|  |  |
| --- | --- |
|  |  |

**Object Detection in Automated Software Testing**

|  |  |
| --- | --- |
| Author  Intern  Extern  Company | Brent Gerets  Bart Stukken  David Vandingenen  Brightest |

Abstract

This application note presents an application to facilitate the entire process of setting up an object detection model to test the presence of certain elements on specific web pages. The conventional method to test the presence of elements on a page in automated software testing involves parsing the page’s HTML. An element being present in the HTML is no guarantee that it is indeed visible to the user, however. In critical situations, it could be important to be sure that a certain element is visible. Because of the increased interest in machine learning over the past few years, object detection models will be used to verify whether an element is visible or not. A comparison of popular and state-of-the-art object detection models is made considering the large number available, to decide which to focus on. Before a model can be used it must be trained on a custom dataset specifically created for a website, since specific elements must be detected. The presented application allows for the creation of custom datasets in a streamlined manner. These can then be used to train the models (YOLOv9 and RT-DETR) inside the application. Furthermore, the models can be tested after training to verify their performance. The application considers the limited knowledge of machine learning most user will have, by simplifying the required inputs to the bare necessities and providing a convenient user experience. The goal of creating a proof-of-concept involving object detection of web elements has been completed. However, it remains open to discussion and research whether machine learning is the best solution to the problem. The application provides a starting point for and encourages the continued research into the use of machine learning in automated software testing. In addition, it serves as a tangible way to determine the effectiveness of object detection in the field of automated software testing.

Conclusion

In this application note, an application to facilitate the entire process of setting up an object detection model to test the presence of certain elements on specific web pages has been presented. This process includes the creation of a dataset, and training and testing of the selected models. A choice of two models is currently available, namely YOLOv9 and RT-DETR. These models where selected for use in the application after making a weighted scoring model to compare and rank several popular and state-of-the-art object detection models. The application allows a user to generate an accurate object detection model in a few simple, streamlined steps, without the need for any machine learning knowledge. A positive user experience is supported by the many quality-of-life features and a user interface that conforms with modern design standards. Thus, the goal of developing a proof-of-concept for an application that facilitates the creation of object detection models to detect elements on web pages has certainly been reached. The only question that remains then is whether machine learning, and specifically object detection, is the right solution for testing the presence of web elements. The entire purpose of an object detection model is to generalize over its training set to be able to detect objects in unseen data. This contradicts somewhat with the requirement of detecting specific web elements. In a way, overfitting is necessary in this case since generalization is not the goal (to a certain extent, slight variations should still be detected). This has also been made clear by the fact that many data augmentation methods, used to prevent overfitting, cannot be applied in this case. It is difficult to answer this question without the application being used in the field by testers, though it seems like exploring options other than object detection would be beneficial.