

**HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**SCHOL OF INFORMATION TECHNOLOGY AND COMMUNICATION**

**-----🙞🙜🕮🙞🙜-----**



**OBJECT DETECTION**

**USING SLIDING WIDOWS REPORT**

**Project name**: Object detection with sliding windows

**Supervisor:**

**Student names:** Group

|  |  |  |
| --- | --- | --- |
| **Trần Trung Hiếu** | **20194764** |  |
| **Tô Xuân Hùng** | **20194772** |  |
| **Ngô Quốc Thắng** | **20194839** |  |
| **Trần Đức Thắng** | **20194842** |  |
| **Phan Nguyên Anh** | **20194727** |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***Hà Nôi, 2021***

***Hà Nội, tháng 10 năm 2012***

Computer Vision project name:

Object detection using sliding windows

Student names: Tran Trung Hieu, To Xuan Hung, Ngo Quoc Thang,

Tran Duc Thang, and Phan Nguyen Anh

Class ICT Global K64, Hanoi University of Science and Technology, No1. Dai Co Viet st., Hanoi, Vietnam

**Abstract.** Much of the focus in the object detection literature has been on the problem of identifying the bounding box of a particular class of object in an image. Yet, in contexts such as robotics and augmented reality, it is often necessary to find a specific object instance—a unique toy or a custom industrial part for example—rather than a generic object class. Here, applications can require a rapid shift from one object instance to another, thus requiring fast turnaround which affords little-to-no training time. Moreover, gathering a dataset and training a model for every new object instance to be detected can be expensive and time-consuming. In this context, we propose a generic 2D object instance detection approach that uses example viewpoints of the target object at test time to retrieve its 2D location in RGB images, without requiring any additional training (i.e. fine-tuning) step. To this end, we present an end-to-end architecture that detects the specific object from its viewpoint. The sliding windows technique is used to sequentially scan the sub-areas of the image and the image-pyramid technique resizes the image solution to scan which will cover the object even if its size is larger than the sliding window. For the training model, the project uses yolov5 library.

**Keywords:** object detection, sliding windows, yolov5, train model, image pyramid

1. Introduction

Our project aimed to design and implement an object detection application using sliding windows algorithm. This is one of the earliest solutions for computer vision and object identification problems. Since the advent of computers, people have always what to enhance the ability of machines on helping and doing more abstract and complicated tasks.

Machines only understand 0 and 1 so how can we make them do more abstract tasks such as identifying objects? This is the problem that we have been solving for a long time. People have been thinking of different algorithms for identifying objects. The algorithm improvement is from using sliding windows algorithm to RCNN to fast-CNN to faster-CNN and to this day Yolo

Our goal was to create a program that implements the sliding windows algorithm with some improvement using the image-pyramid algorithm and for the training model, we use yolov5 library.

1. Problem Definition

The goal was to create a simple application that can identify objects, in this case, we use the application for detecting motorcycles using the sliding windows algorithm. The accuracy of the application should be in the acceptance range. Before running the application, we should train a model based on a data set. After training a model for the application, the input to the core of the program is an image, and the output is a set of cropped images that contain the wanted object.

**1.** The model should be trained using yolov5 library with an appropriate dataset that is divided into 3 sets: train set, valid set, and test set.

**2.** When using the application with an input image, the output should be a window that shows the image that is being identified with a window that slides through the image from left to right, from top to bottom. When the windows slide through an area that has the object class (motorcycle) then the sliding window will turn blue and save the cropped image of that window into a folder “runs”.

**3**. The algorithm will have a supported algorithm for enhancing the accuracy of the application. Which is the image-pyramid algorithm. Basically, the algorithm will resize the image resolution (reduce the image resolution by a factor such as 1.5 after each step). After each time the input image is scanned then the image will be resized and scanned again until the image is resized to a minimum thresh-hold. This algorithm will help the sliding window algorithm to detect objects that is larger than the sliding window itself.

1. Algorithm
   1. Sliding windows algorithm

In the context of computer vision (and as the name suggests), a sliding window is a rectangular region of fixed width and height that “slides” across an image, such as in the following figure:



Figure 1: **Click the image to run video**

There are 3 main attributes of sliding windows:

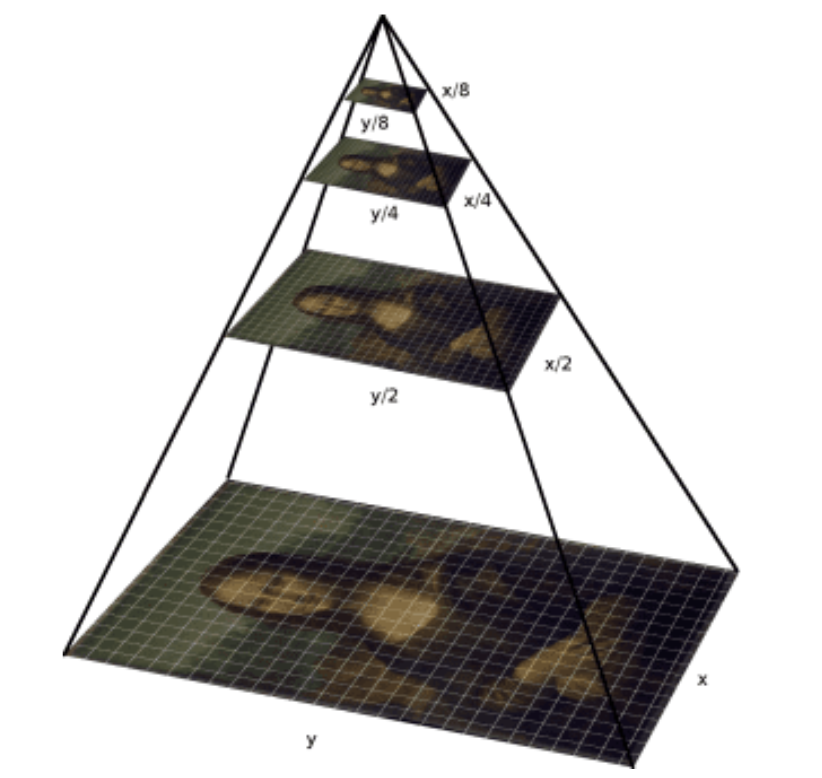
* Windows height and width
* Windows step size (How many pixels each time the window slides)
* Windows moving speed (How long is the delay between each slide)
  1. Image pyramid

