WELDING DEFECT PREDICTION USING AI

BY STAVAN JINTURKAR

1.Introduction

1. Introduction and Problem Statement

Welding is a critical process in manufacturing and construction, used to join metal components permanently. Ensuring the quality of welds is essential, as welding defects can compromise the structural integrity of products. Traditional methods of weld inspection rely heavily on human inspectors, which can be time-consuming and prone to errors. By leveraging Artificial Intelligence (AI), we aim to automate the detection of welding defects to improve both accuracy and efficiency in manufacturing.

2. What are Welding Defects?

Welding defects are imperfections in a welded joint that compromise the joint's strength, durability, and performance. These defects can result from improper welding techniques, incorrect material selection, or environmental factors such as temperature and humidity during welding. Identifying and correcting these defects is crucial to ensure the safety and longevity of welded structures.

3. Types of Welding Defects and their Characteristics

Here are the common types of welding defects and their defining characteristics:

1. **Porosity:**

- **Description:** Porosity refers to small cavities or holes that appear on or inside the welded metal. These occur due to trapped gases during solidification.
- **Characteristics:**
  - Visible as small, scattered holes on the surface or within the weld.
  - Weakens the welded joint.
  - Caused by improper shielding gas, moisture, or contamination in the welding material.

2. **Cracks:**
   - **Description:** Cracks are separations in the welded joint that can occur at the surface, subsurface, or root of the weld. Cracks are highly undesirable as they compromise the weld's strength.
   - **Characteristics:**
     - Can be longitudinal (parallel to the weld), transverse (perpendicular to the weld), or root cracks (at the bottom of the weld).
     - Visible to the naked eye or detectable with non-destructive testing.
     - Caused by rapid cooling, improper preheating, or incorrect weld technique.
3. **Undercut:**
   - **Description:** An undercut occurs when the base metal at the edge of the weld is melted away, forming a groove along the weld bead.

- **Characteristics:**
  - Leaves a depression or groove along the welded joint.
  - Reduces the thickness of the base material and weakens the weld.
  - Caused by excessive welding heat or incorrect angle of the welding torch.

4. **Incomplete Penetration:**
   - **Description:** This occurs when the weld metal does not fully penetrate the joint, leaving unfilled sections.
   - **Characteristics:**
     - Visible as a gap or lack of weld at the root of the joint.
     - Compromises the strength of the weld.
     - Caused by insufficient heat or improper welding technique.
5. **Incomplete Fusion:**
   - **Description:** Incomplete fusion happens when the weld metal does not fully fuse with the base metal, leaving portions unjoined.
   - **Characteristics:**
     - Visible as a gap or lack of bonding between the weld metal and the base metal.
     - Weakens the joint and can lead to structural failure.
     - Caused by inadequate cleaning of the base metal or incorrect welding parameters.

6. **Slag Inclusions:**
    - **Description:** Slag inclusions occur when non-metallic particles become trapped in the weld metal.
    - **Characteristics:**
        - Appears as irregularities or foreign material within the weld.
        - Reduces the weld's strength and may cause cracking.
        - Caused by improper cleaning between weld passes or use of low-quality filler material.

4. Problems Faced in Manual Welding Defect Inspection

Manual inspection of welding defects on assembly lines presents several challenges:

- **Time-Consuming:** Inspecting each weld individually is slow and causes bottlenecks in production, affecting the overall efficiency of the assembly line.
- **High Labor Requirements:** Skilled inspectors are needed to identify defects, increasing operational costs. Scaling this process becomes challenging with labor shortages.
- **Human Error:** Inspections rely heavily on human judgment, which can be inconsistent due to factors like fatigue, distractions, and varying experience levels.

- **Difficulty Detecting Internal Defects:** Surface-level inspections can miss internal defects such as pores or incomplete penetration, which are critical to the strength of the weld.
- **Fatigue:** Repetitive inspections lead to physical and mental fatigue, increasing the likelihood of missed defects and lowering the quality of inspections over time.
- **Inconsistent Reporting:** Manual documentation of defects is often inconsistent, making it difficult to track issues over time and implement effective quality control measures.

