The Research for OpenCV (Open Sources Computer Vision) Object Detections and The System Development

Patsawee Wichittritham[1] , Issara Ruangsakorn[2] , Phasin Keeratithanetkul[3]
Information Technology, Department of Mathematic and Computer science
Rajamangala University of Technology Thanyaburi
Co-operative Education at National Pingtung University, Taiwan

Abstract

Open Source Computer Vision Library (OpenCV) is released under a BSD license and hence it's free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. OpenCV supports a lot of algorithms related to Computer Vision and Machine Learning. This report discussed about Image Recognition and Object Detection with OpenCV. An image recognition algorithm takes an image as input and outputs what the image contains, It have to train the algorithm to learn the differences between different classes for Used to identify and count objects in the image, including object detection.

1. Introduction

Open Source Computer Vision Library (OpenCV) is released under a BSD license and hence it's free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. OpenCV supports a lot of algorithms related to Computer Vision and Machine Learning. This report discussed about Image Recognition and Object Detection with OpenCV[1].

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images[2].

An image recognition algorithm takes an image as input and outputs what the image contains, It have to train the algorithm to learn the differences between different classes for Used to identify and count objects in the image, including object detection.

2. Theory

2.1 Computer Vision[3]

Computer vision is a field of computer science that works on enabling computers to see, identify and process images in the same way that human vision does, and then provide appropriate output. It is like imparting human intelligence and instincts to a computer. In reality though, it is a difficult task to enable computers to recognize images of different objects.

Computer vision is closely linked with artificial intelligence, as the computer must interpret what it sees, and then perform appropriate analysis or act accordingly.

2.1.1 Image acquisition

Image acquisition is the process of translating the analog world around us into binary data composed of zeros and ones, interpreted as digital images.

2.1.2 Image processing

Image processing is the low-level processing of images. Algorithms are applied to the binary data acquired in the first step to infer low-level information on parts of the image. This type of information is characterized by image edges, point features or segments, for example. They are all the basic geometric elements that build objects in images.

2.1.3 Image analysis and understanding

The last step of the Computer Vision pipeline if the actual analysis of the data, which will allow the decision making. High-level algorithms are applied, using both the image data and the low-level information computed in previous steps.

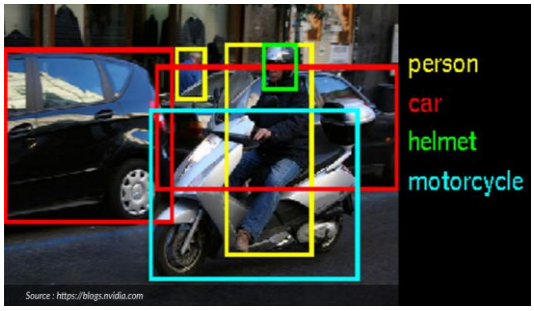

Figure 1 Recognition of objects in images

2.2 Basic Haar like feature

One such method would be the detection of objects from images using features or specific structures of the object in question.However, there was a problem.Working with only image intensities,meaning the RGB pixel values in every
single pixel in the image, made feature calculation rather computationally expensive and therefore slow on most platforms.

This problem was addressed by the socalled Haar-like features, developed by Viola and Jones on the basis of the proposal by Papageorgiou et. al in 1998.A Haar-like feature considers neighbouring rectangular regions at a specific location in a detection window,sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image.

An example of this would be the detection of human faces. Commonly,the areas around the eyes are darker than the areas on the cheeks. One example of a Haar-like feature for face detection is therefore a set of two neighbouring rectangular areas above the eye and cheek regions.

$$f(x,y) = \sum_i p_b(i) - \sum_i p_w(i)$$

Figure 2 Basic Haar like feature types

2.3 Open Source Computer Vision Library (OpenCV)

OpenCV is an open source computer vision library. The library is written in C and C++and runs under Linux, Windows and provides interfaces for Python, Ruby, Matlab and other languages. OpenCV library contains abundant advanced math functions, image processing functions, and computer vision functions that span many areas in vision.

2.4 Cascade classifier

The cascade classifier consists of a list of stages, where each stage consists of a list of weak learners. The system detects objects in question by moving a window over the image. Each stage of the classifier labels the specific region defined by the current location of the window as either positive or negative – positive meaning that an object was found or negative means that the specified object was not found in the image. If the labelling yields a negative result, then the classification of this specific region is hereby complete and the location of the window is moved to the next location. If the labelling gives a positive result, then the region moves of to the next stage of classification. The classifier yields a final verdict of positive, when all the stages, including the last one, yield a result, saying that the object is found in the image. A true positive means that the object in question is indeed in the image and the classifier labels it as such – a positive result.

