Product Validation Using Computing and Digital Image Processing.

João Victor Ramirez Konrath
Fernando de Souza Gonçalves Cassiano
Márcio José Moraes
Robert Bosch Ltda.

ABSTRACT

As the automotive industry continues to evolve rapidly, the development of new types and models of fuel injectors becomes necessary. During the injector development stage, numerous questions arise regarding their design and functionality, requiring study time and investment from the company. Validation is a crucial phase for product development, aiming to analyze its compliance before starting production.

One of the analyses conducted for injection approval is the spray angle. To make this possible, a tool called Angle Analyzer was developed, which validates fuel outputs through images captured of current injection condition on a test bench, enabling the identification of possible inconsistencies in the parts, indicating any deformity or impurity. From these analyses, it is possible to store information about the angle of each spray, and if the test bench is not suitable for capturing the images, the system offers flexibility for manual angle adjustment. It is concluded that the use of the tool provides a variety of important information about the product, facilitating its approval and regulation, saving effort and time.

ABOUT THE AUTHOR

João Konrath is a Software Engineer with technician in System development and Mechatronic by SENAI. He is working on Bosch LTDA since 2022. Recently, delivering automation and digitalization projects for manual process for engineering. Contact e-mail: Joao.konrath@br.bosch.com.br.

INTRODUCTION

The automotive industry is continuously advancing and evolving, leading to the development of more complex components designed to enhance and improve key properties that are significant to the end consumer. Figure 1 illustrates the evolution of a fuel injector.

Fuel economy, stability, reduction of pollutant emissions, performance, and design are examples of factors that have been improved over time. These improvements consistently adhere to increasingly stringent standards set by the industry, necessitating time for the development of new products that meet these requirements.

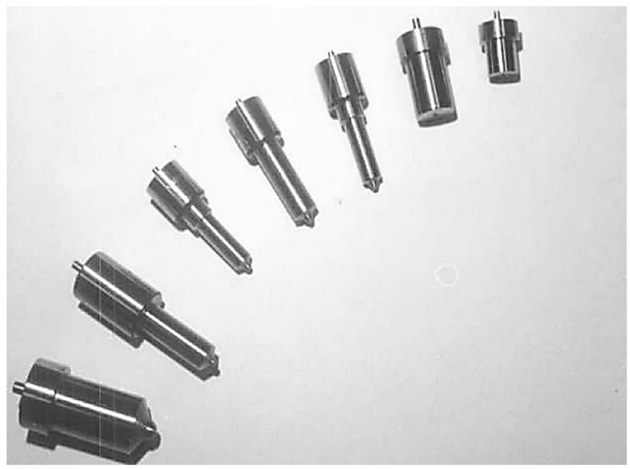


Figure 1. Examples of Injection Nozzles for different vehicles or finalities.
Source: Robert Bosch Ltda. (2014).

Thinking beyond the physical world, we also observe various areas in the digital realm undergoing a highly accelerated evolutionary process, like the advancements in the automotive sector. Several factors contribute to this evolution, including the development of new microprocessors and microcontrollers, highly advanced programming languages, and optimized development methods.

When discussing digital transformation in the automotive field, it is essential to recognize that solutions are evolving drastically and exponentially. The focus is on facilitating the daily lives of those involved in improving the manufacturing and construction of mechanical components for vehicles.

There are numerous examples related to these topics, such as Industry 4.0, Production Planning and Development, and Monitoring systems for automatic preventive maintenance. Each of these contributes through different methods and characteristics, from their application to the results they yield.

The use of Image Processing (IP) is more prevalent than one might imagine; facial recognition, satellite imagery, medical analyses, and image compression are some of the most common examples in our daily lives. The application of this technology in the automotive field can offer significant benefits and advantages.

Image analysis becomes invaluable and crucial for a production line, as it can save hours, days, months, or even years of work by enabling machines to analyze product patterns to assess their quality, determining whether they are suitable for commercialization or production.

The possibilities are endless, provided we have the necessary materials and clearly defined objectives. One of the methods used in Image Processing is pixel analysis, which can identify patterns, errors, or defects.

Applying this technology to injection nozzles, one potential application would be to examine the most critical aspect of their operation: the spray injections. If one of the sprays deviates from the standard or exhibits nonconformities, we can infer that the product has a defect in its operation. Figure 2 illustrates the bores of a nozzle, where such a defect might occur.

