

REAL TIME OBJECT DETECTION USING MACHINE LEARNING AND OPENCV

AbhishekTyagi

Department of Computer Science and Engineering,

IMS Engineering College, Ghaziabad, Uttar Pradesh, India

ABSTRACT

Object detection is a vital feature of applied science. the advantages of object detection is but not restricted to somebody with a academic degree of science. Instead, object detection is growing deeper and deeper into the common components of the knowledge society, disposal a aid where required. This paper can address one such risk, particularly the assistance of a Haar-cascade classifier. the most focus are going to be on the case study of a face detection and object detection like watch detection, pen detection. The goal of the system to be developed is to additional ease and augment the everyday a part of our lives

Keywords- Object Detection, OpenCV, Python, Matlab.

INTRODUCTION

Object detection is often spoken as a technique that's answerable for discovering and distinctive the existence of objects of an explicit category. AN extension of this will be thought-about as a technique of videoprocessing to spot objects from live videos.

1.1 HAAR CHARACTERISTICS

One such methodology would be the detection of objects from pictures mistreatment options or specific structures of the article in question. However, there was a haul. operating with solely video intensities, which means the RGB picture element values in each single picture element within the image, created feature calculation rather computationally dear and thus slow on most platforms. This drawback was addressed by thus referred to as Haar-like

options, developed by Viola and Jones on the idea of the proposal by dada Georgiou ET. Al in 1998. A Haar-like feature considers neighboring rectangular regions at a selected location during a detection window, sums up the picture element intensities in every region and calculates the distinction between these sums. This distinction is then wont to categorise subsections of a video.

1.2 CASCADE CLASSIFIER

The cascade classifier consists of an inventory of stages, wherever every stage consists of an inventory of weak learners. The object is detected in question by the system by moving the window over the image. Every stage of the classifier labels the particular region outlined by the present location of the window as either positive or negative – positive that means that associate object was found or negative implies that the required object wasn't found within the image. If the labelling yields a negative result, then the classification of this specific region is herewith complete and also the location of the window is captive to subsequent location. If the labelling provides a positive result, then the region moves of to subsequent stage of classification. The classifier yields a final finding of fact of positive, once all the stages, as well as the last one, yield a result, oral communication that the thing is found within the image.

A true positive implies that the thing in question is so within the video and also the classifier labels it per se – a positive result. A false positive implies that the labelling method incorrectly determines, that the thing is found

within the video, though it's not. A false negative happens once the classifier is unable to sight the particular object from the video and a real negative implies that a non-object was properly classifier as not being the thing in question. so as to figure well, every stage of the cascade should have a coffee false negative rate, as a result of if the particular object is assessed as a non-object, then the classification of that branch stops, with no thanks to correct the error created. However, every stage will have a comparatively high false positive rate, as a result of although the n-th stage classifies the non-object as truly being the thing, then this error are often mounted in n+1-th and resultant stages of the classifier..

2. LITERATURE SURVEY

2.1 OPENCV

OpenCV(Open provide laptop vision) could also be a library of programming functions within the main aimed toward quantity laptop vision. Originally developed by Intel, it had been later supported by Willow Garage then Itseez (which was later noninheritable by Intel). The library is cross-platform and free to be used below the American Standard Code for Information Interchange file BSD license.

OpenCV lib assist deep learning frameworks TensorFlow, Torch/PyTorch and Caffe. OpenCV is written in C++ and its primary interface is in C++, however it still retains a less comprehensive though thoroughgoing older C interface. There unit of mensuration bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces ar progressing to be found at intervals the on-line documentation. Wrappers in many languages like C#, Perl, Ch, Haskell and Ruby unit of measurement developed to encourage adoption by a wider audience. Since version three.4, OpenCV.js could also be a JavaScript binding for chosen set of OpenCV functions for internet platform. All of the new developments and algorithms in

OpenCV unit of mensuration presently developed at intervals the C++ interface

OpenCV's application areas include:

- Facial recognition system
- Gesture recognition
- Human-computer interaction (HCI)
- Mobile AI
- Motion understanding
- Object identificati
- Segmentation and recognition

2.2 MACHINE LEARNING

Machine learning (ML) is that the scientific study of algorithms and applied math models that laptop systems use to effectively perform a selected task while not exploitation specific directions, wishing on patterns and abstract thought instead. it's seen as a set of computer science. Machine learning algorithms build a mathematical model of sample information, called "training data", so as to form predictions or selections while not being expressly programmed to perform the task. Machine learning algorithms ar utilized in a large kind of applications, like email filtering, and laptop vision, wherever it's impossible to develop AN formula of specific directions for playing the task. Machine learning is closely associated with machine statistics, that focuses on creating predictions exploitation computers. The study of mathematical improvement delivers ways, theory and application domains to the sphere of machine learning. data processing could be a field of study at intervals machine learning, and focuses on beta information analysis through unsupervised learning. In its application across business issues, machine learning is additionally cited as prophetic analytics.

