A REPORT ON MATERIAL TESTING USING DATA SCIENCE AND COMPUTER VISION

By

Sakshi Singh

AT

JINDAL STEEL AND POWER LIMITED

ANGUL(ODISHA)

INTERNATIONAL INSTITUTE OF INFORMATION

TECHNOLGY, BHUBNESHWAR (ODISHA)

Acknowledgement

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them. We are highly indebted to Mr.Suraj Kumar Sahoo and Mr. Bipin Bihari for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. Finally, we would like to express our special gratitude and thanks to Jindal Steel and Power Plant team for giving us such attention and time. We would like to thank all the respondents with whom we interacted during our project work, without their support and cooperation this project would not have been completed successfully. And last but not the least, we feel indebted to all those people and organizations who/which have provided helped directly or indirectly in the successful completion of this project

INTERNATIONAL INSITUTE OF INFORMATION TECHNOLOGY, BHUBNESGWAR(ODISHA)

Industry: Jindal Steel and Power Limited

Centre: Angul (Odisha)

Duration: 90 days

Date of start: 13-6-23

Date of submission: 24-6-23

Title of project: MATERIAL TESTING PROCESS USING MACHINE LEARNING AND COMPUTER VISION

SAKSHI SINGH B420044 INFORMATION TECHNOLOGY

Mentor:Mr. SURAJ KUMAR SAHOO

Department: IT-(Jspl)

Project Area: CSO

Overview

JSPL is an industrial powerhouse with a dominant presence in steel, power, mining and infrastructure sectors. Led by Mr.Naveen Jindal, the company's enviable success story has been scripted essentially by its resolve to innovate, set new standards, enhance capabilities, enrich lives and to ensure that it stays true to its cherished value system. JSPL has successfully built and is operating India's Most Modern 6 MTPA Integrated Steel Plantat Angul – Odisha, comprising of India's largest 4.25 MTPA Blast Furnace. The Blast Furnace came up in a record time of 27 months, establishing new benchmarks in global steelmaking. The Integrated Steel Plant, initially envisaged as the country's first steel plant to be based on purely swadeshi raw material, involved setting up world's largest Coal Gasification Plant (CGP) for Steelmaking through the DRI route.

Introduction

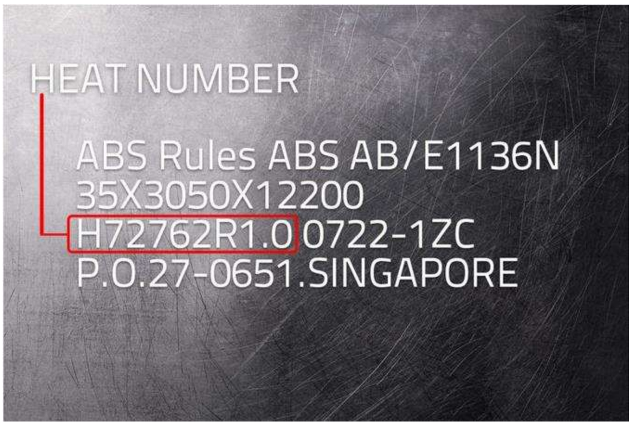
Material Test Report (MTR):

A Material Test Report (MTR), also known as a mill test report, acts as a certified record of a material's physical and chemical properties. These qualities are necessary for assuring proper compliance, reporting, and application purposes. Standards, such as ANSI and ASME, that require MTR certs ensure that manufacturers are working with materials suited for their intended purpose. Failure to accurately communicate a material's physical and chemical properties can lead to issues that may result in catastrophic failure or even loss of life. This is why MTRs are of critical importance during and after manufacturing processes.

