# PROJECT REPORT

**on**

**Cut and Bent Rebar Bend Detection**

**Submitted to**

**KIIT Deemed to be University**

**In Partial Fulfilment of the Requirement for the Award of**

**BACHELOR’S DEGREE IN**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

**Nikhil Choudhary**

21052166

**UNDER THE GUIDANCE OF**

**Pradeep Kumar Malik**



**SCHOOL OF COMPUTER ENGINEERING**

**KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY**

**BHUBANESWAR, ODISHA - 751024**

KIIT Deemed to be University

School of Computer Engineering

Bhubaneswar, ODISHA 751024



CERTIFICATE

This is certify that the project entitled

“*Cut & Bent Rebar Bend Detection”*

submitted by

Nikhil Choudhary

21052166

is a record of Bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during the year 2022-2023, under our guidance.

Date: 23/11/2024

Pradeep Kumar Malik

Project Guide

**Acknowledgements**

We are profoundly grateful to **Pradeep Kumar Malik** of **Affiliation** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. .....................

Nikhil Choudhary

21052166

Contents

|  |  |  |
| --- | --- | --- |
| 1 | Introduction | |
| 2 | Basic Concepts/ Literature Review | |
|  | 2.1 | What is Image Processing |
|  | 2.2 | What is Open-CV |
|  | 2.3 | What is Keras |
| 3 | Problem Statement / Requirement Specifications | |
|  | 3.1 | Project Planning |
|  | 3.2 | Project Analysis (SRS) |
| 4 | Implementation | |
|  | 4.1 | Methodology / Proposal ........................... |
|  | 4.2 | Testing / Verification Plan ……………. |
|  | 4.3 | Result Analysis / Screenshots …………. |
|  | 4.4 | Quality Assurance …………………….. |
| 5 | Standard Adopted | |
|  | 5.1 | Design Standards . . . . . . . . . . . . . . . |
|  | 5.2 | Coding Standards . . . . . . . . . . . . . . |
|  | 5.3 | Testing Standards . . . . . . . . . . . . . . . |
| 6 | Conclusion and Future Scope | |
|  | 6.1 | Conclusion ……………………….. |
|  | 6.2 | Future Scope ………………………. |
| References | | |
| Project Contribution | | |
| Plagiarism Report | | |

*Bend Detection*

Chapter 1

Introduction

The Cut and Bent (CAB) process is crucial for producing various components. However, manual inspection of bends for quality control is time-consuming, prone to human error, and can lead to production bottlenecks. Automating bend detection and measurement offers significant advantages in efficiency, accuracy, and cost reduction.  
  
This project aims to develop a robust and scalable solution for CAB bend detection and measurement using computer vision and deep learning techniques. By leveraging image processing algorithms so we can achieve real-time bend detection, classification, and measurement, enabling automated quality control and process optimization in the rebar manufacturing industry.

Page 1

**Key Objectives:**

* Develop a system for automated bend detection in cut and bent rebar
* Utilize deep learning models for image classification of various Rebar Cases.

**Approach:**

* Extracting the drawings/rebar images from the processed BBS image file and storing the extracted images in a separate folder.
* Pre-Processing the extracted BBS image files by removing noise and converting them to binary images making it easier to process and to work with.
* To prepare the dataset respective class folders were made and images to corresponding classes were inserted.
* Creating the Deep-Learning Model by adding required Layers for feature extraction and training the model on the created labelled dataset.

*School of Computer Engineering, KIIT, BBSR* Page 2

Chapter 2

Some Basic Concepts

What is image processing?

Image processing is a multidisciplinary field that encompasses various techniques and methodologies aimed at analyzing, manipulating, and enhancing digital images. It involves the utilization of algorithms and computational methods to extract meaningful information, improve visual quality, and perform various tasks on images captured by cameras or generated by other means.

