TECHNISCHE UNIVERSITÄT
ILMENAU

DEPARTMENT OF AUTOMATION ENGINEERING
FACULTY OF COMPUTER ENGINEERING AND AUTOMATION

Co-op (Internship) at Bosch eBike Systems

Machine Learning-Based Solution for Test Automation in Visual Analysis

Start Date:	April 1st, 2024
End Date:	August 30th, 2024
Academic Supervisor:	Prof. Dr. Yuri Shardt
Industry Supervisor:	Mr. Mohith Shivakumar
	Mohith.Shivakumar@de.bosch.com
Submitted by:	Uday Kaipa
	Matriculation: 65172
	Date of Birth: October 24th, 1997
	uday-kumar-reddy.kaipa@tu-ilmenau.de
Course of Studies:	Research in computer and system engineering
Submitted on:	November 15th, 2024

Acknowledgments

I would like to express my deepest gratitude to my industry supervisor, Mr. Mohith Shivakumar, and my academic supervisor, Prof. Dr. Yuri Shardt, for their unwavering support throughout the course of this industry cooperation. Their guidance, from the initial stages to the completion of the project, has been invaluable.

I especially appreciate Mr. Shivakumar's efforts in guiding me through each phase of development. His clear communication, collaborative brainstorming, and encouragement during challenging moments were instrumental in overcoming obstacles.

This co-op has been a significant milestone in my career, providing me with the rewarding opportunity to work on advanced and complex system test benches, particularly with the Bosch eBike Smart System 3 (BES3) components. I am truly grateful for the chance to contribute to such cutting-edge eBike technology.

I would also like to extend my heartfelt thanks to Michael Schänzlin, Liliane Gasse, Christoph Klack, Ingo Krüssner, and the rest of the System Testing team for their valuable insights into system testing and for making this experience not only technically enriching but also personally rewarding.

Furthermore, I would like to acknowledge the previous developers of the software used in the system test benches, as their work laid the foundation for my contributions. I am also thankful to the members of

the Google Tesseract forum and other online communities and tools that helped me navigate technical challenges more efficiently.

Lastly, I am grateful to everyone, whether mentioned here by name or not, who has supported me in various ways throughout this project. Your encouragement and assistance have been greatly appreciated, and I sincerely thank you all for your contributions.

Abstract

This report details the work conducted during my co-op at Bosch eBike Systems as part of the system testing team. The primary goal of the co-op was to develop a robust and customizable approach to automate the testing procedures for Bosch's Smart System 3 (BES3) components. My role involved analyzing the existing architecture of the automated system test benches and implementing a new, reliable solution for test automation.

During the co-op, I engaged in both manual testing and the development of an automation framework designed to execute automated test cases efficiently. Leveraging my theoretical knowledge of machine learning, I developed a supervised learning solution using object detection to enhance the accuracy and reliability of the testing process. This solution proved to be highly effective in automating tests and increasing overall test coverage.

Contents

1 Introduction

1.1 About Bosch eBike Systems

Bosch eBike Systems is a division of the Robert Bosch GmbH, a globally renowned leader in engineering and technology, and one of the most prominent employers in Germany and abroad. Bosch is recognized for its innovative solutions, and its eBike systems have become increasingly popular, particularly in Germany and across Europe, where eBikes account for around 15% of the bicycle market [Sys23b].

Bosch eBike Systems operates as an original equipment manufacturer (OEM), providing advanced components for many well-known bicycle brands, rather than producing eBikes directly for consumers [Sys23a]. Bikes equipped with Bosch components can be found throughout Europe, reflecting the brand's strong presence in the market.

The Bosch eBike Smart System (BES) has undergone multiple iterations to meet the evolving demands of the industry. Starting with BES1, which was the initial version, followed by BES2, the system has continued to evolve. The latest release, BES3, incorporates advanced features and improved connectivity [Sys23b]. Despite the release of newer systems, earlier versions such as BES2 are still widely used by customers. This creates a need for continuous customer service and support, ensuring that components across all generations maintain high-quality standards.

Given the complexity of the software embedded in these components, which is regularly updated with new features, it is essential to maintain rigorous testing processes. Manual testing alone is time-consuming and inefficient for keeping pace with frequent updates. Therefore, automating the testing procedures is crucial for ensuring that

Bosch eBike components meet market demands while maintaining reliability and quality.