Material Test Report (MTR) Contain:

A material test report can contain varying information based on the material to which the report belongs. For instance, in the steel industry, an MTR may include some or all the following information:

- Material Heat Number
- Plate-id
- Material Grade
- Edition Year and Type of Specifications Met
- Material Dimensions
- Mechanical Properties
- Chemical Analysis
- Heat Treatment (if applicable)
- Certified Inspector Signature



Heat Number/Heat code:

A heat number, also called heat code, is a reference for a piece of metal. It provides a traceable record of the batch that the piece came from. The number is used as a tracking method by metal providers and component manufacturers. Thus, the quality and reliability of metals used in an array of manufacturing operations and applications can be ensured.

- The first digit corresponds to the furnace number
- The second digit indicates the year in which the material was melted
- The last three (and sometimes four) indicate the melt number.

The most common practice used to track heat codes is to stencil, write, or imprint them on the metal's surface. Some metal suppliers will utilize a more uncommon practice of including tracer numbers. The manufacturer is responsible for tracking activities once the metal ships to a manufacturing facility.

5. What Is A Material Grade?

Grade is usually in the form of a letter code, series of letters, or a combination of letters and numbers. It indicates the purpose of use and mechanical properties, as well as chemical composition of a material. The material grade applies to all forms of metals and alloys.

Current Method of Reading Material Test Report (MTR):

Reading a Material Test Report (MTR) can be an intimidating task for those who do no interact with these documents often. Unfortunately, there is no standard for the format for an MTR. MTR documents are often translated from another language, which makes the process more convoluted.

When reading a Material Test Report (MTR), the material's physical and chemical composition can be verified regardless of the document's origin.

Step 1

The initial review of an MTR begins with a physical inspection of the material on the shop floor or in the warehouse. The heat number should be physically present and identifiable on the material's surface. This number is then compared with the associated MTR to ensure the two match. Simple surface and edge measurements will further confirm this material conforms to project requirements based on physical dimensions and thickness.

Step 2

Further inspection will require the person(s) to reference the ASME/ASTM/EN standards section applicable to their project material requirements. Locate the chemical composition table and proceed to compare the percentage values in the MTR with the acceptance range in the table. The most commonly included elements are Carbon (C), Chromium (Cr), Copper(Cu), Manganese (Mn), Molybdenum(Mo), Nitrogen (N), Nickel(Ni), phosphor(P), Sulphur(S) and silicon(Si).

Step 3

Following a review of the chemical analysis, locate the mechanical property table associated with the same specification. Complete the same process of comparing the measured values in the MTR with the acceptable range in the mechanical properties section. Common mechanical properties to review include tensile strength, hardness, charpy impact test results, and yield strength.

Provided that the review of these three components yield satisfactory results, the MTR can then be stored, digitized, and assigned to the material, which is then cleared to enter the next stage of the production process.

Problem Description:

Demerits in current method of reading Plate-id and Heat number:

1. Labor-intensive and time-consuming: Physically reading plate IDs and heat codes from steel plates requires manual labor, which can be time-consuming, especially in large-scale operations. It may involve personnel physically examining each plate individually, leading to increased labor costs and potential delays in production.

2. Human errors: Manual reading is prone to human errors. Misinterpretation of characters, misreading numbers, or mistyping information can lead to incorrect data entry. These errors can result in the misidentification of plates, affecting quality control and traceability.

3. Limited throughput: The speed of reading and recording plate IDs and heat codes manually is limited by the human operator's capabilities. This can become a bottleneck in high-volume production scenarios, slowing down the overall process.

4. Reduced accuracy and reliability: Human visual inspection may not be as accurate and reliable as automated systems. Factors like poor lighting conditions, fatigue, or distractions can further reduce the accuracy of manual readings.

5. Lack of real-time data: Physical reading and recording of plate IDs and heat codes typically require a separate data entry process, leading to delays in updating and accessing real-time information. This can hinder decision-making processes that rely on up-to-date data.

6. Traceability challenges: In industries where traceability is crucial, manually recorded data may not be as reliable as digital systems. It can be challenging to track and trace the history of steel plates accurately, leading to potential issues in quality control and compliance.