5. Labor Requirements for Manual Welding Defect Inspection

- **Skilled Weld Inspectors:** Highly trained personnel are required to visually inspect welds and identify common defects.
- **NDT Technicians:** For detecting internal defects, non-destructive testing (NDT) technicians are needed, adding to labor costs.
- **Multiple Inspectors:** Large-scale assembly lines often require multiple inspectors to handle the workload, further increasing labor demands.
- **Supervisory Staff:** Quality control personnel are necessary to oversee inspections, maintain consistency, and ensure accurate defect reporting.

6. Importance of Welding Error Detection using AI

1. **Increased Accuracy and Consistency:**

    - AI detects even subtle defects with greater precision and consistency than human inspectors, ensuring higher product quality.
2. **Enhanced Speed and Efficiency:**
    - AI enables real-time analysis, speeding up the inspection process and preventing production bottlenecks.
3. **Cost Reduction:**
    - By automating inspections, AI reduces the need for skilled labor, minimizes rework, and prevents costly recalls.
4. **Real-Time Monitoring and Feedback:**
    - AI provides instant feedback, allowing for immediate corrective actions and reducing waste on the assembly line.
5. **Scalability:**
    - AI can easily scale across multiple production lines, handling increased workloads without additional labor costs.
6. **Detection of Complex and Internal Defects:**
    - AI detects both surface and internal defects, improving the overall reliability and structural integrity of welded products.
7. **Data-Driven Insights:**
    - AI systems collect valuable data, enabling manufacturers to optimize processes and reduce defect rates through continuous improvement.

2. Design Steps and Methodology

1. Data Collection

In developing an AI-based welding defect detection model, the most critical aspect is gathering diverse and high-quality data. This involves multiple stages of data collection to ensure that the model can accurately detect defects across various welding scenarios and equipment types.

1. **Primary-Level Data (Zoomed-In Images)**
   - **Objective:** The first step is to collect zoomed-in images that focus on the welded areas. These images are captured at high resolution to highlight various types of welding defects such as cracks, pores, and bad welding spots.
   - **Data Characteristics:** These images provide a detailed view of the welding defects, making it easier to identify their texture, shape, and other key visual features. This is essential for understanding how defects appear up close.
   - **Importance:** Zoomed-in data helps the model recognize how defects form and look at a microscopic level. For instance, cracks and pores might be small and hard to detect in full-stand images, but they are crucial indicators of structural issues. By training the model on zoomed-in data, we improve its ability to detect fine details and subtle irregularities that could be missed otherwise.
2. **Whole Equipment Data (Full-Stand Images)**
   - **Objective:** After collecting zoomed-in images, we need to scale up by gathering images of entire mechanical models. These full-stand images showcase the welding areas in the context of the entire piece of equipment, allowing the AI to learn how defects manifest in larger assemblies.

- **Data Characteristics:** These images should cover different types of equipment used across industries. The data should reflect various materials, welding techniques, and equipment designs. By doing so, we create a robust dataset that improves the model's generalization capabilities.
- **Importance:** Whole equipment images help ensure that the AI system can detect defects not just in isolated zoomed-in sections but also in a larger assembly. This is crucial when inspecting welds on large components like pipes, beams, or machinery. By exposing the AI to full-stand data, we make the model versatile and capable of identifying defects regardless of the scale or type of equipment.

3. **Equipment-Specific Data**
    - **Objective:** For practical application, we require data specific to the particular equipment manufactured by a given company. This means collecting a large volume of images for the exact machinery or components the AI model will inspect in real-world scenarios.
    - **Data Characteristics:** The equipment-specific data will focus on capturing the welds of that particular piece of machinery or structure. We will collect images at multiple angles, lighting conditions, and with both zoomed-in and full-stand perspectives.
    - **Importance:** Every company has its own equipment design, welding standards, and techniques. By gathering data tailored to specific equipment, the model will be able to specialize in detecting defects for those particular structures. This ensures the AI can deliver highly accurate defect

detection for the equipment it will be applied to in practice. Additionally, expanding this dataset as new equipment types are introduced ensures the model remains adaptable and scalable for future use.