1.2 BES3 Components

The BES3 consists of a wide range of components, each with multiple variants designed to meet specific needs. These components work together to enhance the performance, security, and connectivity of Bosch-powered eBikes. The key components of BES3 include:

1. **Bosch Drive Unit (BDU):**
   The drive unit is the core of the eBike system, powering the bike and ensuring efficient performance. BES3 introduces new variants like the Performance Line CX [Fig 1.1], along with specialized units for cargo bikes and racing. The drive unit also communicates device IDs to verify the use of authorized components. These drive units primarily for pedelec (pedel assistance) and speed-pledec (up to 45km/h) bikes.

Figure 1.1 – Bosch Drive Unit - BDU384Y [Sys24]

2. **Bosch Battery Pack (BBP):**
   Bosch battery packs come in various power capacities and shapes, such as frame-mounted batteries, rack-mounted batteries, integrated batteries, and the DualBattery system. Additionally, the Range Extender provides extra capacity.

For example, the PowerPack 400 [Fig 1.2] is available in two variants: a frame-mounted battery, which can be attached to the bike frame and removed using a lock-and-key mechanism, and the PowerPack 400 Compact Tube, which is integrated into the bike's frame tube. A rack-mounted variant is also available, which attaches to the rear rack.

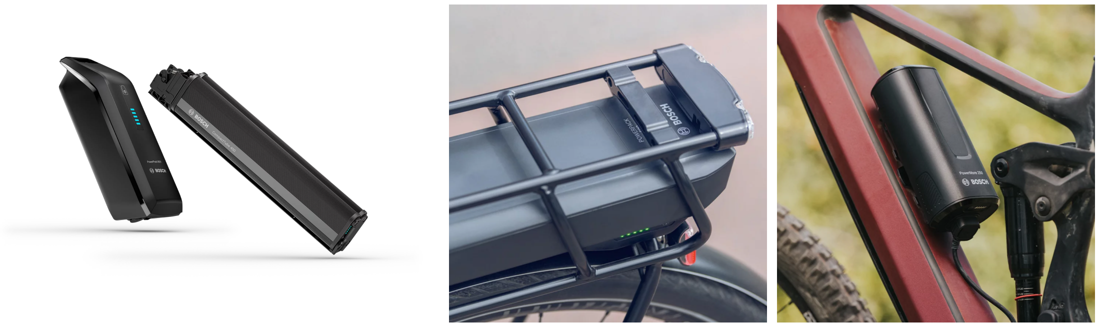

Figure 1.2 – PowerPack 400 frame and compact tube (BBP), Battery Rack and Range extender(PowerMore 250) [Sys24]

3. **Display and Control Units:**

   - **Bosch Head Unit (BHU):** A retrofittable display that serves as the key to unlock the bike and provides essential ride information to the user. The BHU displays crucial information to the rider. For example, BHU 36 [Fig 1.3], which can display navigation information.

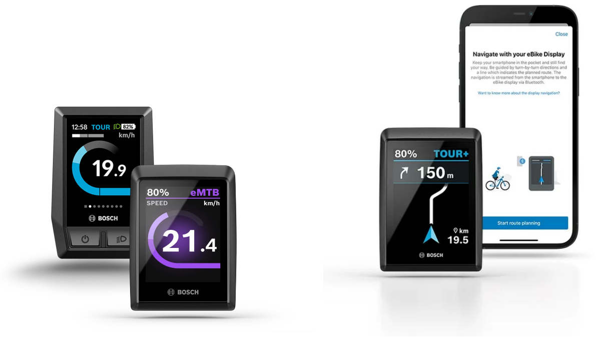

Figure 1.3 – BHUs (BHU36) and BHU with Navigation Screen [Sys24]

- **Bosch Remote Control Unit:** This unit allows the rider to navigate the BHU and adjust assistance modes. As the communication hub of the system, it can be considered the brain of the eBike. The BRC31 and BRC36 include an LED bar and assist mode LEDs that indicate battery charge and active assistance modes using color codes. The BRC38 features a display that provides information; therefore, a bike with the BRC38 does not require a BHU, but it remains compatible.