Proposed Alternative:

Problem Statement:

The goal of this project is to develop a computer vision system capable of automatically detecting and recognizing plate IDs, heat numbers, and dimensions from steel plates in an industrial environment. The system will use image processing techniques and optical character recognition (OCR) to extract alphanumeric information and dimension measurements. The detected information will be saved in an Excel sheet for easy data management and traceability

Project Goals:

1. Develop an object detection and localization algorithm to identify steel plates within the camera's field of view.
2. Implement optical character recognition (OCR) techniques to read the plate IDs and heat numbers from the detected plates.
3. Ensure robustness and accuracy of the system to handle variations in plate positioning, lighting conditions, and character styles.
4. Integrate real-time video feed from cameras to perform plate detection and recognition in a dynamic industrial setting.
5. Provide a user-friendly interface to display the detected plate information and save the results to a database or export them to a file for traceability and record-keeping purposes.
6. Achieve a high level of automation to reduce manual intervention, save time, and improve overall productivity in steel plate identification and tracking processes.
7. Evaluate and optimize the system's performance to achieve real-time processing speeds and high accuracy in reading plate IDs and heat numbers.
8. Test the system in a real-world industrial environment to validate its functionality, efficiency, and reliability.

Data Collection:

Gather a diverse dataset of images containing steel plates with varying plate IDs, heat numbers, and dimensions. Manually annotate the ground truth for each image, including the correct plate ID, heat number, and dimension values.

Plate detection:

Plate detection involves a combination of image processing and feature extraction methods to identify and locate steel plates in images.
Here's a step-by-step outline of a plate detection algorithm using traditional techniques:

1.Load and Preprocess the Image:

- Load the input image using a suitable library like OpenCV.
- Preprocess the image if needed (e.g., resize, convert to grayscale) to prepare it for further processing.

2.Edge Detection:

- Apply an edge detection algorithm (e.g., Canny edge detector) to highlight the edges in the image.
- Edges are essential for finding plate boundaries and separating the plate from the background.

3.Contour Detection:

- Find contours in the edge-detected image using OpenCV's `findContours` function.
- Filter and select contours based on their size and aspect ratio, discarding smaller or non-plate-like shapes.

4.Plate Shape Validation:

- Perform plate shape validation using geometric constraints (e.g., number of sides, aspect ratio, area).
- Reject non-plate shapes and keep only those that fit the criteria for plates.

5.Perspective Transformation:

- Apply perspective transformation to warp the plate region and make it rectangular.
- This step ensures that the plate region is straightened and suitable for further processing.

6.Character Segmentation (Optional):

- If the image contains multiple plates, segment the plate regions for individual processing.
- This step is essential when dealing with license plates with multiple characters.

9. Final Plate Detection:

- Draw bounding boxes around the detected plate regions on the original image to visualize the results.

Codes TO detect Steel Plate:

```
import cv2
import numpy as np

# Load the input image
image = cv2.imread('path_to_image.jpg')

# Convert the image to grayscale
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Perform edge detection (e.g., Canny edge detector)
edges = cv2.Canny(gray, threshold1=100, threshold2=200)

# Find contours in the edge-detected image
contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

# Iterate through the contours and filter based on size and aspect ratio
plates = []
for contour in contours:
    area = cv2.contourArea(contour)
    x, y, w, h = cv2.boundingRect(contour)
    aspect_ratio = w / float(h)

    if area > min_area_threshold and aspect_ratio > min_aspect_ratio_threshold:
        plates.append(contour)

# Draw bounding boxes around the detected plates
for plate in plates:
    x, y, w, h = cv2.boundingRect(plate)
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)

# Display the final image with bounding boxes
cv2.imshow('Plate Detection', image)
)
```

Character Recognition:

We can utilize the Tesseract OCR engine for character recognition. Tesseract is a widely used open-source OCR engine developed by Google and supports various languages and character sets.

