First report delivery
Image recognition of touristic places

Ximena Cervantes Díaz, 276978

Fundamentals of Artificial Intelligence, 281301
September 13, 2019

Project advisor: Dr. José Ignacio Nuñez Varela

Abstract:

Since the beginning of the time, the humanity are looking for the perfect definition of the intelligence, how does the mind works, human behavior, etc. We create the technology as a tool for us, to make our lives more easy and to do things that we can't do.
The computation science was born a few years ago compared to the history of humanity, it was a fast process and year after year it continues growing.
Artificial Intelligence is an important thing in software, but this requires a lot of logic and mathematical operations. We take inspiration of the structure of the human brain to make artificial neural networks which are applied of a lot of thing like pattern recognition, coding, control, optimization, etc., they can learn directly from data easily. Fortunately, we can access to an different data bases or previous projects that will give to us information or we could use it to our objectives. In this case, some compilers and tools was used for this project. Lets define each of them and the solution of the problem.

Problem
It's difficult to the human mind to know which place or monument is looking at, especially if we don't know or remember the name of it. As a turist, it could be useful to know the name of the place where we at, knowing about its history or some stuff that help us to grow the knowledge of culture.

Solution
We can use the artificial intelligence as a tool in this case, with image recognition. All users can take a picture of the monument, landscape or turistic building and the program will compare it the phisical caracteristics to determinate which one is. Those images will be recognized despite changes in scale, camera viewpoint, illumination conditions, etc.

Pattern Recognition

As Cornelius T. Leondes sayd: *"Pattern recognition (PR) is the science and art of giving names to the natural objects in the real world. It is often considered part of artificial intelligence."* (2) In a few words, pattern recognition is a scientific discipline which has the objective to classify objects into a number of categories or classes.
"Is an integral part of most machine intelligence systems built for decision making."(8) This definition means that the pattern recognition uses artificial intelligence, the system has to be trainned to recognize some different objects. Same as artificial intelligence, machine vision is of importance. Machine vision system take images via a camera and analyzes them to make descriptions of that image taked before. This is important in manufactury industry, where they need to do visual inspection or for the automation in the assembly line. There, the images that was taken, have to be analyzed online and the a pattern recognition system has to classify the objects into "defect" or "nondefect" class.
Another important area of pattern recognition is character recognition which has major implications in automation and information handling. Pattern recognition software takes over to recognize the characters, storing that document has a scanned image. Another area is handwriting

recognition systems, where the data will entry not via keyboard, by writing.
There are a lot of others areas like: computer-aided diagnosis, speech recognition, data mining and knowledge discovery, etc., but the principal topic of this project is image recognition, refers to the ability of computer to decipher and understand the information fed to it from an image.

Image Recognition

In context of machine vision is the ability of software to identify objects, places, faces, etc. The computers can use the algorithms in combination with camera and artificial intelligence software to recognize that object. Is one of the most used area at pattern recognition, but its a difficult task compared to the human or animal brain. It requires deep machine learning, it uses neural net processors. There are a lot of image recognition algorithms which compare 3D models, includes smart photo libraries.

Image processing

There is a lot of definitions of image processing but all match in something: is the processing (analyzing and manipulating) of images with algorithms in a computer.
It has applications in many disciplines and fields in science and technology like a television, photography, robotics, medical diagnostics, industrial inspection. For this processing, it's helpful to use Python fot this case, Python has a lot of processing libraries and it's simply to codify.

Python

Python is a programming language very used at technology. It is optimize for software quality, program portability, etc. Is deployed in a wide variety of products and roles because is simple for understanding, readable and maintainable syntax, multiparadigm design; these convert its into a flexible and agile language.
In Python there are many libraries that we can use for image processing. Some of them are: `numPy` (for storing image), `matplotlib` (for display

purposes), `opencv`, `scipy`, `pil`, `scikit-learn` (for building machine learning models for image processing).

OpenCv

At the oficial page of OpenCv, they define it as: *"OpenCv (Open Source Computer Vision Library) is an library that includes several hundreds of computer vision algorithms".(7)*
Also, Bradski sayd: *"Is an opensource written in C and C++. It runs in Linux, Windows and Mac OS. There is active development on interfaces for Python, Ruby, Mathlab and others."(1)*
As after are mencioned, Python uses different processing libraries which are a dependency of OpenCv: NumPy, SciPy, OpenNi, SensorKinect (not used in this case).
As OpenCv was developed mainly to provide computer vision algorithms, its machine learning functionality is restricted to a single module, called `ml`.

CMake

CMake is a tool for building software from sources, surpassing in usage and popularity many other similar tools. Is a build-system generator to describe what the build system should achieve; runs on Linux, Mac OS and Windows.

SIFT method

The images can be similar, sometimes they have different scales and rotations; an Scale Invariant Feature Transform (SIFT) is needed to use in this case. SIFT is an involved algorithm which the thresholds are learned from the pair of images. Enables one to extract local information from digital images. It normalice local patches around robust scale convariant image key points.

SURF method

Speed Up Robust Features method is based on extraction of interesting points from an image, used for description of these points. Is optimized and faster than the SIFT method.

FAST method

Features from Accelerated Segment Test is a corner detection method, which could b used to extract feature points and later used to track and map objects in many computer vision tasks. One of the advantage of the FAST corner detector it's computational efficiency.

