

AI-Powered Defect Detection in Manufacturing

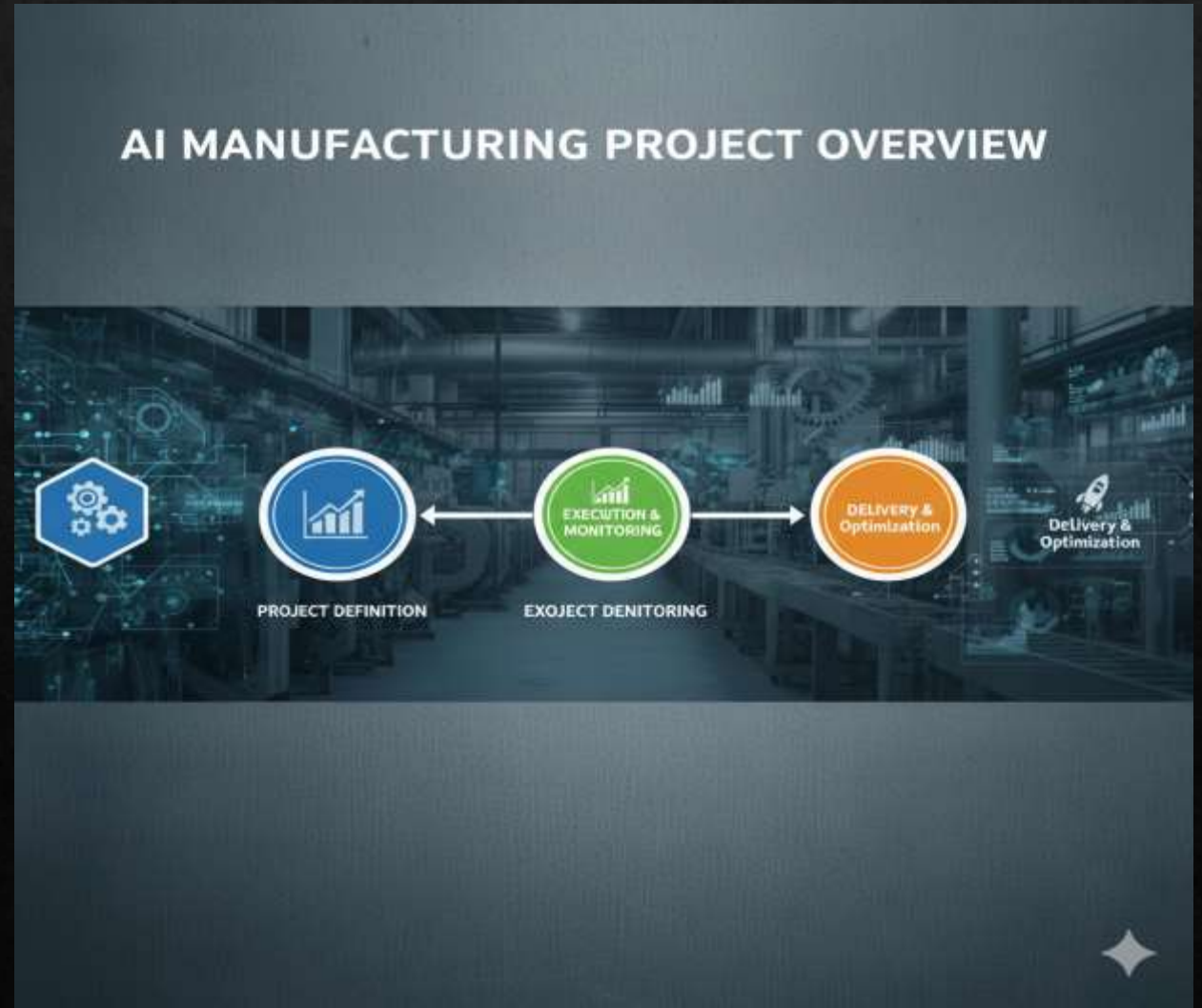
Using Computer Vision & Deep Learning



Presented by: SNEHA KUMARI

Project Overview

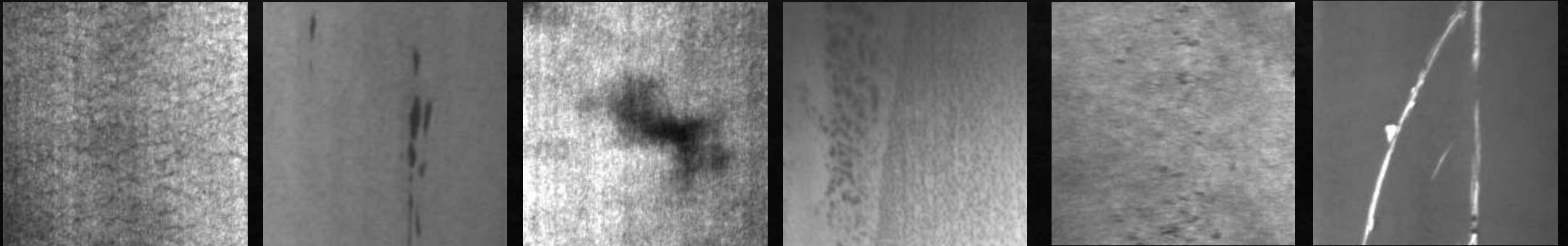
- ❑ Automates surface defect detection using CNN
- ❑ Improves quality control in industries like automotive, electronics, and steel
