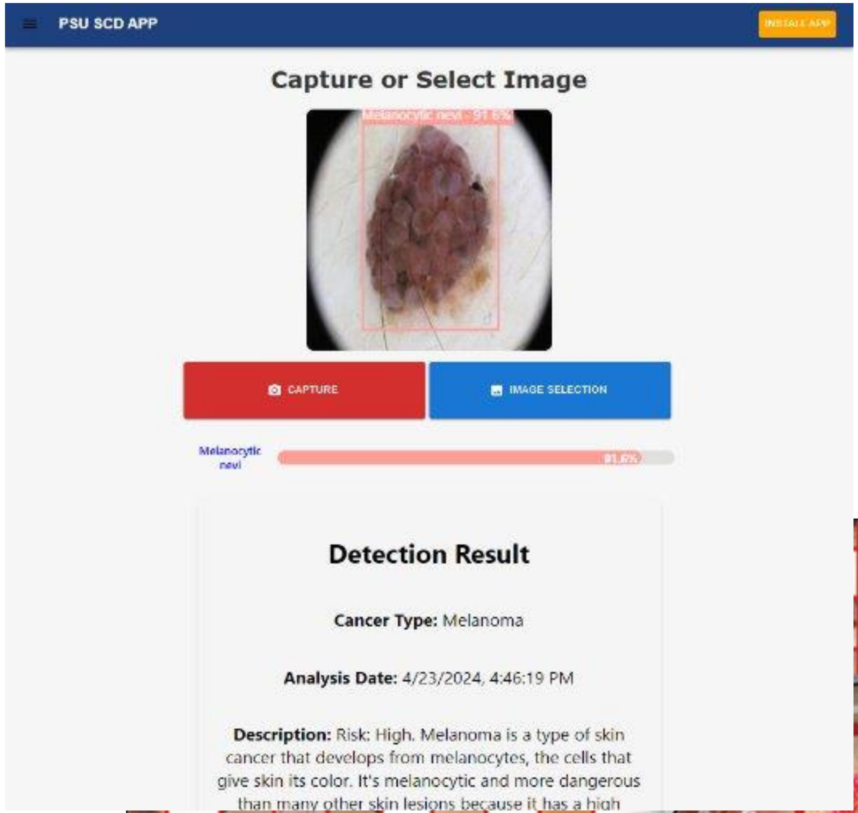
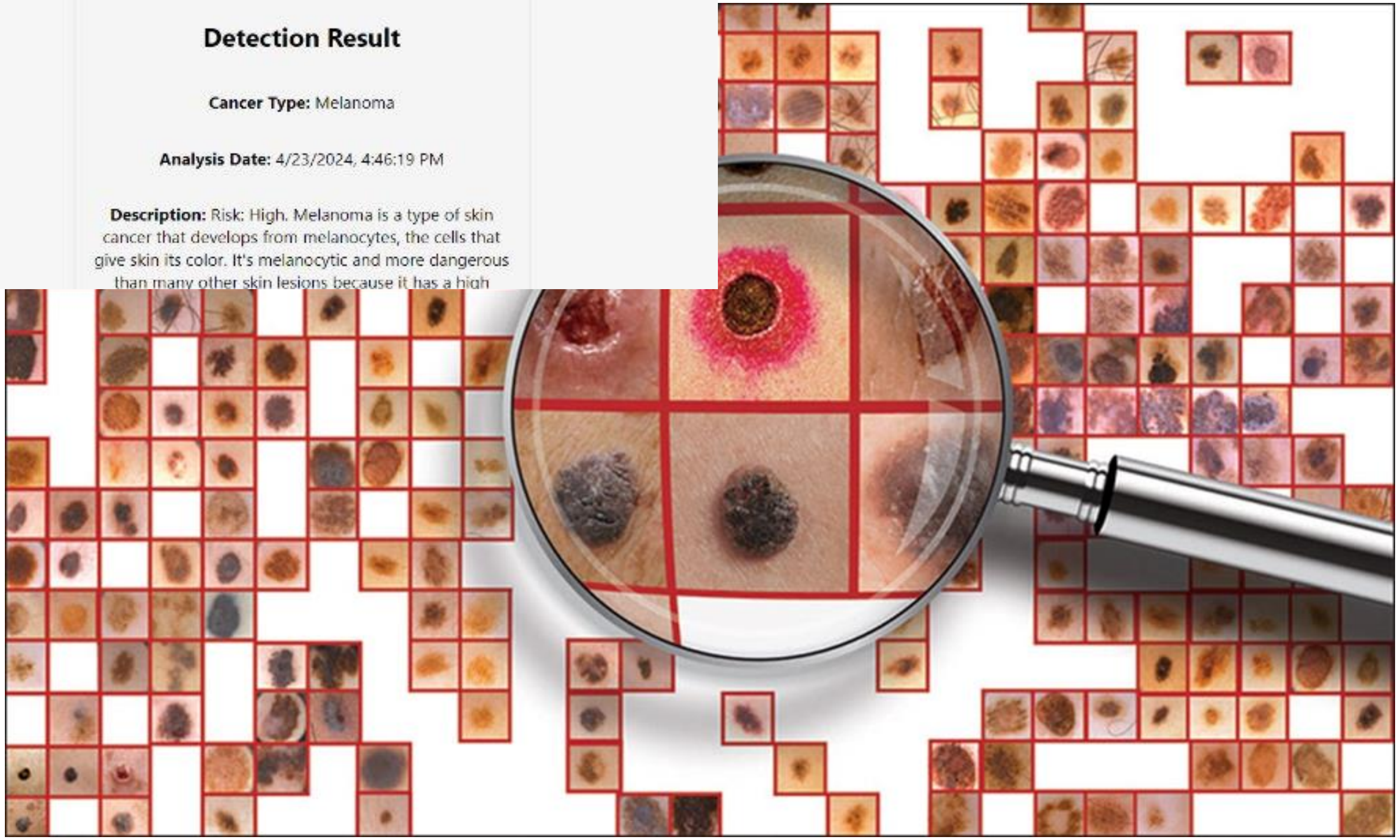
SKIN LESION DETECTION WEB APP PSU-SCD FINAL PRESENTATION