At its core, image processing involves the following fundamental steps:

Image Acquisition: This step involves capturing or obtaining digital images through sensors, cameras, or other devices. These images may be in various formats such as grayscale, RGB (Red, Green, Blue), or other colour models.

Preprocessing: In this step, the acquired images undergo initial processing to enhance their quality and prepare them for subsequent analysis. This may include operations such as noise reduction, contrast enhancement, and image normalization.

Image Enhancement: Image enhancement techniques aim to improve the visual quality of images by emphasizing certain features or reducing unwanted artifacts. This may involve operations such as sharpening, blurring, or adjusting brightness and contrast.

Page 3

Image Restoration: Image restoration techniques are used to recover or reconstruct degraded or damaged images caused by factors such as noise, blur, or compression. These techniques aim to restore the original appearance of the image as much as possible.

Feature Extraction: Feature extraction involves identifying and extracting relevant information or features from images. This may include detecting edges, corners, shapes, textures, or other objects of interest within the image.

Image Segmentation: Image segmentation divides an image into meaningful regions or segments based on characteristics such as color, intensity, or texture. This facilitates further analysis and understanding of the content within the image.

Object Detection and Recognition: Object detection and recognition techniques involve identifying and classifying specific objects or patterns within images. This may include tasks such as face detection, object tracking, or character recognition.

Image Compression: Image compression techniques reduce the storage space required for images by eliminating redundant or irrelevant information while preserving essential visual quality. This is particularly important for efficient storage and transmission of images in various applications.

Image Understanding and Interpretation: Image understanding aims to interpret the content and context of images, enabling computers to extract semantic meaning or make decisions based on visual information. This may involve higher-level tasks such as scene understanding, image classification, or image-based reasoning.

Overall, image processing plays a crucial role in a wide range of applications, including medical imaging, satellite imagery analysis, surveillance, remote sensing, digital photography, and multimedia systems. Its continuous advancement and integration with other fields such as computer vision and artificial intelligence contribute significantly to technological innovation and development in various domains.

Page 4

What is Open-CV?

**OpenCV (Open Source Computer Vision Library)** is a widely-used open-source library designed to provide tools for computer vision, image processing, and machine learning. It supports a variety of programming languages such as Python, C++, Java, and MATLAB, making it versatile for developers across different platforms. Initially developed by Intel in 1999, OpenCV is now maintained by an active community, ensuring continuous updates and enhancements.

**Key Features of OpenCV**

1. **Image Processing:**
   * **Image I/O:** OpenCV can read and write images in various formats such as JPEG, PNG, TIFF, and BMP.
   * **Filtering and Enhancement:** It provides functions for blurring, sharpening, edge detection (e.g., Canny), and applying custom filters.
   * **Color Space Conversion:** Supports conversion between color spaces like RGB, HSV, LAB, and grayscale.
   * **Geometric Transformations:** Includes resizing, rotation, affine transformation, and perspective transformation.
2. **Video Processing:**
   * **Video Capture and Writing:** Supports capturing video from webcams or cameras and saving it to a file.
   * **Motion Detection:** Algorithms for optical flow and background subtraction are available.
   * **Object Tracking:** Includes tracking algorithms such as CamShift and MedianFlow.
3. **Object Detection and Recognition:**
   * **Face Detection:** Features pre-trained Haar cascades and deep learning-based detectors (DNN module).
   * **Object Detection:** Supports YOLO, SSD, and other deep learning-based models.
   * **Template Matching:** Enables locating a template image within a larger image.