3. IMPLEMENTATION

3.1 FEATURE EXTRACTION

Haar Cascades use machine learning techniques in which a function is trained from a lot of positive and negative images. Feature Extraction is the process used in this algorithm. Training data that is used in this project is an XML file called: haarcascade_frontalface_default.xml

4. SAMPLE DATA

Here we are using some images and some real time videos for the object identification and detection. Given below are some examples how our system will detect the objects in real time. It is very useful in video surveillance and object identification.

For object detection we first train our classifier using a lot of images, here we are going to detect a human face

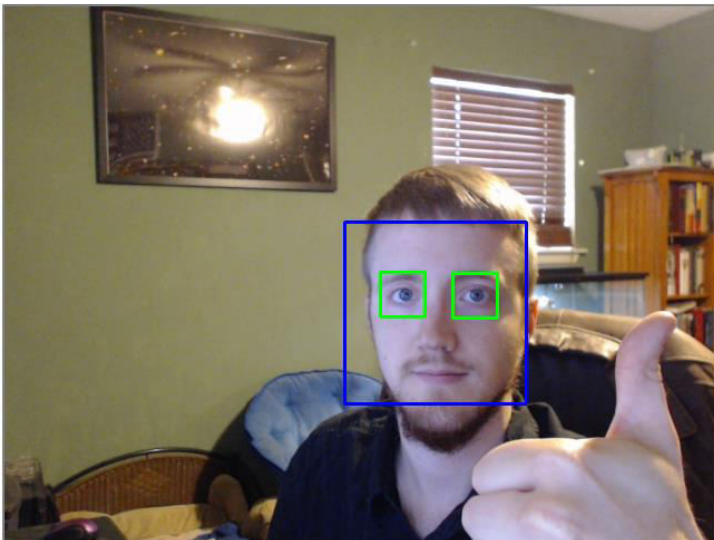


Fig-1.1

In the above samples we first train our classifiers using face images and non-face images and then extract the features. This method uses concept of cascade classifier, that is, instead of applying all the features at the same time we group the features into different stages and these features are applied at different stages. In this we discard the video if it fails in the first stage, else continue to the next stage. When the window passes through every stage we get our desired output as shown in fig-1.1.

5. CONCLUSION

The main goal of the present work has been to introduce the concepts and techniques of computer vision and object/human tracking. We presented some basic concepts of Computer Vision and defined a tracking problem as a framework. We explored the theories of current solutions in visual object tracking. We demonstrated some of the fundamental techniques implemented in Python OpenCV and MATLAB that can be used in human detection and tracking in video. The advantages of OpenCV make it a powerful open source tool at the reach of any user. However, this tool demands considerable programming efforts, even from the first steps when installation of the program is required (sometimes it is thought that the most complicated in OpenCV is the installation process), are also obligatory for a better understanding of the implemented methods in OpenCV. Considering the approaches obtained in the replications and the versatility of the language, these

techniques can be used to develop applications for online video surveillance in processor boards. Future work will consider them as a base point to applications over mentioned boards. We also consider as future research the use of machine learning algorithms in the implementation of surveillance systems

6. REFERENCES

- [1] Ali, A., Jalil, A., Niu, J., Zhao, X., Rathore, S., Ahmed, J., & Iftikhar, M. A. (2016). Visual object tracking—classical and contemporary approaches. *Frontiers of Computer Science*, 10(1), 167-188.
- [2] Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on signal processing*, 50(2), 174-188.
- [3] Danelljan, M., Hager, G., Shahbaz Khan, F., & Felsberg, M. (2015). Convolutional features for correlation filter based visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision Workshops* (pp. 58-66).
- [4] Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3-4), 197-387.
- [5] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning (Vol. 1)*. Cambridge: MIT Press.
- [6] Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media
- [7] Yang, M., Wu, Y., & Hua, G. (2009). Context-aware visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7), 1195-1209.