First steps to this project

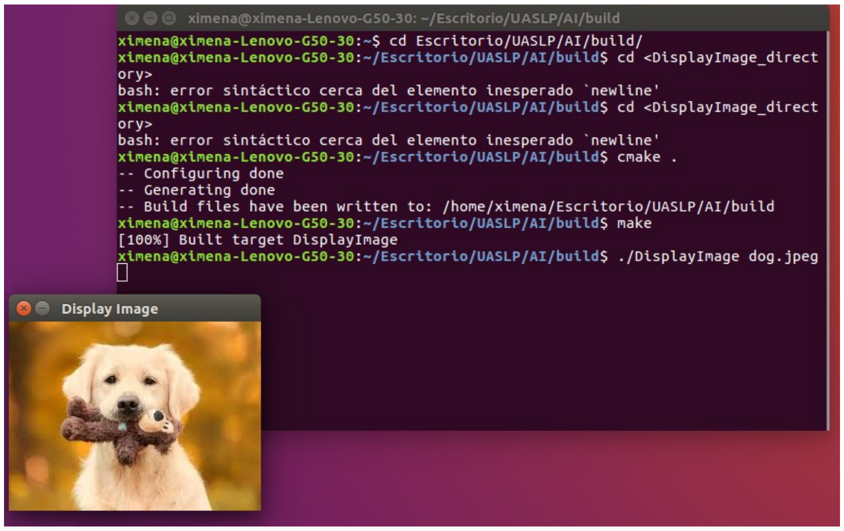
This project is going to be used on Linux operative system, because Ubuntu show to us what's going on with our instructions and it's more confortable to the programmer. This version of Ubuntu is **16.04**.
First, we have to install all the different packages: python – last version (which is included by default), the library of OpenCv, OpenCv package for python, OpenCv documentation, Cmake, pkg-config, build-essential and different package for python.
Before learn python, I stared to make a C**++** program, just for make some tests if it's running OpenCv and Cmake.

```
#include <cv.h>
#include <highgui.h>
using namespace cv;

int main( int argc, char** argv ) {
  Mat image;
  image = imread( argv[1], 1 );
  if( argc != 2 || !image.data ) {
     printf( "No image data \n" );
     return -1;
}

namedWindow( "Display Image", CV_WINDOW_AUTOSIZE );
imshow( "Display Image", image );
waitKey(0);
return 0;
}
```

Next the CmakeLists.txt have to be created for the following text:

```
project( DisplayImage )

find_package( OpenCV REQUIRED )
add_executable( DisplayImage DisplayImage )
target_link_libraries( DisplayImage ${OpenCV_LIBS} )
```

This is for the DisplayImage excecution, Cmake is looking for this file always. Just generate the executable and try to run it. The result is:



```
ximena@ximena-Lenovo-G50-30:~$ cd Escritorio/UASLP/AI/build/
ximena@ximena-Lenovo-G50-30:~/Escritorio/UASLP/AI/build$ cd <DisplayImage_direct
ory>
bash: error sintáctico cerca del elemento inesperado `newline'
ximena@ximena-Lenovo-G50-30:~/Escritorio/UASLP/AI/build$ cd <DisplayImage_direct
ory>
bash: error sintáctico cerca del elemento inesperado `newline'
ximena@ximena-Lenovo-G50-30:~/Escritorio/UASLP/AI/build$ cmake .
-- Configuring done
-- Generating done
-- Build files have been written to: /home/ximena/Escritorio/UASLP/AI/build
ximena@ximena-Lenovo-G50-30:~/Escritorio/UASLP/AI/build$ make
[100%] Built target DisplayImage
ximena@ximena-Lenovo-G50-30:~/Escritorio/UASLP/AI/build$ ./DisplayImage dog.jpeg
```

Schedule

Partial 2:

- Start implementing image recognition
- Use SURF, SIFT and FAST algorithms
- Compare those methods by different tests

Partial 3:

- Start trainning the program with real images of monuments and places on San Luis Potosí
- Know which pointsof the images the program takes
- Make a simply interface

Partial 4:

- Keep trainning the program but with 5 different places
- Try to put a different image which will not be at the data base.
- Final result

References

1. Beyeler, M. (2017). *Machine Learning for OpenCV*. Packt Publishing Ltd.
2. Cornelius T. L.. (1998). Image processing and pattern recognition. San Diego, London: Academic Press.
3. Howse, J., Joshi, P., & Beyeler, M. (2016). *Opencv: computer vision projects with python*. Packt Publishing Ltd.
4. Howse, J., Puttemans, S., Hua, Q., & Sinha, U. (2015). *OpenCV 3 Blueprints*. PACKT Publishing Ltd.
5. Laganière, R. (2014). *OpenCV Computer Vision Application Programming Cookbook Second Edition*. Packt Publishing Ltd.
6. Lutz, M. (2010). *Programming Python: powerful object-oriented programming*. " O'Reilly Media, Inc.".
7. OpenCV. (2019, july 26). Open Source Computer Vision. -, 4.1.1, -. 2019, september 10, Of OpenCv data base.
8. Tavares, J., & Jorge, R. N. (2015). *Computational Vision and Medical Image Processing V: Proceedings of the 5th Eccomas Thematic Conference on Computational Vision and Medical Image Processing (VipIMAGE 2015, Tenerife, Spain, October 19-21, 2015)*. CRC Press.
9. Theodoridis, S., & Koutroumbas, K. (2009). Pattern recognition. 2003. *Elsevier Inc*.