CMPSC 488

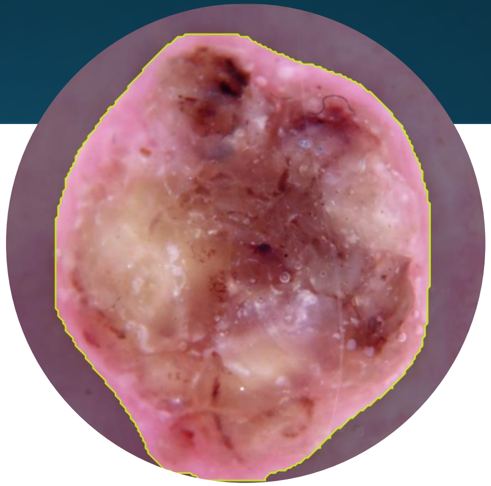
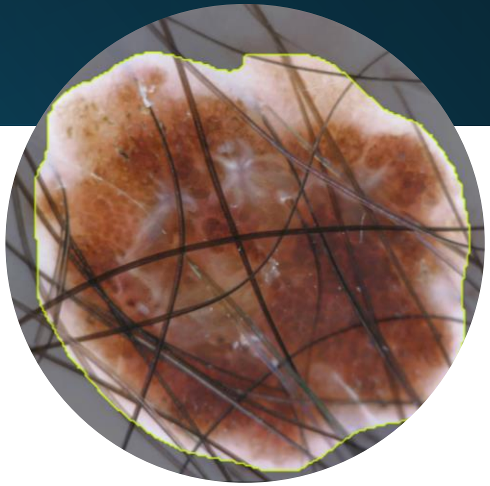
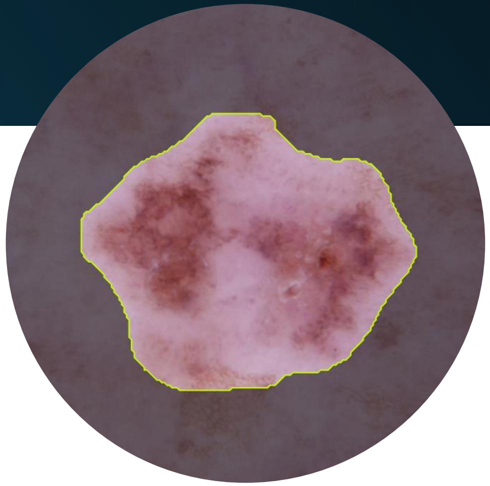
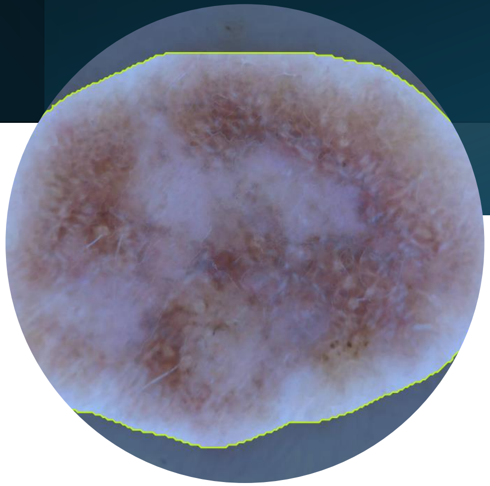
2024 SPRING