Page 5

1. **Machine Learning:**
   * Includes tools for supervised and unsupervised learning, such as:
     + Support Vector Machines (SVM)
     + Decision Trees
     + K-means Clustering
     + Deep Learning (DNN module for TensorFlow and Caffe models)
2. **Image Segmentation:**
   * Implements algorithms for image thresholding, watershed segmentation, and contour detection.
3. **3D Vision:**
   * **Stereo Vision:** Provides tools for depth map generation from stereo image pairs.
   * **Camera Calibration:** Functions for calibrating cameras and correcting lens distortion.
   * **3D Reconstruction:** Supports algorithms for reconstructing 3D scenes from 2D images.
4. **Real-Time Applications:**
   * Optimized for real-time performance using multi-threading and GPU acceleration with CUDA and OpenCL.
5. **Integration with Deep Learning Frameworks:**
   * OpenCV supports models from popular frameworks like TensorFlow, PyTorch, and Caffe using its **DNN module**.

**Key Modules in OpenCV**

1. **Core Module:** Contains basic data structures such as matrices and image containers.
2. **Imgproc Module:** Handles image processing tasks like filtering, transformations, and edge detection.
3. **Highgui Module:** Deals with image and video I/O, including GUI for image display.
4. **Video Module:** Provides algorithms for motion analysis and tracking.
5. **Features2d Module:** Includes feature detection, description, and matching algorithms.

Page 6

**Advantages of OpenCV**

1. **Cross-Platform Support:**
   * Runs on Windows, Linux, macOS, iOS, and Android.
2. **Performance Optimization:**
   * Built with optimizations for modern CPUs and GPUs, making it suitable for real-time applications.
3. **Extensive Documentation:**
   * OpenCV offers detailed documentation, tutorials, and a supportive community, easing the learning curve.
4. **Scalability:**
   * Supports projects ranging from small-scale academic research to large-scale industrial applications.
5. **Wide Adoption:**
   * Used in diverse domains such as robotics, augmented reality, healthcare, and autonomous vehicles.

Page 7

What is Keras?

Keras is an open-source neural network library written in Python. It was developed with a focus on enabling fast experimentation with deep neural networks and providing a user-friendly interface for building and training various types of neural networks. Keras was designed to be modular, extensible, and easy to use, making it suitable for both beginners and experienced practitioners in the field of machine learning and artificial intelligence.

Key features of Keras include:

User-friendly API: Keras provides a simple and intuitive API that abstracts away the complexities of building and training neural networks. It allows developers to define neural network models using high-level building blocks, such as layers and models, without needing to understand the underlying implementation details.

Modularity: Keras follows a modular design, allowing users to easily assemble neural network models from reusable building blocks. It provides a wide range of pre-defined layers for common neural network architectures, as well as the flexibility to create custom layers tailored to specific requirements.

Flexibility: Keras supports a variety of backends for tensor computation, including TensorFlow, Microsoft Cognitive Toolkit (CNTK), and Theano. This enables users to choose the backend that best suits their needs while leveraging the high-level API provided by Keras.

Ease of extension: Keras is designed to be easily extendable, allowing developers to add custom functionality or integrate with other libraries seamlessly. It provides a clean and well-documented codebase, making it straightforward to contribute new features or extensions to the Keras ecosystem.

Page 8

Integration with TensorFlow: Since TensorFlow version 2.0, Keras has been integrated as the official high-level API for building and training neural networks within the TensorFlow framework. This integration provides users with the combined benefits of both TensorFlow's powerful backend and Keras' user-friendly interface.

Community and ecosystem: Keras has a large and active community of developers, researchers, and practitioners who contribute to its development, share best practices, and provide support through forums, documentation, and open-source contributions. This vibrant ecosystem has led to widespread adoption of Keras in various domains, including research, industry, and academia.

Overall, Keras is widely regarded as one of the most accessible and versatile libraries for deep learning, enabling rapid prototyping, experimentation, and deployment of neural network models for a wide range of applications, including computer vision, natural language processing, and reinforcement learning.

