fact that it can be in any kind of position, rotated, mirrored, etc. Such transformations belong to the group of affine transformations, which includes any kind of transformation that preserves distance ratios as well as co-linearity.

• Rotation

Objects appearing under different angles have very different representations when it comes to their pixels. Without taking this into account in some way, a neural network will incorrectly recognize identical patterns as different ones, depending on rotation.

• Mirroring

Similar to the possibility of an object appearing rotated, it can also be flipped, i.e. in a mirrored position. This creates a problem similar to that of rotation.

• Scale invariance

Different scalings of a shape create another problem of the same class. Shapes can appear in different sizes for many reasons. It can, for instance, be due to the perspective or their actual size can differ. Even though humans have little problem dealing with these and the previous transformations they are far from trivial to a computer.

• Translation

The problem of translation is the problem of an object appearing in a different place in the image. Although this is not one of the most difficult problems to overcome, it certainly should not be ignored when creating an object recognition system.

• Shearing and non-uniform scaling

There are also more difficult affine transformations mainly shearing and non-uniform scaling. Small transformations can generally be handled by having good shape descriptors in combination with the generalizing power of the learning system. Larger transformations are harder to accommodate, since there comes a point where it is questionable whether we are still dealing with the same shape.

In Point Tracking The detected objects are represented by points, and the tracking of these points is based on the previous object states which can include object positions and motion.

• Appearance tracking

The object appearance can be for example a rectangular template or an elliptical shape with an associated color. Objects are tracked by considering the coherence of their appearances in consecutive frames. This motion is usually in the form of a parametric transformation such as a translation, a rotation or an affinity. Silhouette tracking: The tracking is performed by estimating the object region in each frame. Silhouette tracking methods use the information encoded inside the object region. I've looked a whole bunch of image recognition approaches. But I'm trying to determine the best for this specific task. Most importantly, the object is made of lines and is not a filled shape. Also, there is no perspective distortion, so the rectangular object will always have right angles in the photograph.

Security

With increasing calls for security at places such as airports, government buildings, public transportation and public institutions, computer recognition has become an invaluable tool. Even though most of its applications are now still directed to face recognition, there is certainly also potential for object recognition to improve security.

Two examples of how object recognition can be used to improve security are:

• Airport scanners. Checking of luggage still depends heavily on human observation. Lack of concentration can let dangerous objects slip past security. Having an automated system could be a useful tool for assisting people that have to do this work.

• Security in banks. Cameras detecting the presence of guns and other arms can help alarm staff about people carrying suspicious items. Techniques for doing this are still under development, but are expected to become operative in the near future.

Lighting

A perennial problem in working with images is the various lighting conditions a picture can have been taken under. The effect of lighting on colors is extremely large, and especially in color based recognition is considered to be one of the most difficult problems. Human eyes within a broad range automatically adjust for brightness. A computer does not have that capability. This greatly reduces its ability to recognize colors. Good examples are the very popular soccer robots, the Aibos. In most cases they are programmed to find their location in the field by identifying colors of objects around them. Every new match to be played by these robots is preceded by a large operation of re-calibrating them to new lighting conditions. This takes a lot of precious time (it can easily take an hour) and can be troublesome because the time constraints often compromise the quality. Since we work with shapes instead of colors we are less dependent on these changing colors, but unfortunately this does not mean that lighting is not a problem for us. Mechanisms like edge extraction are sensitive to lighting conditions. Edges tend to disappear when lighting becomes dim and the difference in brightness of pixels tends to decrease. And brighter images can on their part lead to the extraction of too much edge information, as with a brighter image small color changes can become more visible.

Keras

Keras is a high-level neural networks library, written in Python and capable of running on top of either TensorFlow or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.

Uses

- Allows for easy and fast prototyping (through total modularity, minimalism, and extensibility).
- Supports convolutional networks and recurrent networks, as well as combinations of the two.
- Supports arbitrary connectivity schemes (including multi-input and multi-output training).
- Runs seamlessly on CPU and GPU.

Tensorflow

TensorFlow™ is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.

Conclusion

Recognition is based on decomposition of an object in geons. Recognition is more likely to be based on shape information than on color and texture. Recognition utilizes the relation among entities, in the form of orientation, distance, size and connections. Intersections of lines making up geons are important for recognition, more so than having long untouched lines fully intact.

References

Danang, V. (2013, April 18). Object Tracking.

K, S. Z. (2005, February). R. Gross, Face Databases, Handbook of Face Recognition. p. 22.

Rosebrock, A. (2015, March 23). Sliding Windows for Object Detection in Machine Learning.