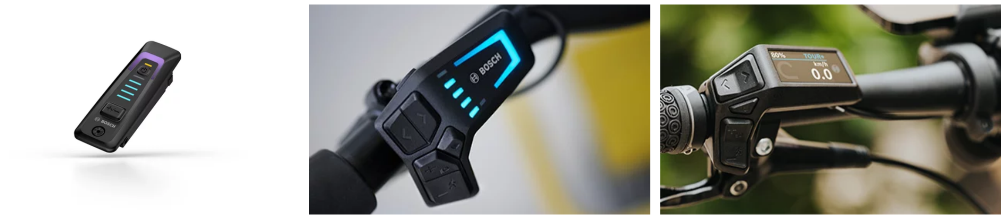

Figure 1.4 – Control Units (BRC31, BRC36 and BRC38) [Sys24]

4. **Bosch Anti-lock Braking System (BAS):**
   Similar to ABS in cars, the BAS ensures a safer braking experience by preventing wheel lock-ups during sudden stops.

5. **eBike Flow App:**
   A mobile app that enhances connectivity and can function as an additional display, allowing users to view ride data and control certain eBike features.

6. **Bosch Connect Module (BCM):**
   The BCM is an independent device connected to the drive unit, featuring a built-in SIM module. It provides additional security features, such as location tracking to prevent theft.

7. **Bosch Power Charger (BPC):**
   Used for charging the battery, BPC is available in various power rating options, supporting fast-charging capabilities to suit different needs.

1.3 Need for visual validation in BHU and BRC Testing

Certain Bosch eBike components, such as the BHU and BRC, present visual output, which makes automated verification more complex. In contrast, components like the BCM, BBP, and BDU provide electrical feedback, which can be easily measured for verification purposes.

The BHU is regarded as particularly important, as it is used to display critical alerts and information to the user, such as error messages, warnings, and the status of key systems like BAS. For this reason, a visual approach has been necessary to analyze the information presented on the display, ensuring that user alerts and other important data are accurately captured.

A typical test bench setup is configured to include multiple interconnected bike components, allowing software to be tested on various models and variants. For example, different BHU models such as BHU35 and BHU37 have been tested simultaneously to ensure compatibility and functionality across different variants.

During this co-op, a supervised learning approach was employed, using object detection to automate the analysis of visual output on the BHU and BRC variants.

2 Methods

2.1 Test Scenario

The System Under Test (SUT), which includes the BHUs and BRCs, is placed inside a dark chamber to eliminate external light and reflections that could interfere with the results. A camera captures images of the SUT inside the dark chamber. The SUT and the camera are controlled by the test bench software (vTestStudio [Gmb24]). These setups are commonly referred to as test benches. The images captured by these test benches are processed to extract relevant information.

The challenge lies in developing a robust, reliable, adaptable, and easy to maintain solution that can accurately retrieve the required information from the captured images.

One example of a test scenario is verifying that state of charge (SoC) is correctly displayed on the BHU. A solution is needed to read the SoC value from images of the devices (BHU and BRC) on the test bench and compare it with the expected SoC value for validation.

Other test scenarios include retrieving and verifying the active assist mode text, checking that error messages are displayed with the correct icon, text, and error code, and ensuring that when the user turns on the bike light, the light icon appears on the BHU display. These elements that display information are referred to as features.

Broadly, all these features can be categorized into two main types: verifying alphanumeric text and verifying the presence of icons at specific positions on the screen.

Although the regions of these features are known, using hard-coded coordinates to locate a specific feature region is not an optimal solution. This is because the text displayed on the screen often varies in length. For example, the SoC value ranges from 0 to 100, with the length of the field varying between 1 and 3 digits. Similarly, the mode text, which indicates the assist mode, could be ECO, SPORTS+, or other variations, leading to different text lengths. Furthermore, changes such as the presence of alerts or icons can shift other displayed information. As a result, determining the region of interest is complex, as it may vary slightly in position and size.

This process can be broken down into two steps. First, the region of interest (where the SoC is displayed) is identified and cropped from the captured image. Next, an optical character recognition (OCR) engine extracts the displayed value.

Supervised object detection models, built on top of convolutional neural networks (CNNs), utilize layers designed to capture patterns associated with the region of interest. These models can detect the region of interest of an object, providing the coordinates needed to extract features in this case.