Jinyoon Kim, Tianjie Chen

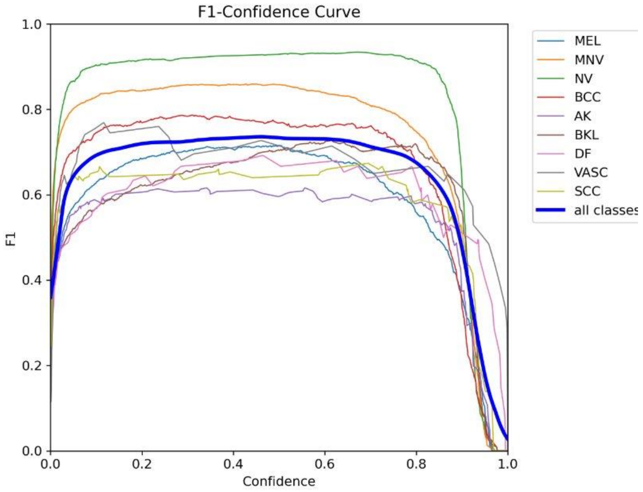
Motivation

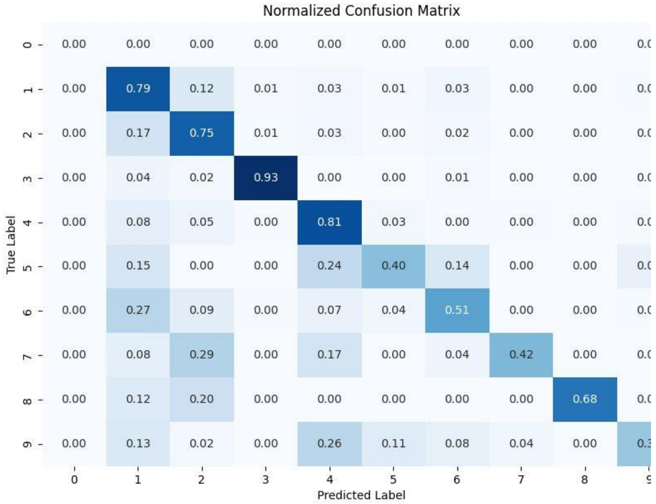
- Create a web application that can detect skin lesion and classify
- Combine available datasets (ISIC 2019 + ISIC 2020)
- Adopt the state-of-the-art machine learning approach
- Application is stable in various environments even without the internet connection.
- Develop the project into further research project that can improve the accurate detection of the architecture.
- Utilize the interpretability framework to analyze whether the model accurately comprehend the features of the data for its decision.

