Minor Project
Image Recognition

Software Design and Application

LOTTE FELIUS - 4423720
ROOS MARTENS - 4359682
YANNICK JANNETTE WALEN - 4231678
YUAN LIU - 4444485

15 January 2018 - TU Delft

Contents

Chapter 1

Introduction

The contemporary world of technology is rapidly evolving; things one could never imagine are now becoming reality. Computer science is the future and that is why the participants of this project chose to take a minor in Software Design and Application. For this minor, each student had the opportunity to start their own minor project about a subject that they found interesting. Initially, most groups started to think about small problems in their daily life that they would be able to solve with a software project, which ought to be feasible with both the (limited) knowledge of the students related to software, and the short time available for finishing the project.

However, an example of such a daily problem is that searching for a particular picture, while scrolling through a picture library, can be very time-consuming. For example, when looking for a picture of a shirt one wants to buy, it would be a lot less time-consuming if one could search on a word such as, in this particular case, 'shirt'. And thereupon, their phone will show all the pictures from their own library with a shirt in it. Summarizing, the problem can be formulated as: *"How to design an application which allows to search through a library of pictures, based on a search query as input, in order to quickly find a particular picture"*.

To solve this problem, a web service that respectively recognizes and labels pictures is designed, so that one can search for a particular picture way more efficiently. A more detailed approach of how it should work is the following: given an input which consist of a single word will act as a search query, and subsequently all the pictures from a certain library, which are all labeled with the word that is used as input, will appear. To accomplish this, it is necessary to use images, features of objects and machine learning, which will be further explained in the following sections.

Chapter 2

Related work

Prior to build the web service and its applications, it is necessary to indicate the gap between existing work and the what will be designed. Therefore, related work is extensively elaborated in order to indicate this gap.

To begin with, the most famous example of an application related is the commonly used image search from Google. However, Google's image search either relies on text surrounding the image or the file name itself. In short, their algorithm matches this text with the search query and subsequently shows the pictures that match (Rupiah, 2016). In addition, Google links all visually similar images.

A second example is a feature of the Iphone. Apple has recently updated their 'Photos app', which now has not only the availability to search photos by either a location or a person, where the latter is indicated by facial recognition, but also to search photos by labels which they call 'search by things' (Apple Support, 2017). However, the updated 'Photos app' is not working properly. Therefore, although this application is quite similar to the proposed project, there is substantial room for improvement.

The last example that is examined is an application called 'CamFind'. The purpose of this application is to recognize an object within a picture that one just took with their phone, to subsequently search for that particular object on the internet. Hence, this application provides the user of information from a particular object recognized in a picture, so that typing queries into a browser is not necessary.

Furthermore, there are a number of other applications that recognize objects in pictures. However, after analyzing existing applications related to image search, one could state that there is still no properly working application for quickly searching trough a (private) photo library. Therefore, the application ought to provide both reliable object recognition for labeling pictures correctly, and the availability for the user to search through a picture library in order to efficiently find a particular picture.

Chapter 3

Proposed methods

In order to accomplish to build an application that provides such facilities as described in the previous chapter, several tools, methods, and (parts of) applications provided on the internet are used. In the sections below, those will be further explained.

3.1 Machine learning

The first subject that was extensively elaborated is object recognition in images. A reliable object recognition method is to use Machine learning, which is used for building this application.

Machine learning is a broad research field of artificial intelligence that deals with the development of algorithms which can make a computer learn things without explicitly having to program it. In this project, it means that the computer could learn to recognize pictures. While this is an easy task for a human, for a computer this can be challenging: by relying on large databases and noticing patterns in different pictures in the database, a computer can be able to recognize images.

Neural networks are commonly used within machine learning, as they are now. Within this project, neural networks take a large database of images as a training set and learn from these pictures, so the computer is able to recognize the same sort of images later. Furthermore, by having a larger set of training images, the network can learn more and therefore improve its accuracy. To use neural networks within this project, a properly working open source neural network library is necessary.

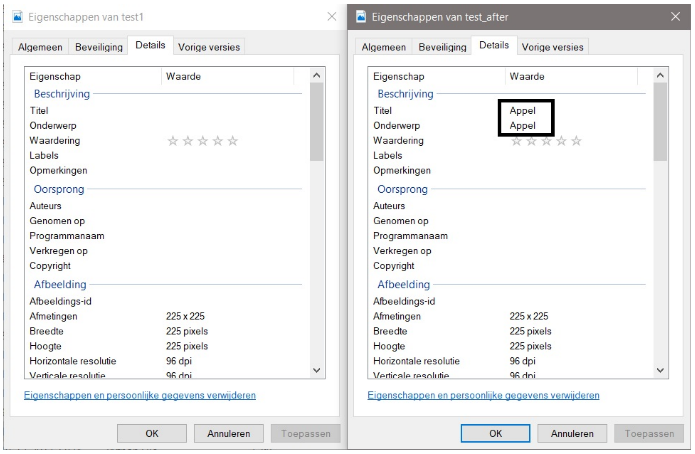
3.2 TensorFlow

One of the best known open source software libraries for machine learning applications is TensorFlow. It is used for research and production at Google.