*School of Computer Engineering, KIIT, BBS* Page 9

*Bend Detection*

Chapter 3

Problem Statement / Requirement Specifications

To develop an automated system for cut and bent rebar bend calculation using image processing techniques, the project will focus on accurately assessing the condition and dimensions of rebar used in construction. This system aims to improve efficiency, reduce errors, and enhance safety in construction projects by providing precise measurements and evaluations of rebar bends and cuts. Below are the key objectives and project planning details:

**Key Objectives**

1. \*\*Rebar Bend and Cut Analysis\*\*: Develop algorithms to analyze images of rebar to detect and quantify bends. The system should accurately measure bends and lengths to ensure compliance with construction standards.

2. \*\*User Interface\*\*: Design a user-friendly interface that allows users to input images of rebar and view the evaluation results. The interface should provide clear visual feedback and actionable insights to assist in construction planning and quality control.

3. \*\*Performance Evaluation\*\*: Test and validate the accuracy and reliability of the image processing algorithms on a diverse dataset of rebar images. Compare the system's performance against traditional manual measurement methods.

**Project Planning**

1. \*\*Define Scope and Objectives\*\*: Clearly outline the goals of developing an automated system for evaluating rebar bends and cuts using image processing.

2. \*\*Gather Requirements\*\*: Collaborate with construction professionals to gather requirements and identify necessary resources, including hardware and software tools.

3. \*\*Research and Literature Review\*\*: Conduct a literature review to explore existing methodologies and algorithms related to image processing for rebar evaluation.

Page 10

4. \*\*Design System Architecture\*\*: Design the architecture of the system, defining components for image processing, feature extraction, classification, and user interface.

5. \*\*Develop Image Processing Algorithms\*\*: Implement algorithms for analyzing rebar images, detecting bends and cuts, and measuring angles and lengths.

6. \*\*Data Collection and Preparation\*\*: Collect a diverse dataset of rebar images and preprocess the data for analysis.

7. \*\*System Integration and Testing\*\*: Integrate components into a cohesive system and conduct testing to ensure accuracy and reliability.

8. \*\*Documentation\*\*: Prepare comprehensive documentation covering requirements, design, implementation, testing, and user guides.

9. \*\*Deployment and Evaluation\*\*: Deploy the system, evaluate performance, and gather feedback for improvement.

10. \*\*Maintenance and Support\*\*: Establish mechanisms for ongoing maintenance, updates, and user support.

11. \*\*Project Management\*\*: Define a schedule, allocate resources, track progress, and manage risks throughout the project lifecycle.

**System Design**

1. \*\*Overall Architecture\*\*: Design a modular architecture that encompasses all components of the system, including image acquisition, preprocessing, feature extraction, classification, and user interface.

2. \*\*Image Acquisition\*\*: Develop a contouring algorithm to crop the required columns and

3. \*\*Preprocessing\*\*: Implement preprocessing techniques to enhance image quality, including noise reduction and normalization.

4. \*\*Feature Extraction\*\*: Develop algorithms to extract relevant features from rebar images.

5. \*\*Classification\*\*: Design classifiers to analyze extracted features and classify rebar images based on bend and cut conditions.

Page 11

By addressing these objectives and planning details, the project aims to provide a robust and scalable solution for cut and bent rebar bend calculation using image processing, offering benefits such as improved accuracy, reduced labour costs, and enhanced construction safety.

*School of Computer Engineering, KIIT, BBSR* Page 12

*Bend Detection*

Chapter 4

Implementation

**4.1 Methodology or Proposal**

**Introduction:**

The objective of this project is to develop a reliable model that can accurately calculate rebar bend counts This task is crucial for automated inspection and quality control in the construction industry.

**Data Collection and Preprocessing:**

Initially, a substantial amount clear and sharp images of cut and bent rebar, must be collected. These images should represent a variety of conditions, lighting scenarios, and orientations to ensure the model's robustness. Image processing techniques like Gray scaling, and applying blurring along with converting the image techniques such as rotation, flipping, and scaling will be employed to enhance dataset diversity and prevent overfitting.

Page 13

**4.2 Testing/Verification Plan**

1. **Data Testing:**

- Verify the integrity and quality of the collected dataset by inspecting sample images and labels.