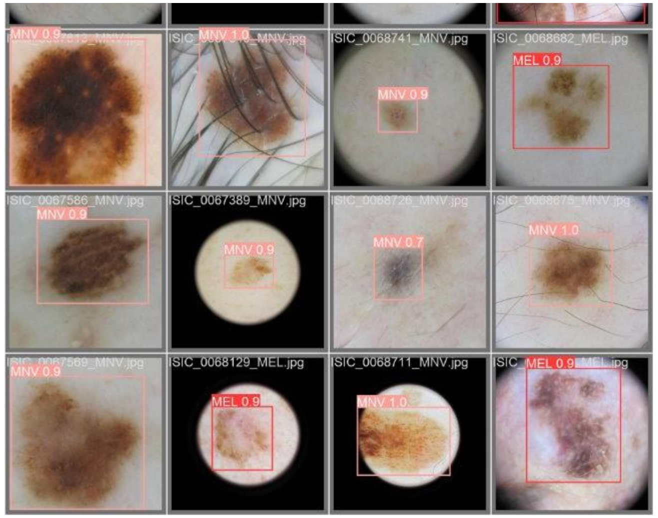
Dataset Preprocessing

- **Combine dataset.** of **ISIC 2019** and **ISIC 2020** datasets
- **Instance segmentation.** is applied to the data annotation
- This is to identify each distinct object of interest in an image for the model training
- **HAM10000.** dataset is used to train a model that helps the segmentation annotation of the combined dataset.
- **Data augmentation.** Data augmentation is used with built-in systems, such as the mosaic augmentation in the Ultralytics system.