4. **Image Labeling**
   - **Objective:** After collecting the images, each one must be carefully labeled to teach the model how to distinguish between different types of welds and defects. Accurate labeling is critical for the AI to learn and generalize from the training data.
   - **Labeling Method:** Using the YOLO (You Only Look Once) format, each image is labeled with bounding boxes that highlight specific defects or good welds. Each box corresponds to one of four predefined categories:
     - **Good Welding:** Areas where the weld is solid, well-formed, and meets the necessary standards.
     - **Bad Welding:** Areas where the weld is incomplete, uneven, or exhibits visible issues such as undercuts or spatter.
     - **Cracks:** Visible separations in the welded material that may compromise its strength and integrity.
     - **Pores:** Small cavities or holes in the weld caused by trapped gases during the welding process.
   - **Importance:** Proper labeling is fundamental for supervised learning. The model needs to be trained with precise examples of each defect type so it can correctly classify them in new, unseen data. The use of the YOLO format, which

enables fast and accurate real-time object detection, makes it ideal for this task. Additionally, labeling both zoomed-in and full-stand images with these four categories ensures that the model is versatile enough to detect defects at both the micro and macro levels.

5. **Database Growth and Scalability**
    - **Objective:** As we collect more data across different types of equipment and welding scenarios, the database will continuously grow, allowing the AI model to improve in accuracy and versatility.
    - **Importance:** Over time, the system will become capable of detecting defects across various industries and equipment types. By gradually expanding the dataset, we can make the AI model applicable to any type of equipment in the future, enabling it to adapt to new equipment designs, materials, and welding standards.

2. Preprocessing

Preprocessing is a crucial step in preparing the collected data for training the AI model. It ensures that the images are standardized, enhanced, and ready for optimal learning by the model. The following preprocessing techniques are applied:

1. **Conversion to Grayscale**
    - **Objective:** The first step is to convert all images from color to grayscale. Welding defect detection relies more on the texture, edges, and contrast of the image than on color information.

- **Importance:** Grayscale images reduce the complexity of the data, as they contain fewer channels (1 channel for grayscale vs. 3 for RGB). This also helps speed up processing while maintaining critical visual features like cracks, pores, and weld quality.

2. **Image Resizing**
    - **Objective:** To standardize the image size across the dataset, all images are resized to a fixed resolution (e.g., 416x416 for YOLO). This ensures that the input dimensions to the model are consistent, which is necessary for efficient processing and accurate detection.
    - **Importance:** Resizing helps the model handle varying image sizes and scales during training and prediction while preserving important details of the welds.

3. **Data Augmentation**
    - **Objective:** Data augmentation is applied to increase the diversity of the training set without collecting more images. Various augmentation techniques are used:
        - **Rotation:** Random rotations (e.g., ±10° or ±30°) are applied to simulate different camera angles.
        - **Contrast Adjustment:** Contrast is varied to simulate different lighting conditions in the images, enhancing the model's ability to detect defects in low or high-contrast environments.
        - **Flipping and Cropping:** Horizontal and vertical flips as well as random crops are applied to diversify the visual appearance of welds and defects.

        - **Scaling and Zooming:** Scaling the images in and out slightly mimics different distances from the object, allowing the model to learn from different perspectives.
    - **Importance:** These augmentations artificially expand the dataset, improving the model's generalization ability and robustness to different environments, lighting conditions, and angles.
4. **Noise Reduction**
    - **Objective:** To reduce unwanted noise in the images, which can interfere with defect detection. Noise reduction techniques like Gaussian smoothing are applied to make the edges and textures of the defects more visible.
    - **Importance:** Noise in the image can distort the edges and textures, which are critical features for detecting weld defects. By reducing noise, we improve the clarity of the defects, helping the model to focus on relevant details.
5. **Edge Detection**
    - **Objective:** Edge detection techniques (such as the Canny edge detector) are used to highlight the boundaries and edges of defects. This helps in distinguishing cracks, pores, and weld boundaries.
    - **Importance:** Welding defects often manifest as irregular edges or boundaries, such as cracks and uneven welding lines. Detecting these edges enhances the model's ability to recognize such anomalies during training.
6. **Texture Detection**

- **Objective:** Texture analysis is performed to identify surface irregularities in the welds, such as pores, roughness, or spatter. Algorithms like Local Binary Patterns (LBP) are applied to capture texture patterns.
- **Importance:** Texture is a vital feature when differentiating between good and bad welds. By detecting and analyzing the texture of the weld surface, the model can better identify defects such as pores, spatter, and rough surfaces.

3. Model Selection

Why YOLO for Object Detection?

We plan to use **YOLO (You Only Look Once)** for its speed and efficiency in object detection, which makes it ideal for real-time welding defect detection. YOLO can detect multiple defects, such as good/bad welds, cracks, and pores, within a single image, which is crucial for comprehensive quality control.