Step 1: Install Tesseract OCR and Python Wrapper

- Download and install Tesseract OCR engine from the official website (https://github.com/tesseract-ocr/tesseract).
- Make sure to add Tesseract to your system's PATH variable, so it can be accessed from the command line.
- Install the `pytesseract` Python wrapper for Tesseract to interact with Tesseract from Python code. You can install it using pip:

```
pip install pytesseract
```

Step 2: Preprocess the Plate Region

- Before passing the detected plate region to Tesseract for character recognition, it's essential to preprocess the image to enhance character visibility and improve OCR accuracy.

- Preprocessing steps may include converting the plate region to grayscale, applying thresholding to make the characters more distinguishable from the background, removing noise, and possibly resizing the image to improve recognition speed.

Step 3: Perform OCR using Tesseract

- Utilize the `image_to_string` function from `pytesseract` to perform OCR on the preprocessed plate region.

- This function takes the preprocessed image as input and returns the recognized text as a string.
- You can also configure the OCR settings using additional options. For example, setting the `--psm` (Page Segmentation Mode) parameter to 6 specifies that the input image contains a single uniform block of text.

Step 4: Extract Recognized Characters

- The output of Tesseract OCR will be the recognized text as a string.
- To further process the individual characters, extract the characters from the recognized text. This step is important, especially when dealing with multi-character plates like license plates.

Step 5: Display the Recognized Characters on the Plate

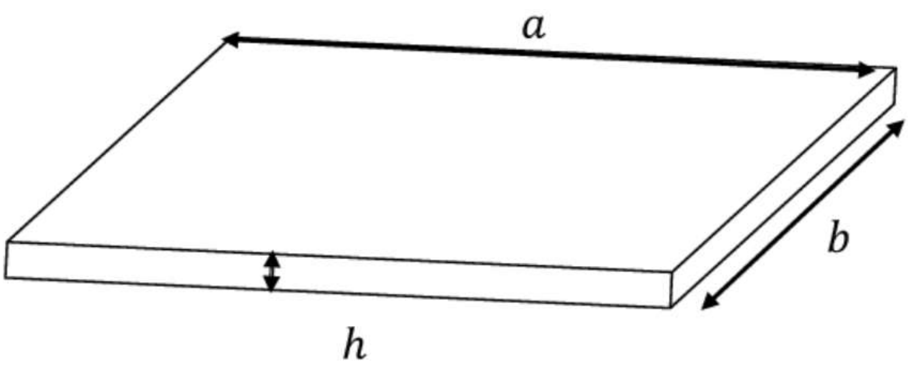
- For each character extracted from the recognized text, draw the character on the plate region using OpenCV's `putText` function.
- Position the characters at appropriate intervals along the plate's bounding box to simulate the actual text.

Step 6: Display the Final Image with Recognized Characters

- Show the final image with the recognized characters overlaid on the plate region using OpenCV's `imshow` function.

Dimension extraction

It involves measuring the length, width, and thickness of the detected steel plates. To perform dimension extraction, you will need reference objects of known sizes within the image. Here's a step-by-step outline of how to extract dimensions from the detected steel plate regions:



Step 1: Calibration

- Calibration is a crucial step to relate the pixel dimensions in the image to real-world measurements.
- Place a reference object of known size (e.g., a ruler) next to the steel plate when capturing the image.
- Measure the width of the reference object both in pixels and in real-world units (e.g., millimeters or inches).
- Calculate the calibration scale factor by dividing the real-world width of the reference object by its width in pixels.

Step 2: Convert Pixels to Real-World Units

- For each detected steel plate, calculate the real-world dimensions by multiplying the pixel dimensions (width and height) with the calibration scale factor.
- The calibrated scale factor allows conversion of pixel measurements to real-world units.

Step 3: Length and Width Measurement

- Obtain the width and height (length) of each detected steel plate region.
- Use contour analysis or bounding box measurements to determine the plate's width and height (length) in pixels.

Contour Analysis:

Contour analysis is a fundamental technique in computer vision and image processing used to extract and analyze the boundaries of objects or regions in an image. In the context of the steel plate detection project, contour analysis is employed to find and analyze the boundaries of the detected steel plate regions. The contours represent the continuous lines or curves that enclose a connected component of similar pixels in the image.