- Randomly select subsets of data for manual inspection to ensure a diverse range of rebar conditions.

- Perform statistical analysis to identify any biases or anomalies in the dataset.

2. **Preprocessing Testing:**

- Validate preprocessing steps such as resizing, normalization, and augmentation by visually inspecting transformed images.

- Ensure data augmentation techniques preserve the semantic content of images while introducing variability.

3. **Model Architecture Testing:**

- Test the proposed CNN architecture on a small dataset subset to ensure proper implementation and functionality.

- Validate the model's architecture by inspecting the summary of layers and parameters.

- Conduct exploratory experiments to evaluate the model's capacity to learn relevant features from input images.

4. **Training Testing:**

- Train the model on a dataset subset and monitor training progress using metrics such as loss and accuracy.

- Verify that the model converges to a stable state and performs reasonably on training data.

- Perform hyperparameter tuning experiments to identify optimal values for learning rate, dropout rate, etc.

5. **Evaluation Testing:**

- Evaluate the trained model on a separate validation set to assess generalization performance.

- Calculate standard classification metrics such as accuracy, precision, recall, and F1-score.

- Generate visualizations like confusion matrices and ROC curves to analyze model behaviour across different classes.

Page 14

6. **Deployment Testing:**

- Deploy the trained model in a test environment to evaluate performance in real-world scenarios.

- Verify that the deployed model meets specified performance criteria and provides accurate predictions.

- Conduct stress testing by simulating high loads or unexpected inputs to assess robustness and scalability.

7. **Documentation and Reporting:**

- Document the testing process, including test cases, methodologies, and outcomes.

- Provide clear and comprehensive reports summarizing testing results, highlighting issues or areas for improvement.

- Communicate findings to stakeholders and obtain feedback for further refinement.

8. **Iterative Testing:**

- Conduct iterative testing and validation cycles to address identified issues or shortcomings.

- Incorporate feedback from stakeholders and domain experts to continuously improve model performance and reliability.

**4.4 Model Deployment**

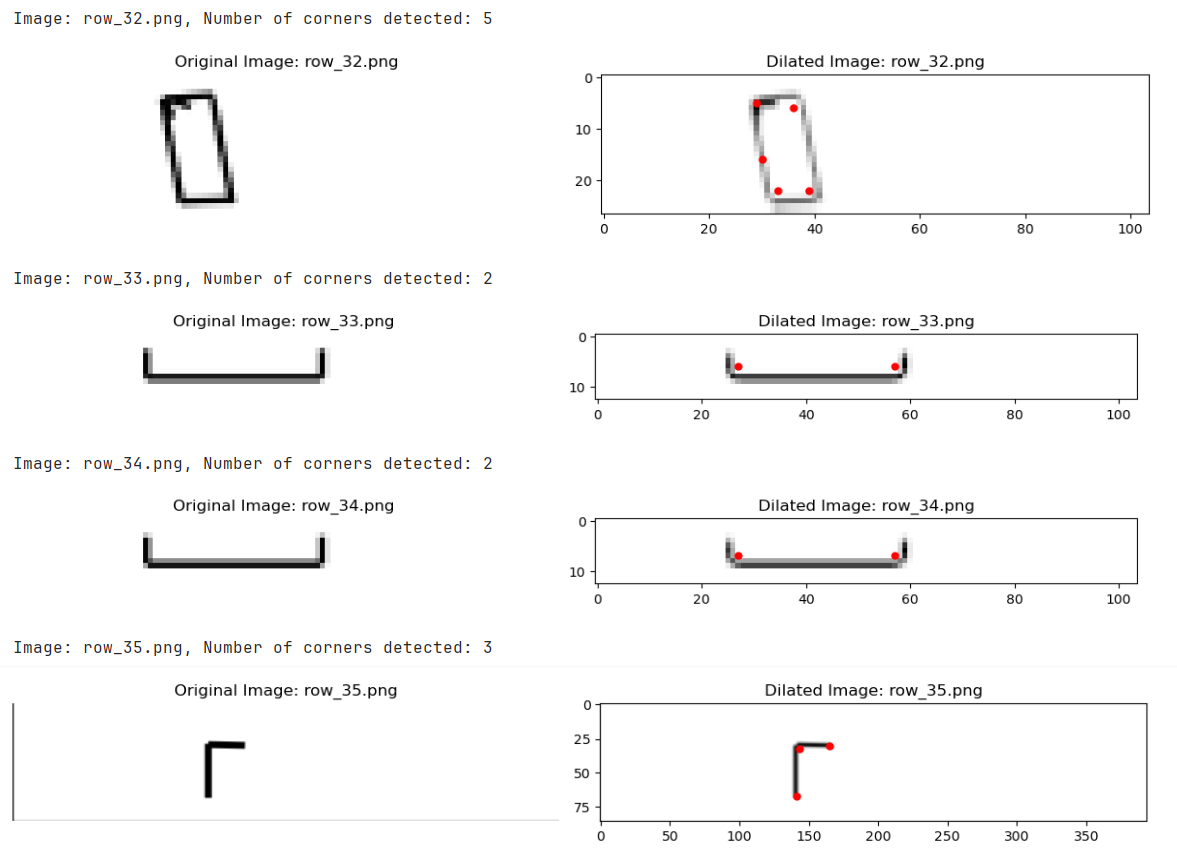
After training is complete and performance is satisfactory, the model will be deployed in production environments for real-time inference. The deployment process may involve converting the trained model into a lightweight format suitable for edge devices or cloud-based services. Retraining techniques and continuous monitoring will be applied to sustain model performance over time.