Detection Model

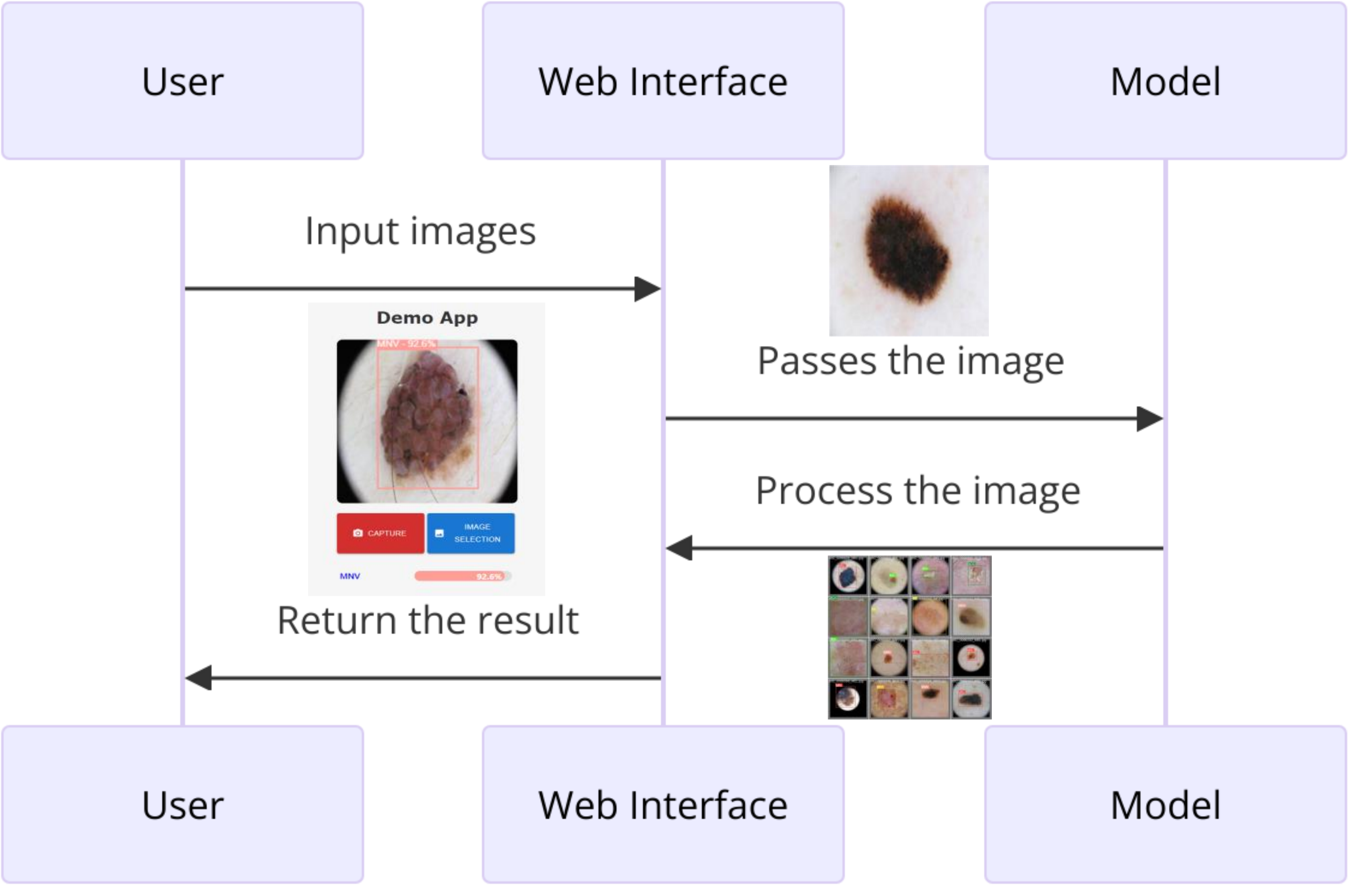
Model Type: YOLOv8

- **Tested Models:** Mask2Former, OneFormer, SwinV2, YOLOv8
- **Decision:** Adopted YOLOv8 as the baseline
- **Reason:** Demonstrated the highest accuracy and speed

Evaluation Metrics

- **Metrics Used:** Confusion matrix, F1 scores
- **Performance:** YOLOv8 showed the best performance

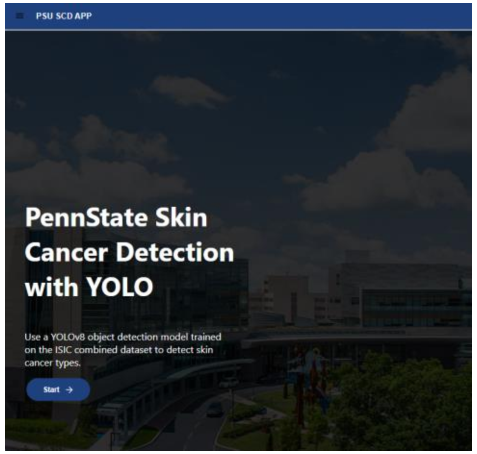
Web Interface (React)

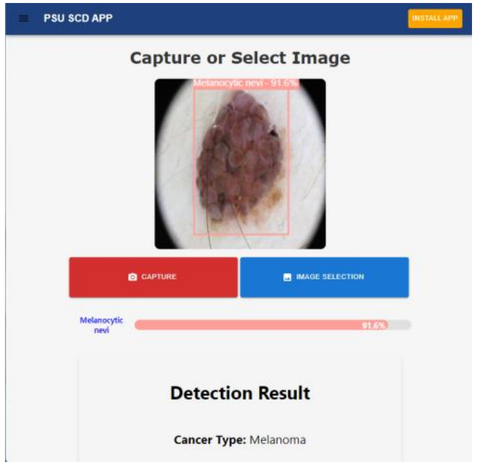


- User passes image to web interface
- Web interface passes image to the uploaded model
- Trained model outputs the result through the web interface
- Analysis can be saved to the device