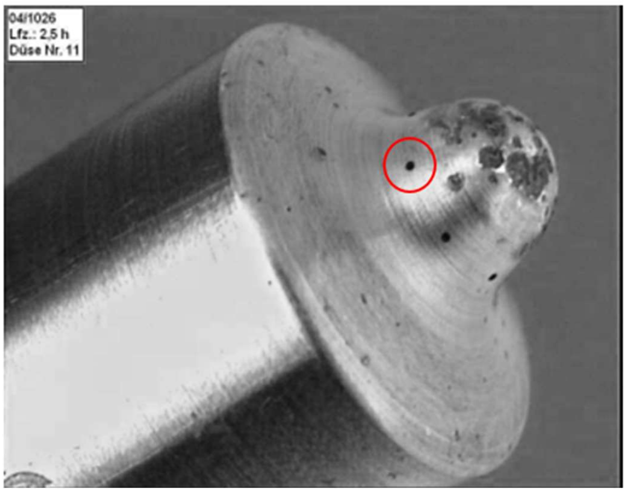


Figure 2. Image of an Injection Nozzle. Bores where the fuel is dispensed.

Source: Robert Bosch Ltda. (2024)

The fuel is ejected through these small holes, and any impurity on then can completely alter the product's operation.

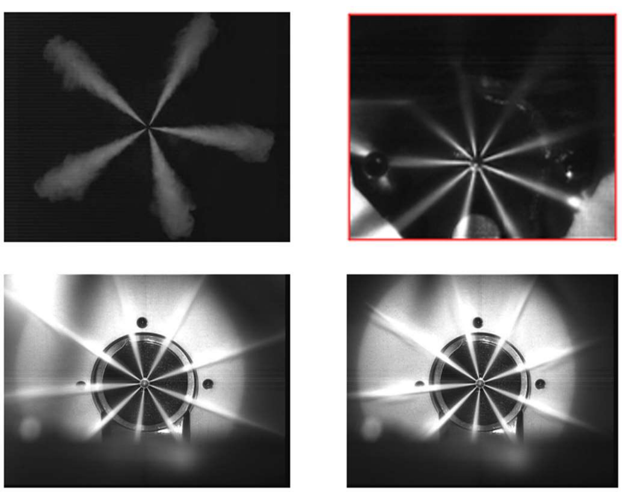


Figure 3. Injection images of different nozzles.

Source: Robert Bosch Ltda. (2024)

Last year, a new type of nozzle began production at Bosch's Curitiba plant. This product originated from China, and we lacked information or data about it, which complicated the approval process. To address this, we developed a system that can automatically analyze the angles of injection sprays, significantly reducing the team's workload.

The use of image processing can assist in identifying defects in this product. One of the factors influencing whether the injection is acceptable is its angle. Depending on the spray injection angle, we can determine if the product has issues or is ready for sale.

IMAGE PROCESSING

Before delving into the project details, it's important to understand how digital image processing works. Unlike human eyes, a computer does not automatically recognize characteristics in an image. Therefore, certain procedures are necessary to enable analysis of digital images.

ACQUISITION – Image acquisition is the process of converting a three-dimensional real-world image into an analog image using optoelectronic transduction, which transforms light into electrical signals.

A capture device performs dimensionality reduction, converting 3-D scenes into a 2-D representation.

Currently, CCD (Charge Coupled Device) cameras are the most used when high accuracy is required. They utilize a matrix of photosensitive semiconductor cells that store an electric charge proportional to the luminous energy. To obtain colored images, CCD cameras use a combination of prisms and color filters to translate colors into RGB components.

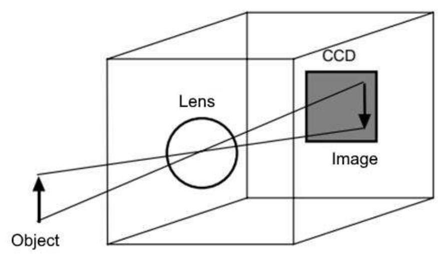


Figure 4. Schematic vision of CCD camera.

Source: Ogê Marques Filho, Hugo Vieira Neto. Processamento Digital de Imagens, 1999, p. 21.

DIGITALIZATION – It used two different process to make the digitalization, spatial and amplitude discretization.