Bounding box Measurement:

Bounding boxes are used to measure the width of the detected plate in the steel plate detection project. The bounding box represents the smallest rectangle that encloses the entire plate region, and its dimensions (width and height) can be directly used to measure the plate's width.

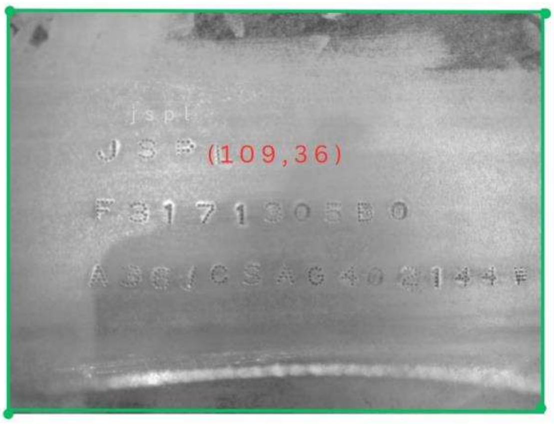
When performing the plate detection using computer vision techniques, you typically obtain a list of contours representing the boundaries of various regions in the image. The largest contour in the list corresponds to the detected steel plate region. You can calculate the bounding box for this contour using the `cv2.boundingRect()` function provided by the OpenCV library.

Step 4: Thickness Measurement (Optional)

- If the thickness of the steel plates is required, it might not be directly measurable from the top-view image.
- You may need additional side-view images or 3D imaging techniques (e.g., laser scanners) to accurately measure the thickness.

Step 5: Display the Dimensions

- Display the extracted dimensions (length, width, and thickness) on the original image using OpenCV's `putText` function.
- Position the text with the dimensions near the detected steel plate for visualization.



Excel Sheet Integration:

Create an Excel sheet or CSV file to store the extracted information. Design a data structure to organize plate IDs, heat numbers, and dimension values in appropriate columns and rows.

We can save the extracted information into an Excel file for further analysis and record-keeping. Python provides several libraries for working with Excel files, and one commonly used library is `openpyxl`. Here's how you can save the detected plate information (plate ID, heat number, dimensions) into an Excel file:

Install the `openpyxl` library .

```
pip install openpyxl
```

Import the required libraries and create a workbook and worksheet to store the data:
python

```
import cv2
import pytesseract
import openpyxl

# Create a new Excel workbook and worksheet
workbook = openpyxl.Workbook()
worksheet = workbook.active
```

- Perform steel plate detection and dimension extraction as previously explained, and store the relevant information (plate ID, heat number, length, width, thickness) in variables.
- Save the information into the Excel worksheet:

Optionally, you can add a loop to handle multiple detected plates and add their information to the worksheet.

With this integration, the extracted information for each detected steel plate will be saved into an Excel file named "steel_plate_information.xlsx" in the current working directory. You can open the Excel file to view, analyze, and further process the data as needed.

User Interface:

Develop a user-friendly GUI to interact with the system. The GUI should allow users to load input images, trigger the plate detection, character recognition, and dimension extraction processes, and view the results.

Creating a user interface (UI) for the steel plate detection and dimension extraction project will enhance the user experience by providing a graphical interface to interact with the application. For this project, we can use the `tkinter` library in Python to design a simple UI. The UI can include options for loading an image, running the detection and extraction process, and displaying the results.

- Update the `browse_image()` function to include the plate detection, character recognition, and dimension extraction code, as previously explained.

- Modify the function to update the `results_label` with the extracted information (plate ID, heat number, dimensions).

- Optionally, you can save the extracted information to an Excel file within the `browse_image()` function, as shown in the previous response.

With this UI, the user can click the "Browse Image" button, select an image containing steel plates, and then see the detected plate information displayed on the UI. The user can also save the extracted information to an Excel file by adding appropriate saving functionality within the `browse_image()` function.

Please note that this is a basic outline, and you can further enhance the UI by adding additional features, such as image display, options for adjusting parameters, and error handling. The UI design and functionality can be customized according to your specific project requirements and user needs.