- **Real-Time Detection:** YOLO processes the entire image in one go, allowing for quick identification of defects, which is important in manufacturing lines.
- **Multi-Object Detection:** Each image may have multiple defects, and YOLO can handle detecting all of them simultaneously, which saves time and increases accuracy.

YOLO Architecture and Adaptation for Multi-Label Classification

- **Image Input:** The model will take in standardized images, resized to a uniform resolution (e.g., 416x416), making it adaptable to various scales of welding images.

- **Bounding Box Predictions:** YOLO divides the image into grids and will predict bounding boxes for each grid cell to localize defects like cracks or bad welds.
- **Multi-Label Classification:** The model will be adapted to classify and label four key types of welding conditions: good welds, bad welds, cracks, and pores.
- **Customization for Welding Data:** By labeling our images with these categories and training YOLO, we aim to allow the model to accurately detect different defects across zoomed-in and full-stand images.

3. Model Training & Evaluation

Libraries to Be Used

The following Python libraries will be utilized for data preprocessing, model training, and evaluation:

1. **TensorFlow**
2. **PyTorch**
3. **OpenCV**
4. **Albumentations**
5. **Matplotlib/Seaborn**
6. **Scikit-learn**
7. **Pandas**
8. **Numpy**
9. **YOLOv5/Ultralytics**

Training:

- **Training Process:** The model will be trained on a dataset of welding images, where each image is labeled with bounding boxes indicating good welds, bad welds, cracks, and pores. The YOLO architecture will be used, with input images resized and preprocessed as described earlier.
- **Number of Epochs:** We plan to train the model for **50–100 epochs** to ensure it learns both simple and complex patterns in the data. We may adjust the number of epochs based on the model's performance during validation.
- **Batch Size:** A batch size of **16 or 32** will be used, depending on the computational resources available. Larger batch sizes can speed up training but require more memory, while smaller ones may provide more frequent updates to the model weights.
- **Hyperparameter Tuning:** We will fine-tune the following parameters:
    - **Learning Rate**
    - **Confidence Threshold:** This will be tuned to minimize false positives or false negatives.
    - **Anchor Boxes:** The anchor boxes for bounding box predictions will be optimized based on the size of detected defects in the images.

Evaluation Metrics

To assess the model's performance, we will use the following evaluation metrics:

- **Precision**

- **Recall:** The proportion of true positives among all actual positives (defects present in the image). This metric helps measure how well the model detects all defects.
- **F1-Score**
- **Mean Average Precision (mAP**

Validation Techniques:

  - We will use an 80/20 **train-validation split** to ensure the model is evaluated on unseen data during training.
  - **K-fold cross-validation** may also be applied to minimize variance in results and ensure that the model generalizes well across different splits of the data.

Challenges

- **Overfitting:** With a small dataset or overly complex model, overfitting may occur. To tackle this, we will apply data augmentation, use dropout layers, and adopt early stopping during training.
- **Imbalanced Data:** In welding images, there might be an imbalance between good welds and defects (bad welds, cracks, pores). This could cause the model to be biased toward detecting good welds.
- **Noise in Data:** Welding images may contain noise or visual distortions that affect the accuracy of defect detection.

Feature Expansion: Detecting Missing Parts in Structures

In addition to detecting welding defects, the AI model will also be extended to identify missing parts in structures. This feature is critical for manufacturers who want to ensure that all components of a mechanical structure are assembled correctly before it goes into operation.

Data Collection:

Manufacturers will provide images of fully assembled structures. These images will serve as the reference standard. Additionally, images of the same structures with missing parts will be collected or synthetically generated to train the model in identifying incomplete assemblies.

Training Process:

The model will be trained to learn the visual characteristics of a complete structure. It will be fine-tuned to detect differences between the provided image and the standard reference image, highlighting any missing parts or abnormalities in the assembly.

- **Multi-Label Classification:** The model will classify parts as "present" or "missing," similar to how it classifies defects like cracks or bad welds.
- **Object Detection Techniques:** YOLO can be used to detect missing parts in the same way it detects welding defects, by predicting bounding boxes around missing components.