Page 15

**4.3 Result Screenshots**

**A screenshot of a computer

Description automatically generated**

****

Page 16

**A screenshot of a computer

Description automatically generated**

Page 17

**Guidelines for Quality Assurance**

**Assurance of Data Quality:**

- Ensure the dataset is well-labelled, representative, and diverse.

- Perform preprocessing operations such as normalization, augmentation, and validation to maintain data quality.

**Design and Implementation:**

- Verify that the chosen image processing techniques are suitable for the task of bend calculation.

- Ensure best practices like edge detection, segmentation, and feature extraction are effectively implemented.

**Process Monitoring:**

- Monitor the accuracy and reliability of the image processing pipeline.

- Implement validation checks to ensure consistent performance and prevent errors.

**Evaluation Metrics:**

- Use standard evaluation metrics to validate the effectiveness of the image processing techniques.

- Create visualizations to assess the system's ability to calculate and count bends

**Parameter Optimization:**

- Document the process of optimizing parameters, including chosen values and their impact on performance.

- Use systematic methods to ensure optimal parameter settings.

**System Deployment:**

- Ensure the deployed system meets necessary performance requirements in real-world settings.

- Continuously monitor the system's performance and apply updates as needed.

- Verify that the deployed system complies with all relevant laws and regulations.

**Record-keeping and Reporting:**

- Maintain detailed records of all stages of the system development process, including data collection, preprocessing, evaluation, and deployment.

- Provide clear and comprehensive reports outlining the system's performance, key findings, and recommendations for improvement.

- Gather feedback from subject matter experts and stakeholders to identify areas for enhancement.

- Establish a feedback loop to incorporate lessons learned and improve the quality of the system and related processes over time.

Page 18

Chapter 5

Standards Adopted

**5.1 Design Standards**

**1. Adherence to Engineering Standards:**

1. Ensure compliance with relevant engineering standards pertaining to bend count calculations.
2. Consult with experts and industry professionals to ensure compliance with established design standards.

**2. Utilization of Image Processing Techniques:**

1. Employ suitable image processing algorithms and techniques, such as edge detection, segmentation, and feature extraction, for effective rebar image analysis.