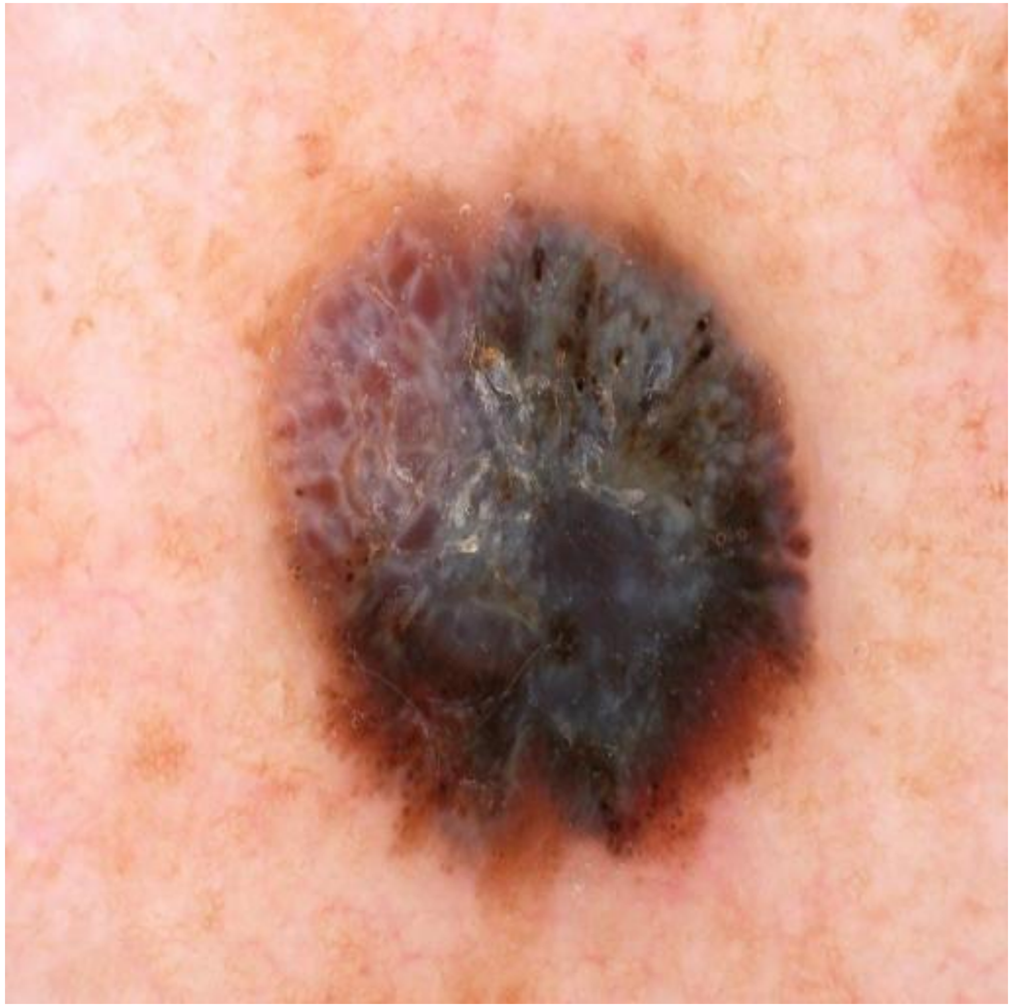
Web Application Online

- **Link: https://psu-scd.netlify.app or QR code scan**
- **Responsiveness:** Using a Progressive Web Application (PWA), the application is responsive across various OS and device environments.
- **Accessibility:** Since the model is downloaded within the application, it can operate even in offline environments.
- **Landing Page and Main Page:** The Landing Page introduces the application and provides a link to the repository. The Main Page processes and displays the results of the user's input image.