Challenges:

- **Variety of Structures:** Different structures from different manufacturers may require a more generalized model.
- **Labeling Data:** Each image of the structure will need to be accurately labeled.
- **Partial Obstruction**

4. App Design Overview

User Interface (UI)

The app will provide an intuitive interface for manufacturers to upload images and receive defect detection results. Below is a basic flow of how a user will interact with the app:

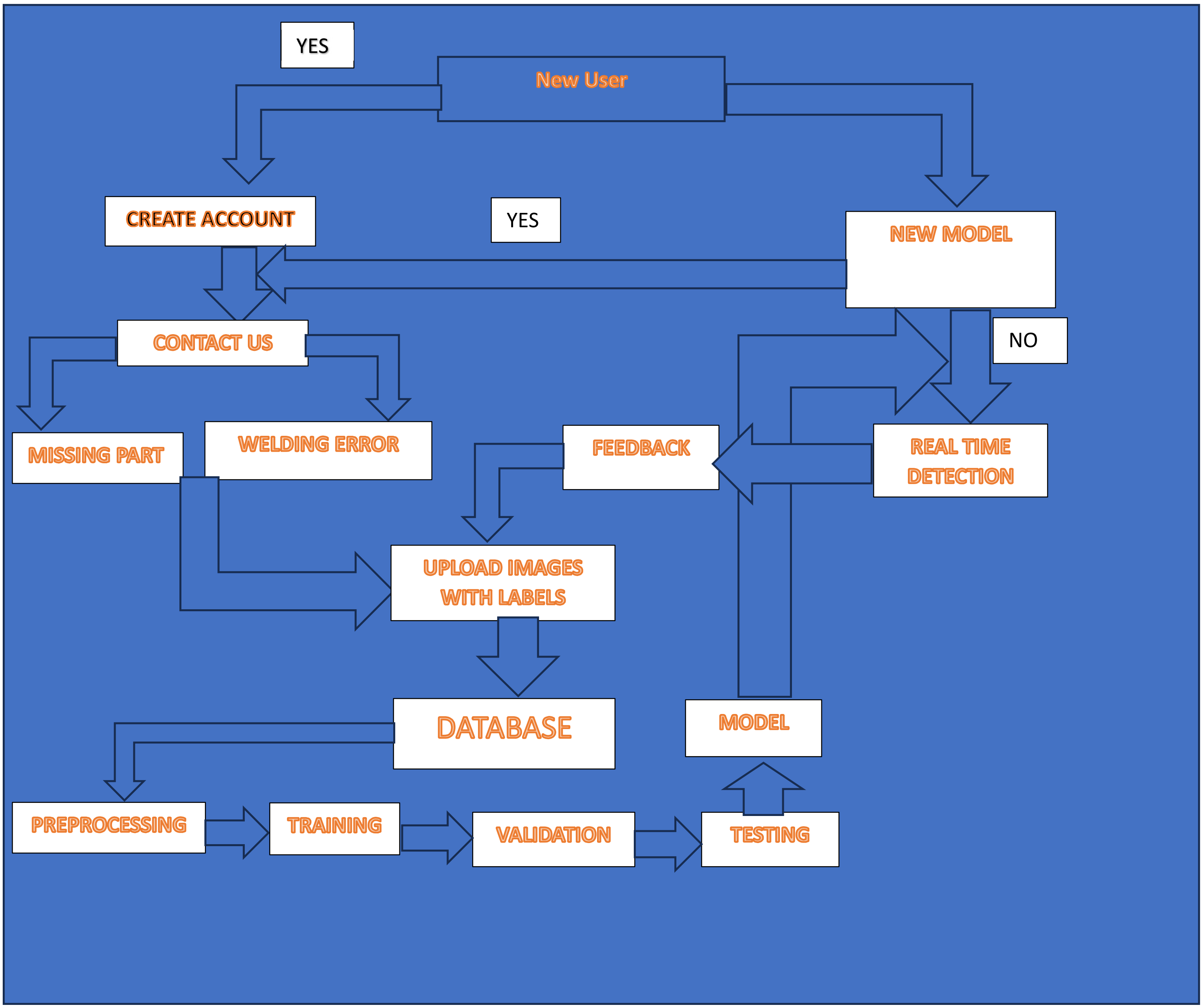
1. **Sign-Up/Login Page:**
    - New manufacturers will register by providing their company details and contact information.
    - After signing up, they will be connected with an employee to understand their specific requirements (e.g., welding error detection or missing part detection).
2. **Image Upload:**
    - Once the manufacturer is registered, they will be able to upload images of the equipment they want to inspect for welding errors or missing parts.
    - The upload page will have two options:
        - **Welding Error Detection**
        - **Missing Part Detection**
3. **Real-Time Detection:**
    - After uploading images, the model will process them, and results will be displayed with highlighted areas showing good welds, bad welds, cracks, pores, or missing parts, depending on the selected option.
4. **Results & Feedback:**
    - Users will be able to review the detected labels for each uploaded image.