- ❑ Reduces manual inspection effort



Dataset

NEU Surface Defect Database (NEU-DET)

- ◆ 6 defect types: Craze, Inclusion, Pitted Surface, Scratches, Rolled-in Scale, Patches
- ◆ High-resolution steel surface images



Model Architecture

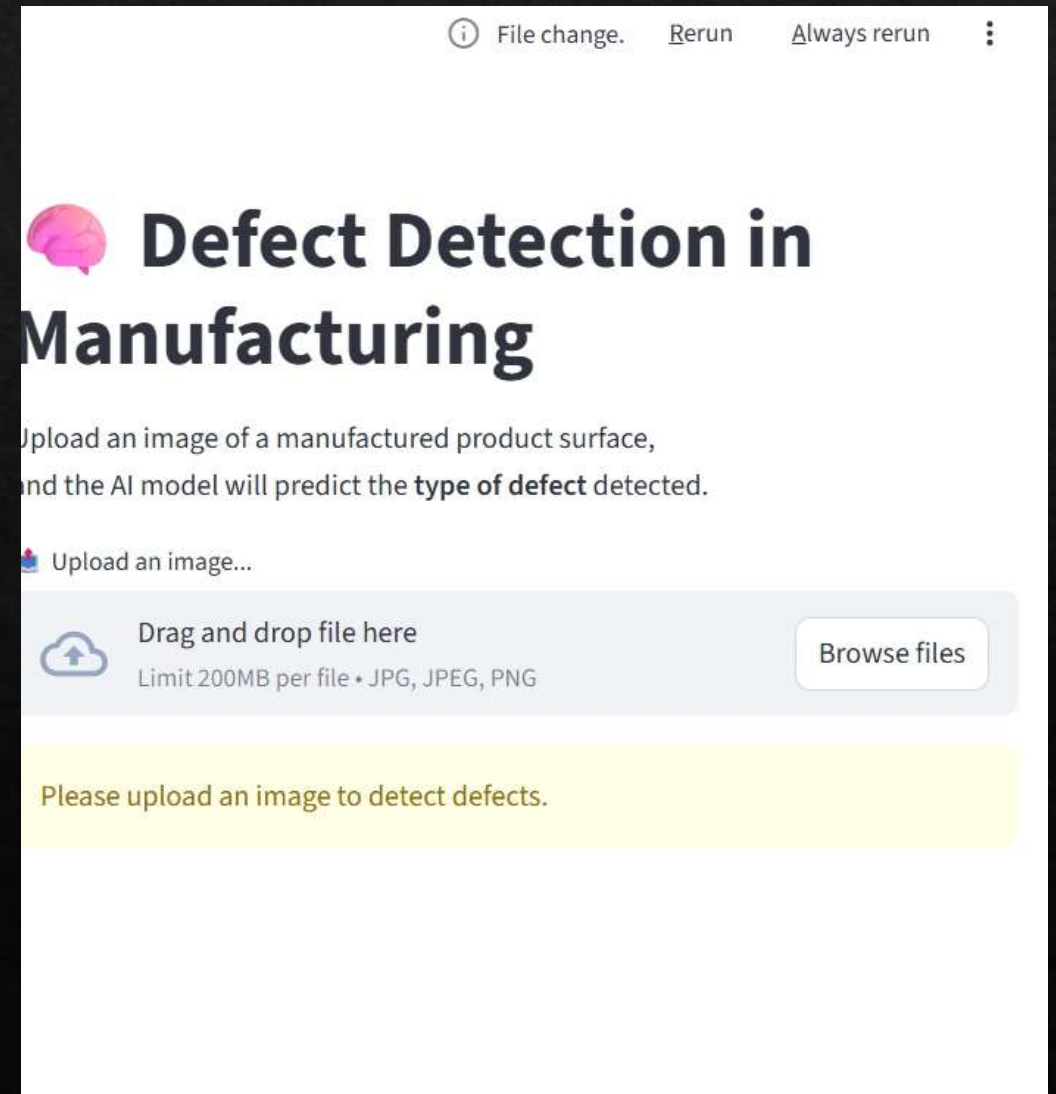
Convolutional Neural Network (CNN)