Spatial discretization its responsible to convert each pixel of analogic image into a *M* per *N* matrix, the higher value of *M* and *N* results on a best quality image. The amplitude discretization makes pixels assume an integer value between 0 and 2”-1, the higher *N* value, most the numbers of gray levels present on the image.

The figure 5 shows a digitalized pixel of an image. This matrix it followed pixel by pixel.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Figure 5. Digitalized pixel of an image.

Source: Ogê Marques Filho, Hugo Vieira Neto. Processamento Digital de Imagens, 1999, p. 22.

CONVOLUTION AND MASK – Many operations used in image processing are based on the same fundamental concept: convolution with masks. This involves mathematical operations between two vectors, one representing the mask and the other the image.

Two-dimensional convolution (2D) involves sliding the mask over the image, multiplying the mask by the corresponding area of the matrix it covers. The resulting values are summed to create a new pixel value in the output image.

Following the example:

$$\text{Image } (I) = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 1 \\ 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 1 & 2 \\ 3 & 4 & 5 & 6 & 7 \end{bmatrix},$$

$$\text{Mask } (M) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

I have dimension: 5 x 5

M have dimension: 3 x 3

Output dimension: When we don't have padding, output (O) is calculated by:

(5 – 3 + 1) x (5 – 3 + 1) = 3 x 3

Calculation: For the first pixel of O, overlap M on top-left side of I:

$$O_{(0,0)} = \begin{bmatrix} 1 & 2 & 3 \\ 6 & 7 & 8 \\ 2 & 3 & 4 \end{bmatrix} \text{X} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$O_{(0,0)} = 16$$

After sliding the mask for all slices of the image matrix, making the multiplications and sums between the matrixes, the convolution result will be:

$$O = \begin{bmatrix} 16 & 17 & 27 \\ -6 & -5 & 5 \\ 17 & 27 & -17 \end{bmatrix}$$

The mask that was used is a sharpening filter, positive values on the center and negatives on the sides, showing the transitions and intensities.

When is applicated a filter or made some process in an image, the same process of the example is repeated, but whit different masks.

DEVELOPMENT

The first step in development is understanding how we can analyze the image and the process pipeline of this project. An important aspect is selecting appropriate filters to apply to the images to enhance computational analysis.

FILTER PIPELINE - The first filter applied is the grayscale filter, which converts the image colors to shades of gray, making it easier to perform binarization. Binarization transforms the image into a new image with only two colors: one representing the background and the other representing the object of interest. After binarization, erosion is applied to reduce noise in the image by decreasing white regions and amplifying black regions. Finally, the borders of the spray can be identified.

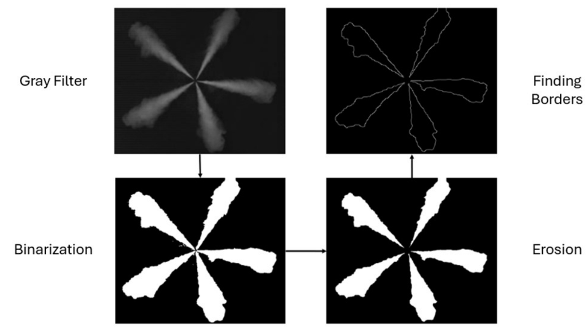


Figure 6. The filter pipeline process applied on injection image.

Source: Robert Bosch Ltda. (2024)

Each border identified is a line, and each line is composed of numerous pixels. The number of pixels can vary depending on the image quality or size. To find the borders, the multi-stage Canny Edge Detection algorithm is used.

DEVIATION POINTS – Identifying deviation points is one of the most crucial steps in the process. These points basically are locals where the border line make a detour, judging these locals a possible deviation. To find them, a threshold value must be established, and the distances between each pixel and the next along the border are compared. When this distance exceeds the threshold, the previous point is considered a deviation point.

The threshold value can be adjusted according to user requirements. As mentioned earlier, the number of pixels depends on certain image characteristics, and the distance between two pixels can vary as well. Setting the threshold is essential to ensure this step functions correctly. Figure 5 illustrates the same process applied to identical images with different thresholds, showing a significant difference in the number of identified points.

Standardly, the threshold came with value 3 and can be changed as the user necessity.