TensorFlow is a library for array data calculations and computations that can be used to conduct neural network and deep learning. It provides low level programming to work with mathematics as well as methods for defining neural network layers. Therefore TensorFlow is good to use with unstructured data like images. However, TensorFlow is not that easy to use. This is where Keras comes into the picture: it is a more user-friendly API built on top of TensorFlow. Unfortunately, Keras did not work, hence only TensorFlow itself is used. Still there are a few advantages, such as increased control over neural networks. TensorFlow offers more advanced options and you can tweak it as you like, which comes in handy if you are developing some special deep learning methods. Moreover, operations on gradients or weights can be done very easily with TensorFlow.

3.3 Image recognition and adapting to video feed

Furthermore, a Python script which uses TensorFlow to recognize certain images is created. Tensorflow's object recognition neural network is used, which returns a score from 0 to 1 stating how much the images resembles the object. The threshold can be modified to user's preference, a higher threshold will result in less, but more accurate, results and vice versa. Moreover, a feature is implemented that has the capability to add a tag to an image's EXIF Data. EXIF data is short voor exchangable image file, it is a format which stores all information of a JPEG file. This is important since the user should be able to search on a tag and find the picture he is looking for quickly. After running the python script with a certain image or collection of images, the tag(s) will be added and the user will be able to search the images locally, which is quicker (figure 1).



Eigenschap	Waarde
Beschrijving	
Titel	
Onderwerp	
Waardering	☆☆☆☆☆
Labels	
Opmerkingen	
Oorsprong	
Auteurs	
Genomen op	
Programmanaam	
Verkregen op	
Copyright	
Afbeelding	
Afbeeldings-id	
Afmetingen	225 x 225
Breedte	225 pixels
Hoogte	225 pixels
Horizontale resolutie	96 dpi
Verticale resolutie	96 dpi

Eigenschap	Waarde
Beschrijving	
Titel	Appel
Onderwerp	Appel
Waardering	☆☆☆☆☆
Labels	
Opmerkingen	
Oorsprong	
Auteurs	
Genomen op	
Programmanaam	
Verkregen op	
Copyright	
Afbeelding	
Afbeeldings-id	
Afmetingen	225 x 225
Breedte	225 pixels
Hoogte	225 pixels
Horizontale resolutie	96 dpi
Verticale resolutie	96 dpi

Figure 3.1: figure 1

Moreover, a feature was implemented so that objects could be detected and recognized in a (real-time) video feed using our web cam. However, the implementation of this feature was not necessary to reach our final goal.

3.4 Web service

Also, a web service is created in which a photo library can be uploaded. For this Socket.IO and Node.JS are used. Socket.IO ensures that there is real-time communication between a server and one or multiple clients (Node.JS and an HTML web page).

To keep track of how much a file is already uploaded, full control is given to the Node.JS server to request specific blocks of data, and the HTML form will see these requests and send the needed information to the server.

The way socket.IO works is: either the server or the client ”emits” an event, and then the other side will pickup this event in the form of a function with the option of passing data back and forth.

After the basis was coded, the web service itself could be created. The web service works by connecting to the server through a browser. Through the browser multiple images files can be uploaded. Once the upload completes the python code for object detection is executed using the uploaded files. Once this completes these images will have been tagged with the detected objects. The images are the put in a folder on the server which can only be accessed by the user to whom the images belong. The user has the ability to now search for detected objects in his images. There is also an option to download all the images with their new tags. Windows can now use it’s own search function on these images to find detected objects.

Chapter 4

Conclusions

The project turned out to be a success and we managed to make a web service which recognizes images from your own collections and tag them. The tagged images can than be downloaded in a zip file, which allows the user to search for the desired images.

However, Tensorflow's object detection has a limited amount of 'detectable' objects. The way to add objects to this list is feeding Tensorflow's neural network 300 to 500 images, which are manually labeled. We looked for a way to make this an automatic process, unfortunately no solution was found. Therefore, only the existing list of objects was used.

Bibliography

[1] Apple Support., https://support.apple.com/en-us/HT207368.

[2] Rupiah, K., *How does the Google Image search algorithm work - and is it racist?* https://mg.co.za/article/2016-05-26-how-does-the-google-image-search-algorithm-work-and-is-it-racist Mail & Guardian, 2016, May 26.

[3] EXIF Data Viewer., *http://exifdata.com/*

[4] Github Repository Image Recognition., https://github.com/YWalen/image-data-science

[5] Github Repository Image Recognition., *all files related to software,* https://github.com/YWalen/tf-imagefind