Portfolio

Computer Vision & Anomaly Detection for Industrial Systems

Tianlang Liu

Vision Algorithm Engineer | CATL, Germany

📍 Germany | 💼 Open to relocation

🔗 linkedin.com/in/tianlang-liu

✉️ tlliu_liesmars@outlook.com

Surface Defect Detection with 2D + 3D Fusion

Tech Stack: Mask RCNN, PCL, Python, OpenCV

Problem:

Surface **holes** were often **missed by standard 2D Mask RCNN detection**, leading to false negatives and quality concerns.

Solution:

Developed a detection algorithm based on integration of **point cloud** and 2D images using Mask RCNN, checking suspected areas with **2D features, 3D features and registered areas depth**.

Results:

Reduced manual inspectors from 12 to 0; achieved **0% missed defects and <0.4% false alarms** across 200k+ products.

My Contribution:

Independently handled **data preprocessing**, model design and training, **PCL-based postprocessing**, and deployment scripting.

Die Surface Damage unsupervised Detection

Tech Stack: PatchCore, OpenCV, PyTorch

Problem:

Unseen defects on the die surface caused **YOLO to miss detections**. Retraining was not feasible due to the limited defect samples.

Solution:

Applied **unsupervised anomaly detection** using PatchCore and **post-classification** with ResNet, combined with classical image preprocessing in OpenCV to enhance contrast and localization of anomalies.

Results:

Achieved **0% missed defects** and **<0.14% false alarms** on over **200,000 units**, significantly improving early-stage defect detection.

My Contribution:

Designed the **full inspection pipeline**, tuned feature extraction and thresholding, and validated the system in large-scale production environments.

YOLOv8 Small Object Detection with Self-Attention

Tech Stack: YOLOv8, PyTorch, OpenCV

Problem:

Standard YOLOv8 models **struggled with detecting small wrinkles in X-ray images**, leading to missed defects and unstable recall.

Solution:

Integrated a self-attention mechanism into the YOLOv8 architecture to improve focus on small targets. Enhanced the feature representation for subtle defects while maintaining real-time inference speed in offline batch mode.

Results:

Achieved **0% missed defects and <0.3% false alarms** on offline inspection batches, significantly improving small-defect recall compared to baseline YOLO.

My Contribution:

Modified the **YOLOv8 backbone and attention structure**, trained and validated the model on defect datasets, and integrated it into the existing pipeline for batch analysis.

Electrode Orientation Filtering using YOLO + DBSCAN

Tech Stack: YOLO, OpenCV, Scikit-learn (DBSCAN), Python

Problem:

Electrodes **misaligned** in orientation were difficult to detect **using angle thresholds alone**, causing **defects to slip through** automated QA filters.

Solution:

Used **YOLO** to detect electrode **masks** and evaluated their alignment via overlap with its ROI. Applied DBSCAN clustering on overlap scores to **dynamically determine threshold** boundaries and filter out outliers.

Results:

Enabled reliable detection of orientation anomalies under varying production configurations. **Reduced human checking workload** and stabilized process quality.

My Contribution:

Modified the orientation filtering algorithm, tuned **DBSCAN parameters** for dynamic thresholding, and implemented the full postprocessing logic in Python.