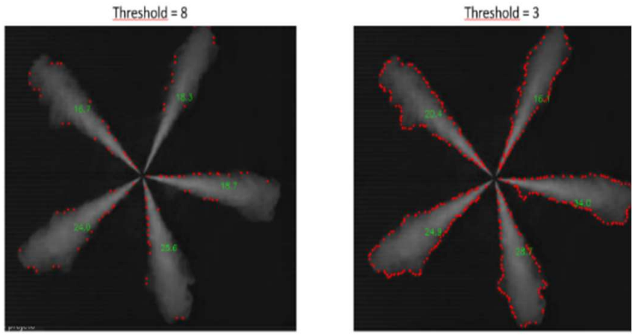


Figure 7. Deviation points with different threshold in the same image.

Source: Robert Bosch Ltda. (2025)

The red dots represent the deviation points. Higher being the threshold, smaller is the quantity of deviations we will be on the border.

FINAL POINTS – This step identify the final of the image tracing a route leaving the center following each deviation points.

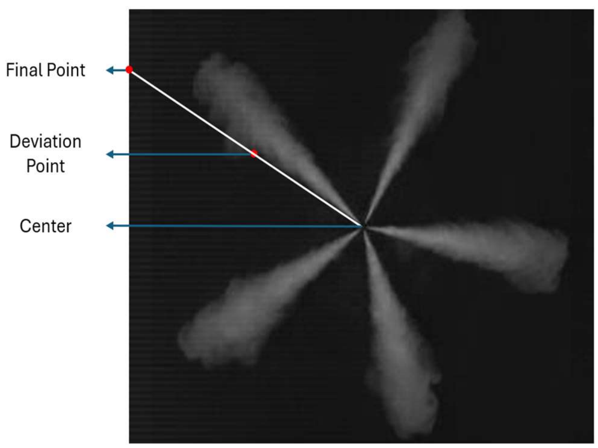


Figure 8. Example of one final point founded for a specific deviation.

Source: Robert Bosch Ltda. (2024)

Is returned some final points, as well as the quantity of deviation points found. A temporally list is created on each spray and store all the final points on it.

DISTANT POINTS – To find the extremities of the border, we pass for that step. The system calculates the Euclidean distance between all final points. Considering two points, (x1, y1) and (x2, y2), we have the following equation:

$$\sqrt{((x2 - x1)^2 + (y2 - y1)^2)} \quad (1)$$

With this equation, we can identify the traced line among them. Each value saved in the temporary list is applied using this formula along with all other values. In the end, all results are compared, and the respective points with the highest values are considered the most distant points of the spray.

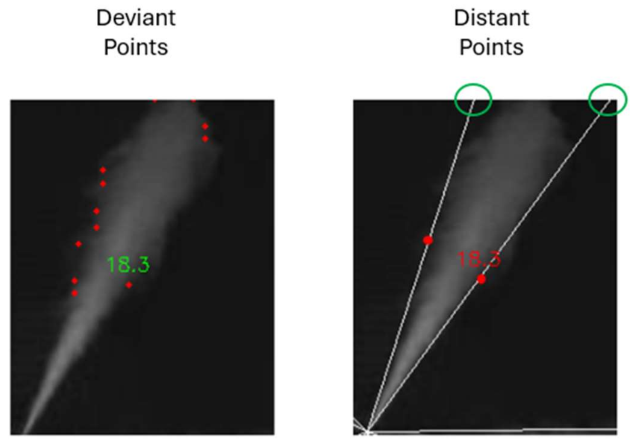


Figure 9. Distant points of a spray.

Source: Robert Bosch Ltda. (2024)

COUPLE POINTS – With all final points identified, we can connect each one to its closest point using the Euclidean distance. A pair of close points theoretically represents the extremity of the spray.

Two pairs of points may share a common point because one of them couldn't find a closer match. However, this is not problematic and will be resolved later without affecting the angle calculation of the other pair.

After identifying the pairs of points, we draw a line from the center to each final point and save them in another temporary list.