- If the user finds any false predictions (e.g., missing or incorrect labels), they will be able to manually correct them by drawing bounding boxes and adding labels.
- The corrected data will be sent back to improve the model's accuracy.

Functionalities

- **Upload Image Functionality:**
  - Users will have the ability to upload images directly from their device into the app.
- **Real-Time Detection:**
  - Once images are uploaded, the model will run in real-time and display highlighted areas with defects or missing parts in the image.
  - This functionality will provide immediate feedback to the user.
- **Result Display with Defect Classification:**
  - After processing the images, the app will show a visual overlay of the detected areas, labeled as good welds, bad welds, cracks, pores, or missing parts.
  - Each image will have the ability to zoom in for detailed inspection.
- **User Feedback & Corrections:**
  - Users will have the option to correct false detections by editing labels and drawing bounding boxes.

- This feedback loop will help retrain the model with new, more accurate data.



5. Results and Future Enhancements

Review of Predictions and Changes:

The current version of the model processes images and detects welding errors such as:

- Good Welds
- Bad Welds
- Cracks
- Pores

Users can upload images and the model will predict defects, allowing the user to review the results. If the predicted labels are incorrect or incomplete, users have the ability to adjust bounding boxes and re-label parts of the image. This feedback mechanism helps improve the model's predictions over time.

Sample Images with Detected Welding Errors:

(You can insert sample images here with YOLO predictions highlighting defects such as good welds, bad welds, cracks, and pores.)

Future Work:

- **Improving the Model:**
  - **More Data:** Gathering more images of different welding types and defects across a variety of equipment will improve the robustness of the model.
  - **Better Preprocessing:** Enhance the preprocessing pipeline by experimenting with different augmentation techniques, filtering methods, and noise reduction strategies.

    - **Model Fine-Tuning:** Conduct further hyperparameter tuning and experiment with other architectures like Faster R-CNN or SSD to see if they perform better than YOLO for welding defect detection.
- **App Performance Enhancements:**
    - **Real-Time Video Detection:** Expanding the model to process real-time video data from assembly lines, allowing continuous monitoring of weld quality.
    - **Integration with Manufacturing Systems:** Incorporate the app into existing manufacturing systems for seamless data flow, enabling direct feedback to engineers or automated systems when defects are detected.
    - **Edge Deployment:** Deploy the model on edge devices (e.g., cameras on-site) for faster detection without the need for cloud-based processing.

---

6. Conclusion and References

Conclusion:

This AI-driven welding defect detection model has demonstrats significant potential for automating the inspection process. By leveraging object detection methods, we can now efficiently identify issues such as cracks, bad welds, and pores in real-time, reducing human labor and improving accuracy. Additionally, the app's capability to identify missing parts in structures makes it a versatile tool for manufacturers.

expansion, it could become a key part of industrial quality control workflows, improving productivity and reducing errors in manufacturing lines.

 Welding Defects: Causes, Prevention, and Repair (2010). American Welding Society. AWS D1.1 Structural Welding Code.

 ASM Handbook, Volume 6: Welding, Brazing, and Soldering (1993). ASM International.

 Lancaster, J. F. (1993). **The Physics of Welding**. Pergamon Press.

 Kou, S. (2003). **Welding Metallurgy**. Wiley-Interscience.

 Redmon, J., Farhadi, A. (2018). YOLOv3: An Incremental Improvement. [arXiv preprint arXiv:1804.02767].

 Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. NIPS.

 He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep Residual Learning for Image Recognition. CVPR.

 Goodfellow, I., Bengio, Y., Courville, A. (2016). **Deep Learning**. MIT Press.

 Russakovsky, O., et al. (2015). **ImageNet Large Scale Visual Recognition Challenge**. International Journal of Computer Vision.