**3. Bend Calculation Accuracy:**

1. Aim for high levels of accuracy and precision in bend calculation algorithms to ensure dependable evaluation outcomes.
2. Utilize mathematical models and algorithms that accurately represent the physical properties of bends, such as the bending moment and stress-strain relationship.
3. Implement techniques for bend angle, length, and depth measurement to provide comprehensive bend information.

Page 19

**4. Data Integrity:**

1. Implement measures to uphold data integrity throughout the image processing pipeline, encompassing preprocessing stages and post-processing analysis.
2. Validate the accuracy and reliability of the image processing algorithms and bend calculation techniques using appropriate metrics.

**5. Validation and Verification:**

1. Validate image processing algorithms and bend calculation techniques against ground truth data and verify their performance using appropriate metrics.
2. Collaborate with experts and industry professionals to validate the accuracy and reliability of the bend calculation evaluation process.

Page 20

**5.2 Coding Standards**

Adherence to best practices in programming and software development remains crucial for maintaining code quality and readability. Here are coding standards for implementing image processing algorithms:

***Coding Standards:***

**1. Modularity** Design image processing algorithms in a modular manner to enable code reuse and enhance maintainability.

**2. Descriptive Naming:** Utilize descriptive variable and function names to improve code comprehension and readability.

**3. Comments and Documentation:** Incorporate comments to elucidate the purpose and functionality of key algorithmic steps, and document the code for future reference.

**4. Optimization:** Optimize code for efficiency and performance, considering computational complexity and memory usage.

**5. Error Handling:** Implement robust error handling mechanisms to address unexpected scenarios and ensure the reliability of the image processing pipeline.

**5.3 Testing Standards**

Testing standards play a pivotal role in validating the accuracy and reliability of image processing algorithms utilized in bend count evaluation. Adherence to recognized testing standards is crucial to ensure comprehensive validation and verification of the image processing pipeline.

***Testing Standards:***

**1. Accuracy Testing:** Validate the accuracy of image processing algorithms by comparing results against ground truth data or manual annotations.

**2. Robustness Testing:** Assess the robustness of image processing algorithms, image quality fluctuations. Page 21

Chapter 6

Conclusion and Future Scope

6.1 Conclusion:

The Cut and Bent (CAB) process is crucial for producing various components. However, manual inspection of bends for quality control is time-consuming, prone to human error, and can lead to production bottlenecks. Automating bend detection and measurement offers significant advantages in efficiency, accuracy, and cost reduction.  
  
This project aims to develop a robust and scalable solution for CAB bend detection and measurement using computer vision and deep learning techniques. By leveraging image processing algorithms and machine learning models, we can achieve real-time bend detection, classification, and measurement, enabling automated quality control and process optimization in the steel industry.

In this report, we explored the development of a computer vision model used to extract and store number of bends from a bar bending schedule which is a crucial data used by industrial manufacturing companies.

The process began with image preprocessing, where we implemented custom image preprocessing functions to enhance the intricate features of the cut and bent rebar shapes. These preprocessing steps included contour detection and adaptive thresholding to obtain and store available images in a format that can be easily processed making the code efficient.

Page 22

6.2 Future Scope:

**1.**Predictive maintenance: The model can be used to predict potential bend failures in products before they occur, enabling proactive maintenance and preventing downtime.

**2.**Quality control and optimization: By accurately measuring bends, the model can help identify and address quality issues, leading to improved product quality and reduced defect rates.

**3.**Compliance and regulatory compliance: The model can be used to verify compliance with industry standards and regulations, ensuring that products meet safety and performance requirements.

**4.**Innovative design and manufacturing processes: The model can help optimize design and manufacturing processes, reducing waste and improving efficiency.

**5.** Enhanced product safety: By detecting and quantifying bends, the model can help identify potential safety risks and ensure that products meet safety standards.

**6.**Advanced materials and components: The model can be applied to new materials and components, enabling more accurate and efficient inspection and design.