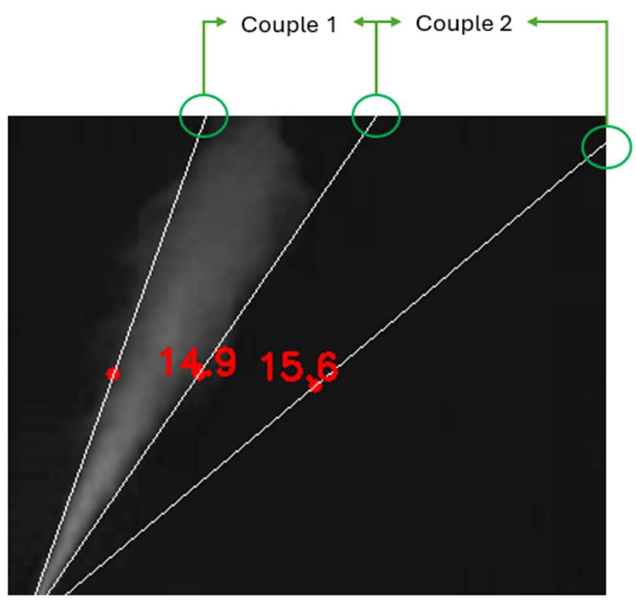


Figure 10. Couple points found on spray.

Source: Robert Bosch Ltda. (2024)

The line doesn’t necessarily have to be visible, the Figure 8 just made her to exemplify how the operation works. On daily use we could set whether it will appear or not.

CALCULATE ANGLE – Once the temporary list is prepared, we can calculate the angle of the spray. To do this, we take the two lines and perform some calculations.

The first step is to define the vectors, using the line values to determine their components.

The vector 1 is the difference between the points line1[1] and line1[0].

$$v_1 = (line1[1][0] - line1[0][0], line1[1][1] - line1[0][1])$$
$$v_2 = (line2[1][0] - line2[0][0], line2[1][1] - line2[0][1]) \quad (2)$$

Vectors defined; the scalar product would be calculated using the equation 4.

$$v_1 . v_2 = v_{1x} . v_{2x} + v_{1y} . v_{2y} \quad (3)$$

Where v_{1x} and v_{1y} are components of v_1 and v_{2x} and v_{2y} are components of v_2. These values are necessary to calculate the magnitude.

$$|v_1| = \sqrt{v_{1x}^2 + v_{1y}^2}$$
$$|v_2| = \sqrt{v_{2x}^2 + v_{2y}^2} \quad (4)$$

The cosine is calculated using these results, and the vectors values.

$$\cos(\theta) = \frac{v_1 . v_2}{|v_1||v_2|} \quad (5)$$

The value is transformed in radian, after that, converted in degrees to show in the screen.

$$\theta_{radian} = \arccos(\cos(\theta)) \quad (6)$$

$$\theta_{degrees} = \frac{\theta_{radian} . 180}{\pi} \quad (7)$$

Finally, we have in degrees the value between two extremity lines of the spray. This calculation is repeated in those couple points list, returning the respective values that need to be show for the final user.

SOFTWARE – To centralize and optimize these functions, ensuring they work effectively without problems, errors, or difficulties, a software application was developed for the engineers working in this area.

The software was created using Python with OpenCV for image processing and displaying analysis results, PySide6 for creating a user interface, and NumPy for handling matrices and vectors to perform the image-related processes discussed in the previous sections of this article.

Python is an easy way to solve this kind of solution, your simple syntax and uncountable libraries, facilitates the process to make the necessary functions save a good time of development. Using the library PyInstaller we can make an executable file for our clients run the application on his own computer without install extra languages or resources.

In 3 months, the solution starts to show some results, analysis, images, degrees values and a utilizable interface. The next months was directed to increase and make actualizations in all functions or UI, always keeping contact with the final user to take new ideas and possible alternatives for problems that appears sometimes.