https://psu-scd.netlify.app



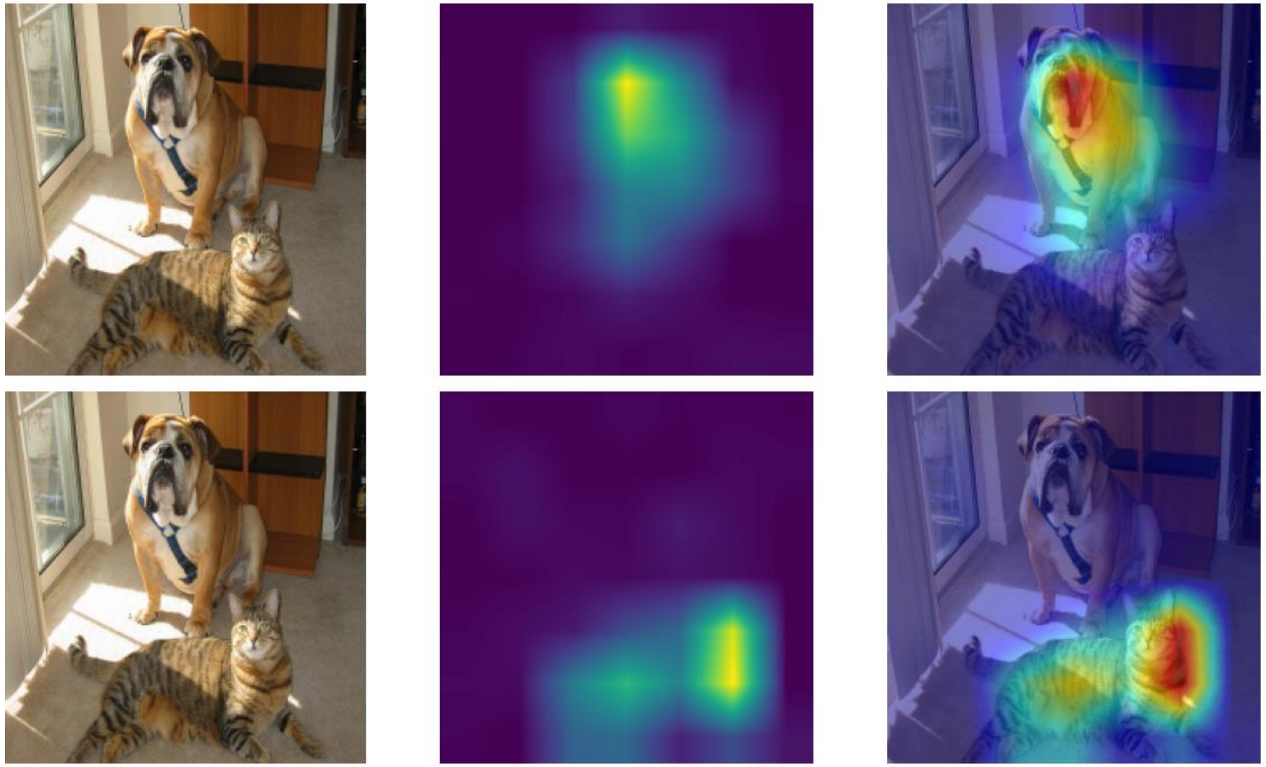
MEL

MNV

Further Development into Research

- **Limitations.** Medical decisions based on machine learning cannot solely rely on simple image detection due to legal reasons.
- **Changed Research Goal.** Enhance the accuracy of the detection process.
- **Phase 1:** Reinforced image preprocessing.
- **Phase 2:** Modify the model directly.

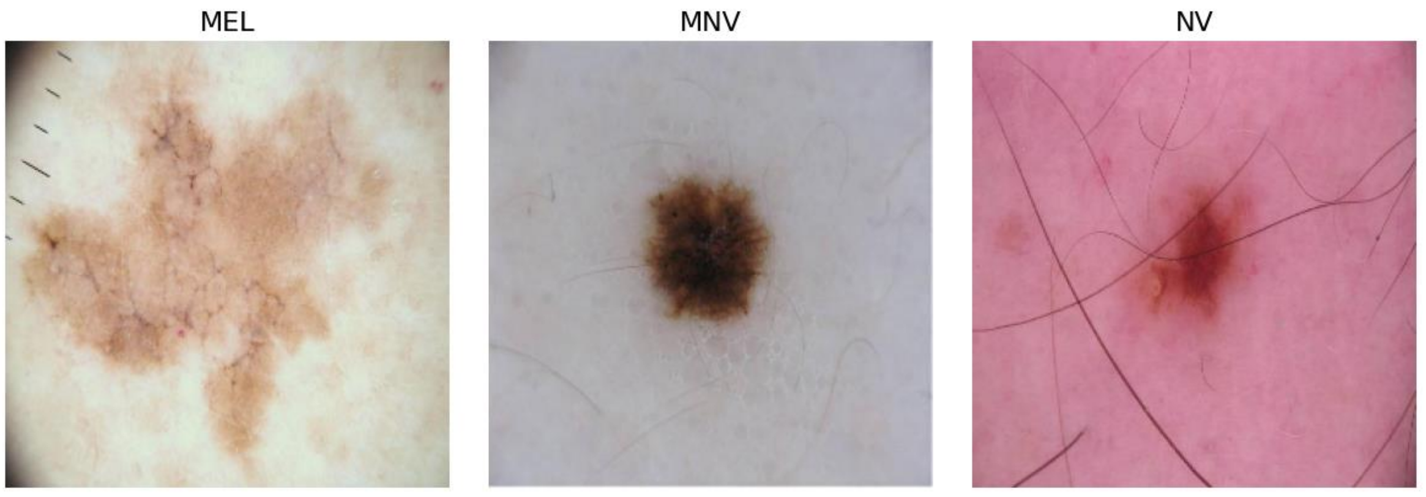
Functional Requirements - Explainability