An “image pyramid” is a multi-scale representation of an image.

Utilizing an image pyramid allows us to find objects in images at different scales of an image. And when combined with a sliding window we can find objects in images in various locations.

At the bottom of the pyramid, we have the original image at its original size (in terms of width and height). And at each subsequent layer, the image is resized (subsampled) and optionally smoothed (usually via Gaussian blurring).

The image is progressively subsampled until some stopping criterion is met, which is normally a minimum size has been reached and no further subsampling needs to take place.



* 1. YOLOv5 open source for training CV model

YOLO means “You Only Look Once” object detection and image segmentation model developed by Ultralytics. The YOLOv5 model is designed to be fast, accurate, and easy to use, making it an excellent choice for a wide range of object detection and image segmentation tasks. It can be trained on large datasets and can run on various hardware platforms, from CPUs to GPUs.

In this case, the project will use YOLOv5 to train a custom model from a labeled dataset: [https://universe.roboflow.com/hva/schoolllll/dataset/1#](https://universe.roboflow.com/hva/schoolllll/dataset/1) . The data set contains a train set, a validate set, and a test set.

1. Experimental Results and Evaluation
   1. Experimental Results

Graphical user interface, chart, line chart

Description automatically generated

Figure 2: result

Chart, treemap chart

Description automatically generated

Figure 3: Confusion matric

1. Advantages

The sliding windows algorithm is a powerful and useful technique for detecting objects in images or videos. It works by dividing the image into multiple smaller windows and running the detection algorithm on each one. This allows for a more efficient search of the image and can be used to detect objects of various sizes and shapes. Furthermore, this algorithm can be used to detect objects in real time by using a stream of video frames.

The main advantage of the sliding windows algorithm is its accuracy and efficiency. By running the detection algorithm on each window, the system can accurately detect objects of various sizes and shapes. Additionally, it can also detect objects in real time, which is a great advantage for applications such as surveillance and autonomous driving.

1. Limitation

However, there are some drawbacks associated with this algorithm. Firstly, it is computationally expensive, as it requires multiple passes over the image. Additionally, the algorithm can be vulnerable to false positives and false negatives, as the detection algorithm may not always be able to accurately detect objects. Therefore, it is important to consider these drawbacks when using the algorithm, in order to improve its accuracy and efficiency.

Also, it has a critical limitation. Which is, by scanning multiple areas and resized images of the original image, an object may be detected multiple times. It will be hard to point out the correct area of the detected object since there are too many times the object is detected.

1. Solution for the limitation

Shape

Description automatically generatedThe solution for the limitation is by using an algorithm that will store each detected object's location by the x-axis, y-axis, and its width and height (basically is a rectangle boundary of the object location). After that, it will analyze the list of those location data for getting the list of objects in the image. For an easier illustration of how the algorithm works:

Because the sliding windows algorithm will scan the image multiple times then there will be many rectangle boundaries of an object so we need to find the best one that will suit that object. In the image example above, each rectangle is a boundary of a detected object in the process of scanning an image using sliding windows. The blue rectangle boundary should be considered as a specific object while the black boundary rectangle would be eliminated.

The idea for the process of this algorithm is like so:

1. If inside the boundary there is more than 1 boundary that has the same level, then it’s not a boundary identifier for an object
2. Else it’s an object identify boundary indicator

However, we haven’t successfully implemented this algorithm in our application. Hence, this would be our next update for the application in future.

1. Conclusion

Many improvements could be made to the implementation in its current form. They can come either from using more algorithms or improving the existing model.

The sliding algorithm can be considered as falling behind compared to newer algorithms (RCNN, fast and faster RCNN, or YOLO). However, sliding windows is still one of the earliest algorithms for solving Computer Vision problems.

In conclusion, the sliding windows algorithm is an effective and efficient method for detecting objects in images and videos. It is designed to recognize objects of different sizes and shapes, as well as objects in real-time, and the algorithm can provide accurate results with minimal false positive or false negative detections. However, the sliding windows algorithm can be computationally expensive, so it is important to take this into consideration when implementing the algorithm. Furthermore, it is essential to consider the potential for false positive and false negative detections when using the algorithm, as this can significantly affect the accuracy and efficiency of the detection results. By being aware of all of these factors, it is possible to use the sliding windows algorithm to effectively and accurately detect objects in images and videos.

References

1. Sliding windows: <https://pyimagesearch.com/2015/03/23/sliding-windows-for-object-detection-with-python-and-opencv/>
2. Image pyramid: <https://pyimagesearch.com/2015/03/16/image-pyramids-with-python-and-opencv/>
3. YOLOv5: https://github.com/ultralytics/yolov5