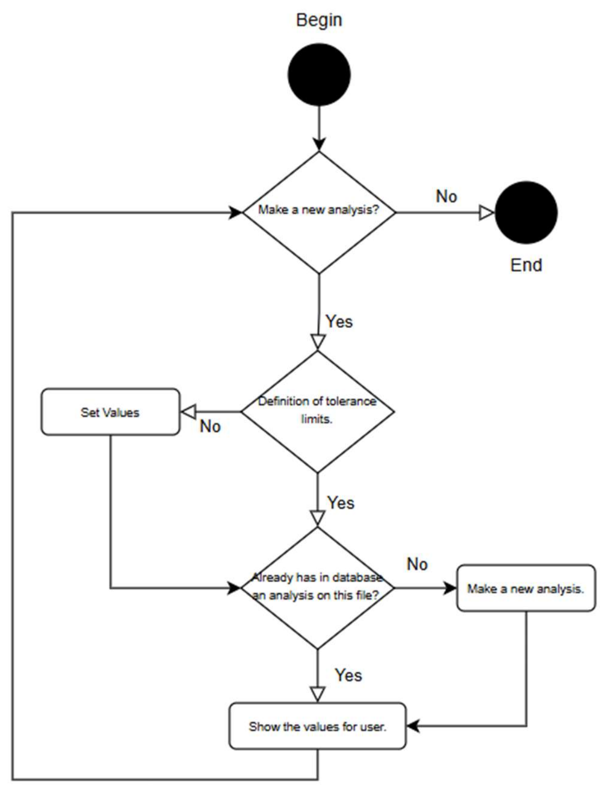


Figure 11. Flow chart of the software process.

Source: Robert Bosch Ltda. (2024)

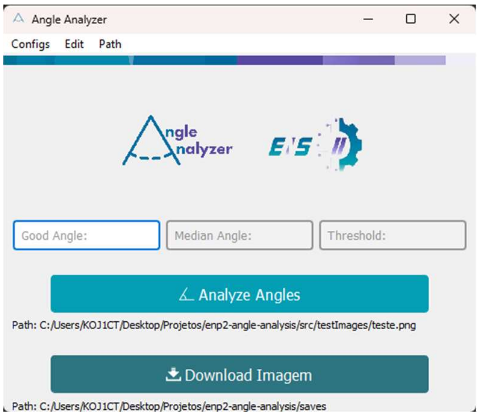
These functions where applied in the UI, focusing on the user experience to being accessed whit effortless ways.

<u>User UI</u> – The simpler and more objective the interface, the better it is. After several updates, three screens were created with clean and intuitive desktop visuals, free from unnecessary elements. These screens are: Main Screen (with two different layouts), Settings, and Viewing Interface.

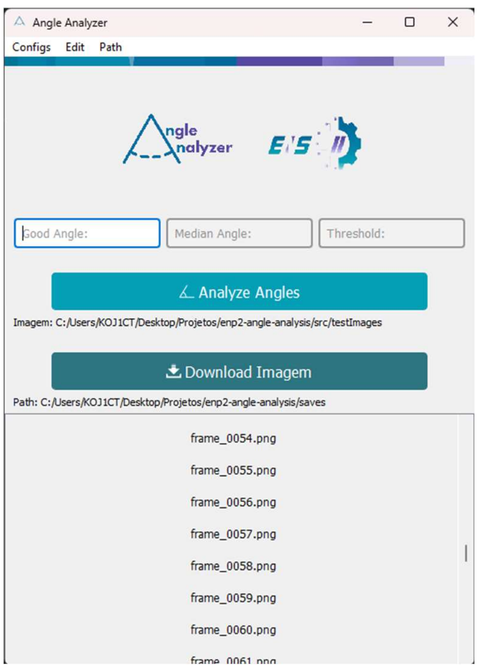
On the Main Screen, users can input the image or folder to be analyzed, set acceptable and unacceptable angle values, define the threshold value, select elements to appear finally, and access the settings screen. At the top, there is a Navbar where some of these functions can be accessed.

Below the "Analyze Angle" button, the user can see the path set for analysis, making it easy to view this information when the system is opened on different days or by different engineers. The "Download Image" button similarly displays the path where the image was saved.

When analyzing a folder, a scroll area with buttons is generated, where each button corresponds to an image. Clicking a button will analyze the respective image.



(a)



(b)

Figure 12. Main Screen layout (a) when would analyze an image, (b) when would analyze a path of images.

Source: Robert Bosch Ltda. (2024)

The Settings Screen is responsible for adjusting all analysis elements, such as color and size. This feature is important depending on the image size, illumination, or nozzle model.

For each element, there is a slider to change its size or thickness, followed by a square displaying the current color. Clicking the square allows the color to be changed.

Next to these editing elements, there is a preview showing how the element will appear, making it easier for engineers to understand how their analysis will look. These changes are saved automatically.