The development of a computer vision model for bend detection and calculation has the potential to revolutionize manufacturing processes, enabling more efficient and accurate inspection of products. This model leverages advanced image processing techniques and machine learning algorithms to identify and quantify bends in various materials, such as metal, plastic, or composites.

Page 23

Key Features:

**1.** Real-time bend detection: The model can process high-resolution images or videos in real-time, enabling immediate identification of bends in products during manufacturing or assembly.

**2.**Accurate bend measurement: The model utilizes advanced image processing algorithms to accurately measure the angle, length, and depth of bends, providing valuable data for quality control and optimization.

**3.** Customizable for various materials: The model can be trained to recognize and quantify bends in different materials, ensuring its applicability across various manufacturing applications.

**4.** Integration with existing systems: The model can be integrated with existing manufacturing systems, such as automated inspection machines (AIMs) or robotic arms, to enhance overall product quality and efficiency.

*School of Computer Engineering, KIIT, BBSR* Page 24

*Bend Detection*

***References***

<https://qaqcinconstruction.com/bar-bending-schedule-rebars-bbs/>

<https://www.ijcrt.org/papers/IJCRT2104593.pdf>

<https://ijarcce.com/papers/automation-of-bar-bending-schedule-software-for-building/>

*School of Computer Engineering, KIIT, BBSR* Page 25

*Bend Detection*

**CONTRIBUTION REPORT:**

**CAB Rebar Bend Calculation**

NIKHIL CHOUDHARY-21052166

**Abstract:**In the construction industry, ensuring the structural integrity and safety of rebar is crucial. Manual inspection of rebar bends and cuts is often time-consuming, subjective, and prone to errors. To address these challenges, we propose an Automated Rebar Bend and Cut Inspection System using Image Processing. This project aims to automate the inspection process through computer vision and machine learning, enhancing efficiency and accuracy in evaluating rebar conditions. By providing precise measurements and assessments, the system will contribute to improved construction quality and safety.

**Individual Contribution and Findings:**The source code for the project and the entire project report was solely written and developed by Nikhil Choudhary (21052166).

**Individual Contribution to Project Report Preparation:**All sections of the project report, including the introduction, basic concepts, literature review, problem statement, requirement specifications, project planning, project analysis, implementation, standards adopted, and future scope, were entirely authored by Nikhil Choudhary (21052166).

**Individual Contribution for Project Presentation and Demonstration:**The entire project, including the presentation and demonstration of all chapters, will be conducted by Nikhil Choudhary (21052166).

Full Signature of Supervisor: Full signature of the student:

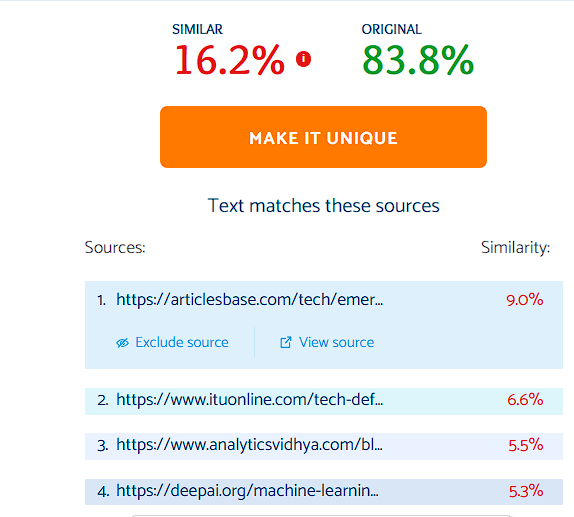
……………………………. ……………………

*School of Computer Engineering, KIIT, BBSR* Page 26

PLAGIARISM REPORT

**(This report is mandatory for all the projects and plagiarism**

**must be below 25%)**



*School of Computer Engineering, KIIT, BBSR* Page 27