A false positive means that the labelling process falsely determines, that the object is located in the image, although it is not. A false negative occurs when the classifier is unable to detect the actual object from the image and a true negative means that a nonobject was correctly classifier as not being the object in question. In order to work well, each stage of the cascade must have a low false negative rate, because if the actual object is classified as a non-object, then the classification of that branch stops, with no way to correct the mistake made. However, each stage can have a relatively high false positive rate, because even if the n-th stage classifies the non-object as actually being the object, then this mistake can be fixed in n+1-th and subsequent stages of the classifier.

3. Methodology

3.1 Create Classified

3.1.1 Fast computation of Haar-like features

This is a divide of image space with the four types of images that they have defined from Haar-like features as shown (A - B - C - D). Calculate the color table of the image by integrating the color table in the white space minus the color table in the black space. In the synthesis process, the results are used in Adaptive Boost Learning Algorithm for Adaboost. because of the amount of data available, Adaboost is a good solution.

3.1.2 Positive image

Positive image is the object, we want to detect we use positive 120 images in grayscale 400x400 pixel

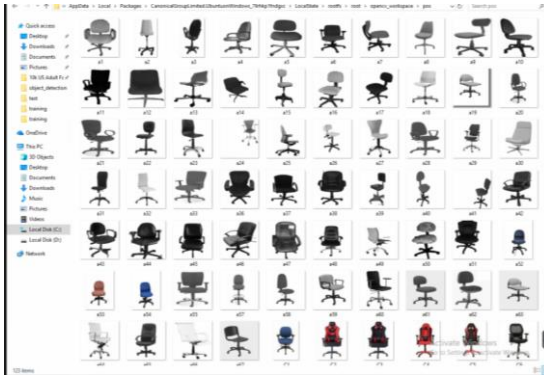

Figure Positive image

3.1.3 **Negative image**

Negative image it is the image we don't use to detect in this project we use the image of room and inside the image it don't have any office's chair In this project we use 1200 negative image with grayscale 400 multiply 400 pixel

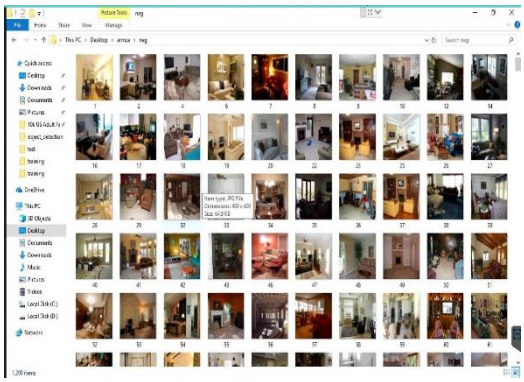

Figure Negative Image

3.1.3 **Create Vec file**

We created a vector file (vec) from all of the images we have using the opencv_createsamples provided in the opencv after we installed it.

For the program opencv_createsamples creates a vector file for an image that is not related to what we are interested in. And we will have one file vec extension and keep the vec in the folder we have set.When we have the vec file, we will have to merge all the vec files into the same file.

We only need one cascade fil

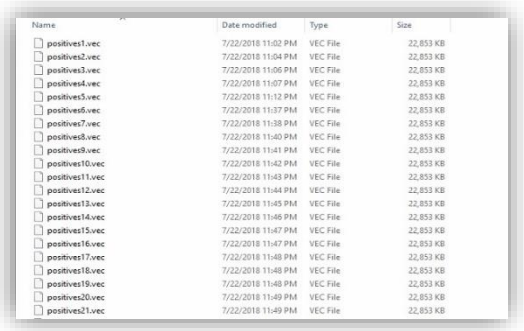

Figure Vec file

3.1.4 **Training work**

We can look at the operation report at the command ubuntu cascade.xml in the folder we work. The data information in the ubuntu command will tell you the following parameters:

HR = hit rate is the ratio of the number of samples to be extracted correctly.

FA = False Alarm is the ratio of error numbers.

Which affects performance and the accuracy of the cascade file we will have in each stage.

```
===== TRAINING 24-stage =====
<BEGIN
POS count : consumed   100
NEG count : consumed    50
Precalculation time: 100
+----+---------+---------+
|  N |    HR   |    FA   |
+----+---------+---------+
|   1|        1|        1|
+----+---------+---------+
|   2|        1|        1|
+----+---------+---------+
|   3|        1|        1|
+----+---------+---------+
|   4|        1|        1|
+----+---------+---------+
|   5|        1|    0.961|
+----+---------+---------+
|   6|   0.9995|   0.8155|
+----+---------+---------+
|   7|   0.9995|   0.8175|
+----+---------+---------+
|   8|   0.9995|   0.7465|
+----+---------+---------+
|   9|   0.9995|    0.549|
+----+---------+---------+
|  10|   0.9995|   0.6155|
+----+---------+---------+
|  11|   0.9995|    0.444|
+----+---------+---------+
END>
Training until now has taken 1 days 12 hours 39 minutes 10 seconds.
```