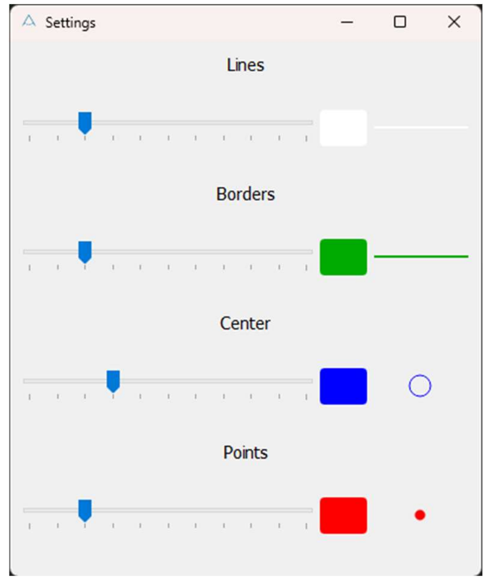


Figure 13. Settings Screen.

Source: Robert Bosch Ltda. (2024)

<u>Analysis Screen and Its Functions</u> - The Analysis Screen is the main feature of the system, allowing users to conduct their analysis in the most effective way possible.

A high-quality image aids the backend process in identifying all necessary details. The Angle Analyzer includes methods to ensure analysis is possible even if users lack test benches with professional equipment for perfect images. Users can move, add, or remove the extremity spray lines and adjust the detected center of the nozzle, addressing this need. When these elements are modified, the angle value will also change.

Even with a good image, Image Processing is not 100% accurate. The Analysis Screen addresses this issue by offering features that make analysis easily customizable. The ability to zoom in on the image helps users move lines and centers with greater precision, allowing any type of image to be analyzed within the system.

Lines can be moved by clicking on the red points with the left mouse button, allowing them to follow the cursor as desired. This enables manual alignment with the sides of the spray or alternative analyses referencing other points on the image. For precise alignment, a zoom function has been implemented.

If a line is not generated automatically, it can be added by double-clicking with the cursor. Similarly, if a line is generated unnecessarily, it can be deleted by right-clicking on the red points.

In the top left corner, users can view the median angle of all values displayed on the screen and access a button to recalculate the final points if the center needs adjustment.

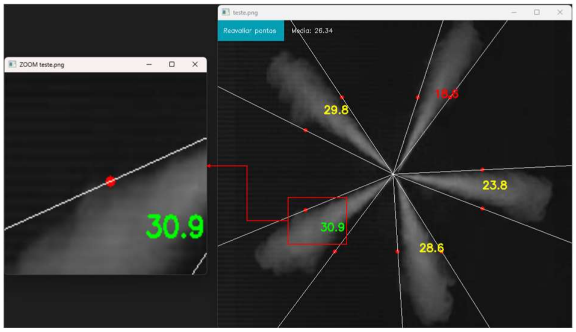


Figure 14. Analysis image with zoom function on signposted area.

Source: Robert Bosch Ltda. (2024)

Json Save Method – Gaining time through multiprocessing is a key focus for technology professionals, aiming to save time by reducing activities or optimizing system functions.

The Angle Analyzer employs a method to prevent repeated analysis on the same image, minimizing storage space usage and enabling fast searches within its database.

JavaScript Object Notation (JSON) files are commonly used in APIs and web applications. Their structure consists of data values classified by keys, making it easy to locate information and facilitating data transfer between different technologies.

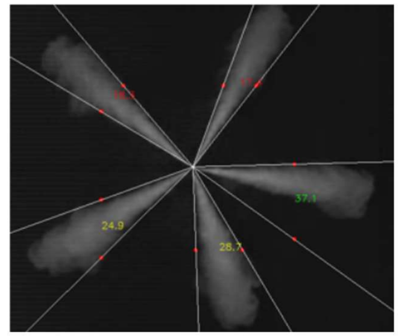
```
{
    "name": {
        "first": "João",
        "last": "Konrath"
        },
    "age": 22,
    "hobbies": ["soccer","gym"]
}
```

Figure 15. Basic JSON example file.

Source: Robert Bosch Ltda. (2024)

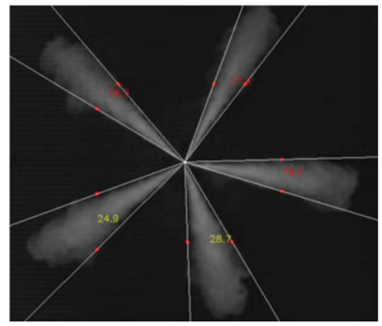
Key points of analysis are saved in a JSON file, specifically the center point and the current final points. This eliminates the need to repeat all pipeline steps the next time an engineer wants to analyze the same image.