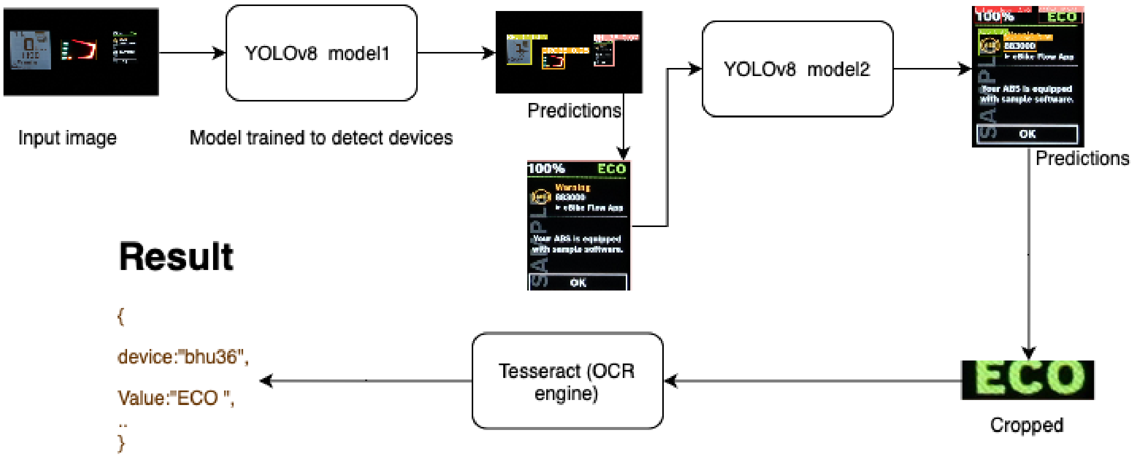

Figure 2.1 – The architecture used for visual recognition

Ultralytics, a machine learning library, was used to facilitate the training process. The Python library provided by Ultralytics leverages the popular YOLO (You Only

Look Once) architecture, a single-stage object detector based on CNNs [JKAH24]. The architecture utilized for this purpose is shown in Fig. 2.1.

First, a YOLOv8 object detection model is trained using images containing various types of devices. Once trained, the model detects these devices in unseen images and provides their bounding box coordinates. These coordinates are then used to crop the image, isolating the region of interest.

Next, the cropped image is passed to a second device-specific model (referred to as YOLOv8 model2 in Fig. 2.1). This model is trained to detect elements and icons, referred to as features under test, such as error and warning icons, the SoC region, or error messages. The second model predicts the presence of these features and provides their coordinates.

If the feature is text-based (e.g., verifying an expected text value), the coordinates are used to crop the feature, and the cropped image is passed to Tesseract, an open-source optical character recognition (OCR) engine. Tesseract extracts the text from the image, and the extracted text is returned as the final result. This result is then compared against the expected value provided by the test bench to determine whether the test passes or fails.

If the feature involves verifying the presence of an icon, the predictions from the second model are directly used to check if the icon is detected.

The second model is device-specific, meaning that for devices such as BHU36, BHU37, BHU35, and BRC38 (which share the same features), a single model can be utilized. However, for devices like the BHU32, which features a black-and-white LED display, a separate model trained specifically for those devices is required.

2.2 Retraining the Model

Ultralytics is designed to offer a streamlined approach to training models with minimal code. While some understanding of machine learning may be necessary for debugging, the development and usage process can become more manageable when annotated data is available for training [Inc24].

To annotate data for object detection models, we used **labelme**, an open-source tool that offers a simple graphical user interface (GUI). With **labelme**, users can draw bounding boxes around objects in an image and assign labels to them. The annotations for each image are then exported as a JSON file.

A project was created with various utilities to streamline the training process, making the implementation of new features or retraining more efficient. For example, an annotation converter was developed to transform **labelme** annotations (which differ from the format expected by Ultralytics) into the correct format. Additional utilities were created for image preprocessing, data organization, and model testing.

Given that new features are often added or new device types introduced, a simplified training process is essential for ensuring adaptability. This solution enables the system to be easily retrained whenever new features are implemented or new device types are produced.