Figure training work stage

3.2 The function for capture the image

When the object change position our program can capture the image and we set the rule of range x and y is between plus - minus 5

```
if ( ((Xtemp[n][0] >= Xtemp[n][1] - 5) and (Xtemp[n][0] <= Xtemp[n][1] + 5))
     and ( (Ytemp[n][0] >= Ytemp[n][1] - 5) and (Ytemp[n][0] <= Ytemp[n][1] +
                show = 'Not Change'
                statuss = 'Ready'

            else:
                cv2.putText(img,"Change" + str(show), (0, img.shape[0] - 40),
                            cv2.FONT_HERSHEY_TRIPLEX, 0.5,(255, 255, 255), 1)

                cv2.imwrite('Capture Image ' + str(i[0]) + '.jpg', img)
                print ('Capture Now : Capture Image '+ str(i))
                print(i[0])
                i[0] +=1
                show = 'Change'
```

Figure Function capture image

3.3 GUI

This is the gui we make by Qt design

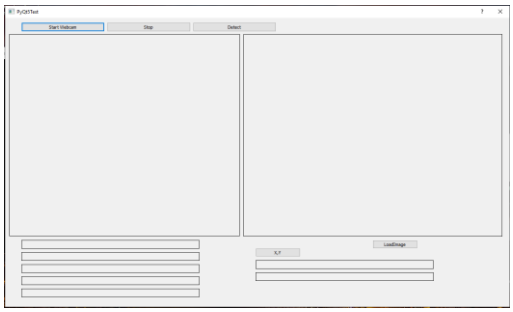

Figure GUI by Qt design

4. Result

Program can detect and capture the image when the object changes the position

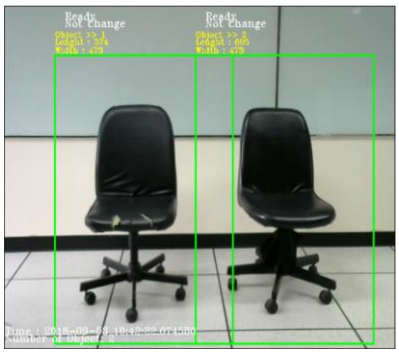

Figure Program can detect object

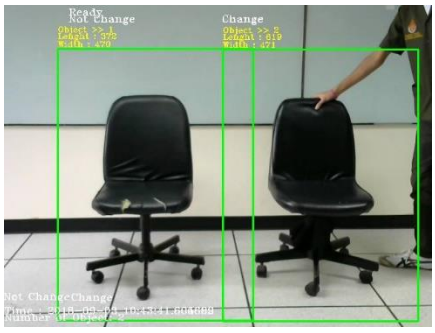

Figure Capture image

5. Acknowledgement

We would like to thank Professor Chu-Fu Wang for assistance, knowledge and cooperation, We are also grateful to Dr.Nipat Jongsawat for the opportunity and advice. In addition to working instructions, there are also tips for living in Taiwan.

References

[1] Adrien Kaiser, “What is Computer Vision?” [Online]. Available: https://hayo.io/computer-vision/. [Accessed Jan 2017].

[2] Nidhi, “Introduction to OpenCV,” Image Processing and Object Detection, 2015.

[3] Opencv.org, “About OpenCV,” [Online]. Available: https://opencv.org/. [Accessed 2017].

[4] OpenCV-Python Tutorials, “Face Detection using Haar Cascades,” [Online]. Available:https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html#face-detection. [Accessed 2017].

[5] T. M. Inc., “Train a Cascade Object Detector,” [Online]. Available: http://www.mathworks.se/help/vision/ug/train-a-cascadeobject-detector.html#btugex8. [Accessed Nov 2014].

[6] V. Jones, “Rapid object detection using a boosted cascade of simple features,”

Computer Vision and Pattern Recognition, 2001.

[1] R. Issara Information technology
Rajamangala University of Technology Thanyaburi
46/232 Donmung,Donmung Bangkok
Tel 084-7764622
Wave-issara@hotmail.com

[2] K. Phasin Information Technology
Rajamangala University of Technology Thanyaburi
32/228-229 Klong Luang Klong one Pathumtani
Tel 082-4237415
Phasin98@gmail.com

[3] W. Patsawee Information Technology
Rajamangala University of Technology Thanyaburi
41/151 Nawamin Bungkum Bangkok
Tel 094-5566689
Patsawee.wi@gmail.com