The system locates the key with the same path and simply traces the lines using the data stored in the JSON. Any changes made by the user are automatically updated in the JSON file.




```
"C:/Users/KOJ1CT/Desktop/Projetos/enp2-angle-analysis/src/testImages/teste.png": {
    "Actual Points": "[(437, 510), (293, 510), (0, 534), (0, 355), (598, 244), (598, 477),
    "Center Point": "(284, 252)"
},
```

(a)




```
"C:/Users/KOJ1CT/Desktop/Projetos/enp2-angle-analysis/src/testImages/teste.png": {
    "Actual Points": "[(437, 510), (293, 510), (0, 534), (0, 355), (598, 244), (598, 346),
    "Center Point": "(284, 252)"
},
```

(b)

Figure 16. Examples of analysis (a) before making changes, (b) after making changes.

Source: Robert Bosch Ltda. (2024)

Another JSON file stores all the settings made by user on Settings Screen, together with the path of search image and save path local.

```
{
    "Path": "C:/Users/KOJ1CT/Desktop/Projetos/enp2-angle-analysis/src/testImages",
    "Save Path": "C:/Users/KOJ1CT/Desktop/Projetos/enp2-angle-analysis/saves",
    "Lines Color": "(255, 255, 255)",
    "Borders Color": "(0, 170, 0)",
    "Center Color": "(0, 0, 255)",
    "Adjust Color": "(255, 0, 0)",
    "Lines Width": 2,
    "Borders Width": 2,
    "Center Width": 3,
    "Adjust Width": 2,
    "Language": "en"
}       You, 7 minutes ago • Uncommitted changes
```

Figure 17. Settings JSON is stored all the edit settings settled by user.

Source: Robert Bosch Ltda. (2024)

CONCLUSION

In conclusion, the integration of various technologies can significantly address and facilitate numerous challenges and tasks. This article highlighted the impact of technology on the automotive industry, showcasing a solution that transforms complex product approval processes into manageable and tangible tasks.

The continuous evolution of technology, particularly the increasing use of artificial intelligence, offers promising opportunities for enhancing existing solutions and developing new ones. In the industrial sector, AI is proving invaluable in streamlining administrative tasks and optimizing production lines.

The Angle Analyzer exemplifies how AI methods can enhance accuracy, provided there is a robust photo database supported by an appropriate test bench. The benefits of such technological solutions are extensive, including substantial financial savings. By identifying errors in products and processes, companies can conserve resources both locally and globally.

The Angle Analyzer represents a versatile tool that can be adapted to solve various challenges within the product approval environment. Image processing, with its vast potential, can yield excellent results when thoroughly studied and effectively applied. As we continue to explore and exploit these technologies, the possibilities for innovation and improvement are boundless.

REFERENCES

[1] Arlindo Fernandes de Aquino Neto, Silvio Roberto Fernandes de Araujo. **ACELERADOR DO CÁLCULO DE DISTÂNCIA EUCLIDIANA EM HARDWARE.** 2020. Available in: < https://repositorio.ufersa.edu.br/server/api/core/bitstreams/20b46849-3e17-4235-ab17-92b0e6a2dac6/content>

[2] Márcio Portes de Albuquerque, Marcelo Portes de Albuquerque. **Processamento de Imagens: Métodos e Análises.** 2000. Available in: < https://mesonpiold.cbpf.br/e2002/cursos/NotasAula/PDSI.pdf>

[3] Ogê Marques Filho, Hugo Vieira Neto. **PROCESSAMENTO DIGITAL DE IMAGENS**. 1999. Available in: < http://projetoaprendizagemgrupo4.pbworks.com/w/file/fetch/96395952/Processamento%20Digital%20de%20Imagens.pdf>

[4] Michael P. Ekstrom. **Digital Image Processing Techniques.** 1984. Available in: < https://books.google.com.br/books?id=h9qN1KlX388C&lpg=PP1&dq=digital%20image%20processing&lr&hl=pt-BR&pg=PP1#v=onepage&q&f=true>

[5] **Canny Edge Detection.** OpenCV Documentation. Available in: < https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html

> [Accessed: Apr. 28, 2025]