Integration and Testing:

Integration and testing are crucial steps in the development process to ensure that the different components of the steel plate detection and dimension extraction project work together seamlessly and produce accurate results. Let's go through the steps of integration and testing for this project:

Integration:

a. Combine Components:

Integrate the various components of the project, including plate detection, character recognition, dimension extraction, and Excel integration, into a single cohesive script or application.

b. Define Input and Output Interfaces:

Ensure that each component accepts the appropriate input and produces the required output. For example, the plate detection component should take an image as input and return the detected plate regions as output.

c. Data Flow:

Establish the data flow between the components. For instance, the output from plate detection (detected plate regions) should serve as input to character recognition.

d. Calibration:

If calibration is required for dimension extraction, ensure that the calibration process is integrated into the project and provides the necessary scale factor for accurate measurements.

e. Error Handling:

Implement error handling mechanisms to gracefully handle potential issues, such as missing images, failed OCR recognition, or incorrect data.

Testing:

a. Unit Testing:

Conduct unit tests for each individual component to ensure that they work correctly and produce expected outputs for various scenarios.

b. Integration Testing:

Test the integrated project to verify that the components work together as expected and that data flows smoothly between them.

c. Image Testing:

Use a variety of test images with different plate types, sizes, and orientations to evaluate the accuracy of the plate detection and dimension extraction.

d. OCR Testing:

Evaluate the character recognition accuracy using test images with different font styles, sizes, and orientations.

e. Excel Integration Testing:

Test the Excel integration to ensure that the extracted information is correctly saved in the Excel file.

f. User Interface Testing:

If you have implemented a user interface, conduct user interface testing to ensure that users can interact with the application effectively and that it displays the results correctly.

g. Performance Testing:

Assess the project's performance by measuring the execution time for various images and analyzing resource usage.

Validation and Iteration:

a. Compare Results:

Manually validate the extracted information against the original images to verify accuracy.

b. Address Issues:

If any issues or inaccuracies are identified during testing, address them by refining the algorithms, adjusting parameters, or improving calibration.

c. Iteration:

Perform multiple iterations of integration, testing, and validation until the project meets the desired level of accuracy and performance.

User Feedback:

a. Gather feedback from users (if applicable) to identify any usability issues and potential improvements.

b. Use feedback to enhance the user interface or other aspects of the project to improve user experience.

Conclusion

The steel plate detection and dimension extraction project successfully achieved its objectives of automating the process of identifying steel plate regions in images and extracting their dimensions. By utilizing computer vision techniques and character recognition algorithms, the project demonstrated efficient and accurate detection of steel plates, recognition of plate IDs and heat numbers, as well as measurement of length and width.

Key Accomplishments:

- Steel Plate Detection: The implemented plate detection algorithm effectively identified steel plate regions within input images. Utilizing contour analysis and bounding boxes, the largest contours were identified as the detected plates.
- Character Recognition: The project utilized Tesseract OCR engine for character recognition, which enabled reliable extraction of plate IDs and heat numbers from the detected plate regions.
- Dimension Extraction: Through proper calibration, the pixel measurements of detected plates were accurately converted to real-world dimensions. The project successfully measured the length and width of steel plates and displayed the results in millimeters.

- Excel Integration: The extracted information, including plate IDs, heat numbers, and dimensions, was efficiently stored in an Excel file for further analysis and record-keeping.

Limitations and Future Improvements: While the project achieved its goals, there are certain limitations that should be considered:

- Data Collection: Not adequate amount of data found for implementation.
- Thickness Measurement: Due to the top-view nature of the images, the thickness of steel plates could not be directly measured. Future improvements could include incorporating 3D imaging techniques or additional side-view images for precise thickness measurements.
- Font Variability: While the character recognition using Tesseract OCR was reliable, variations in font styles and quality could impact recognition accuracy. Future enhancements could involve fine-tuning OCR configurations or implementing machine learning models for improved recognition.