2.3 Fine-Tuning Tesseract

Tesseract, the open-source OCR engine based on LSTM neural networks [ocr24], supports text extraction in various languages and fonts. As described by its developers in the user manual, achieving accurate OCR results typically requires selecting the appropriate configuration parameters, choosing the correct language and font models, and applying image preprocessing techniques.

In some cases, however, it has been observed that fine-tuning Tesseract's language model (LSTM) with custom data may be necessary to improve accuracy. For example, the BHU32 uses a 7-segment font on its LED display, which initially led to inconsistent text extraction. By fine-tuning the base Tesseract 4.0 LSTM model with custom training data derived from BHU32 images, the accuracy of text predictions significantly improved, enabling highly accurate recognition of the 7-segment font on the display.

In cases like this, using machine learning in test automation offers significant advantages. A purely programmatic verification approach would be too rigid and could fail even with minor changes in feature formatting. This would require constant debugging and maintenance of the automation code, which is highly impractical. In contrast, a machine learning approach abstracts much of the implementation and can be easily retrained when feature changes occur, making it more adaptable and efficient.

3 Conclusion

The co-op at Bosch eBike Systems has been a valuable experience, as it allowed me to apply and expand my knowledge in system testing and automation. I gained hands-on experience working with BES3 components and successfully developed a machine learning-based object detection approach for visual output testing on eBike display devices.

Additionally, working with advanced HIL testing tools and managing diverse BES3 components deepened my understanding of scalable testing solutions and enhanced my technical and problem-solving skills. This co-op has solidified my interest in machine learning and system testing, equipping me with industry-relevant experience that will be pivotal in my future career.

I am grateful to Bosch eBike Systems and my supervisors for their support. As the complexity of testing grows, this experience has opened new avenues for using machine learning in test automation and visual verification.

Bibliography

[Gmb24] GmbH, Vector I.: *vTESTstudio*. https://www.vector.com/at/en/products/products-a-z/software/vteststudio/. Version: 2024. – Accessed: 2024-10-06 2.1

[Inc24] Inc. ultralytics: *ultralytics*. https://docs.ultralytics.com. Version: 2024. – Accessed: 2024-10-06 2.2

[JKAH24] Jegham, Nidhal ; Koh, Chan Y. ; Abdelatti, Marwan ; Hendawi, Abdeltawab: *Evaluating the Evolution of YOLO (You Only Look Once) Models: A Comprehensive Benchmark Study of YOLO11 and Its Predecessors*. https://arxiv.org/abs/2411.00201. Version: 2024 2.1

[ocr24] ocr tesseract: *tesseract-ocr*. https://tesseract-ocr.github.io/tessdoc/. Version: 2024. – Accessed: 2024-10-06 2.3

[Sys23a] Systems, Bosch eBike: *eBike Brands with Bosch Components*. https://www.bosch-ebike.com/en/ebikes/brands, 2023. – Accessed: 2024-10-02 1.1

[Sys23b] Systems, Bosch eBike: *Mobility of the Future: eBikes with Bosch Components*. https://www.bosch-ebike.com/en/everything-about-the-ebike/stories/mobility-of-the-future, 2023. – Accessed: 2024-10-02 1.1

[Sys24] Systems, Bosch eBike: *The Smart System*. https://www.bosch-ebike.com/en/products/the-smart-system. Version: 2024. – Accessed: 2024-10-03 1.1, 1.2, 1.3, 1.4

[ZF13] Zeiler, Matthew D. ; Fergus, Rob: Visualizing and Understanding Convolutional Networks. In: *CoRR* abs/1311.2901 (2013). http://arxiv.org/abs/1311.2901

Declaration Of Authorship

I, Uday Kaipa, hereby affirm, that I have prepared the present work without the unauthorized help of third parties and without the use of any aids other than those specified. The data and concepts taken directly or indirectly from sources are marked with a reference to the source. This work has not been submitted to an examination board in the same or a similar form, or published in any other way, either at home or abroad.

Ilmenau, 2024.11.26

Kuday

Uday Kaipa

Approval of Industry Supervisor:

Kusterdingen, 2024.11.26

pki, BOSCH, DE, M,
O,
Mohith.Shivakumar

Digital unterschrieben von pki,
BOSCH, DE, M, O, Mohith.Shivakumar
Datum: 2024.11.19 11:31:53 +01'00'

Mr. Mohith Shivakumar