- ◇ Input: 128×128 images
- ◇ Output: 6 defect classes
- ◇ Optimizer: Adam
- ◇ Loss: Categorical Cross-Entropy
- ◇ Accuracy: $\sim 94\%$



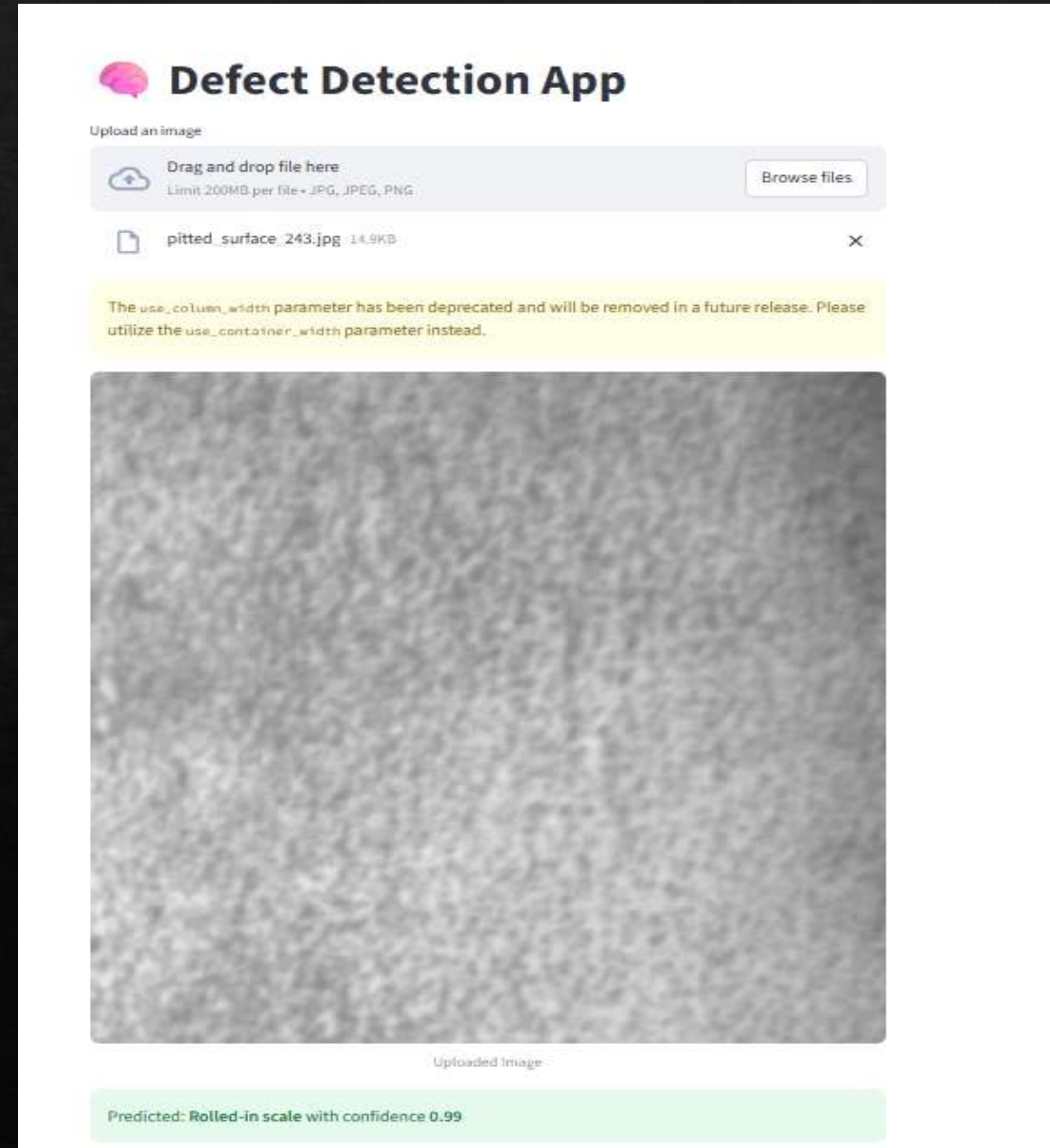
Streamlit App

- ◆ Upload interface for defect images
- ◆ Real-time prediction with confidence score
- ◆ Simple UI for non-technical users



Demo Output

- ◆ Example Prediction:
- ◆ Input: Steel surface image
- ◆ Output: Rolled-in-Scale(Confidence: 0.99)
- ◆ Fast and accurate results



Project Structure & Tech Stack

- ◆ Organized folders for data, models, notebooks, and app
- ◆ Key files: train_defect_detector.ipynb, app.py, defect_detection_model.h5

Languages & Libraries:

- ◆ Python, TensorFlow, Keras, NumPy, Pillow, Streamlit

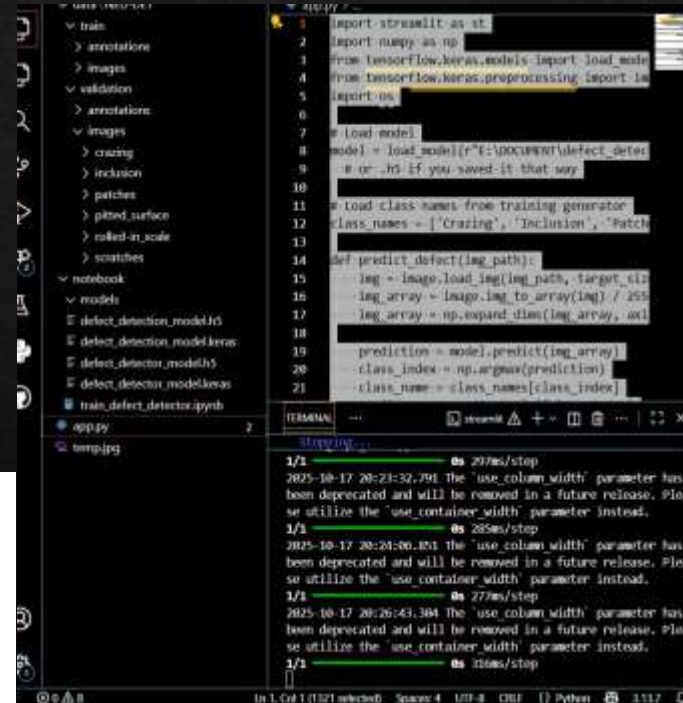
Tools:

- ◆ Jupyter Notebook, VS Code



Results & Use Cases

- ❖ High defect classification accuracy
- ❖ Useful for real-time industrial inspection
- ❖ Can be integrated with camera systems



```
1 import streamlit as st
2 import numpy as np
3 from tensorflow.keras.models import load_model
4 from tensorflow.keras.preprocessing import image
5 import os
6
7 # Load model
8 model = load_model('r:\document\defect_defect_model.h5')
9 # or .h5 if you saved it that way
10
11 # Load class names from training generator
12 class_names = ['Cracking', 'Inclusion', 'Patch']
13
14 def predict_defect(img_path):
15     img = image.load_img(img_path, target_size=(256, 256))
16     img_array = image.img_to_array(img) / 255
17     img_array = np.expand_dims(img_array, axis=0)
18
19     prediction = model.predict(img_array)
20     class_index = np.argmax(prediction)
21     class_name = class_names[class_index]
```



Future Improvements

- ◆ Live camera input
- ◆ Cross-material defect detection
- ◆ Cloud deployment
- ◆ Feedback-based model retraining



Thank you