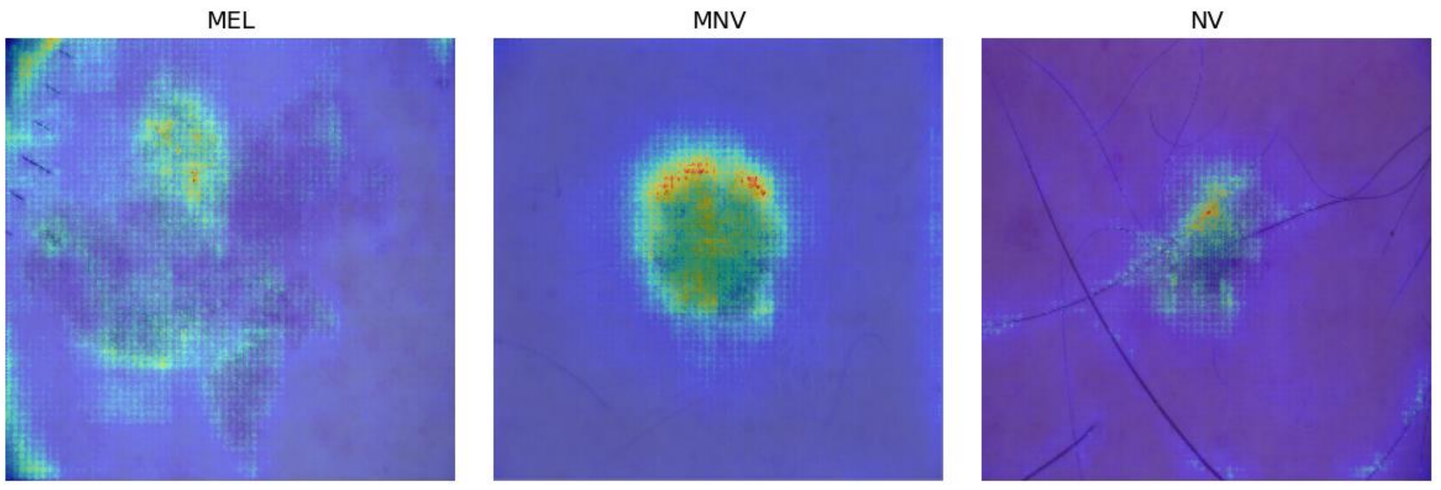


- **Regulatory requirement:** Compl y with emerging regulations on AI/ML
- **Model validation:** Identify and eliminating biases
- **User friendliness:** Explain results to people without domain knowledge
- **Mitigation of risks:** Cybersecurity, fraudulent detection, etc.

How was explainability used in this project?

- **Model validation**
  - Current model will treat artifacts such as hair and dark corners as features.
  - This motivated us to develop methods to remove such artifacts.
- **Feature Identification**
  - People without domain knowledge will find explanations generated by pixel attribution-based methods difficult to understand.
  - Addressing this issue will be one of our future research directions.



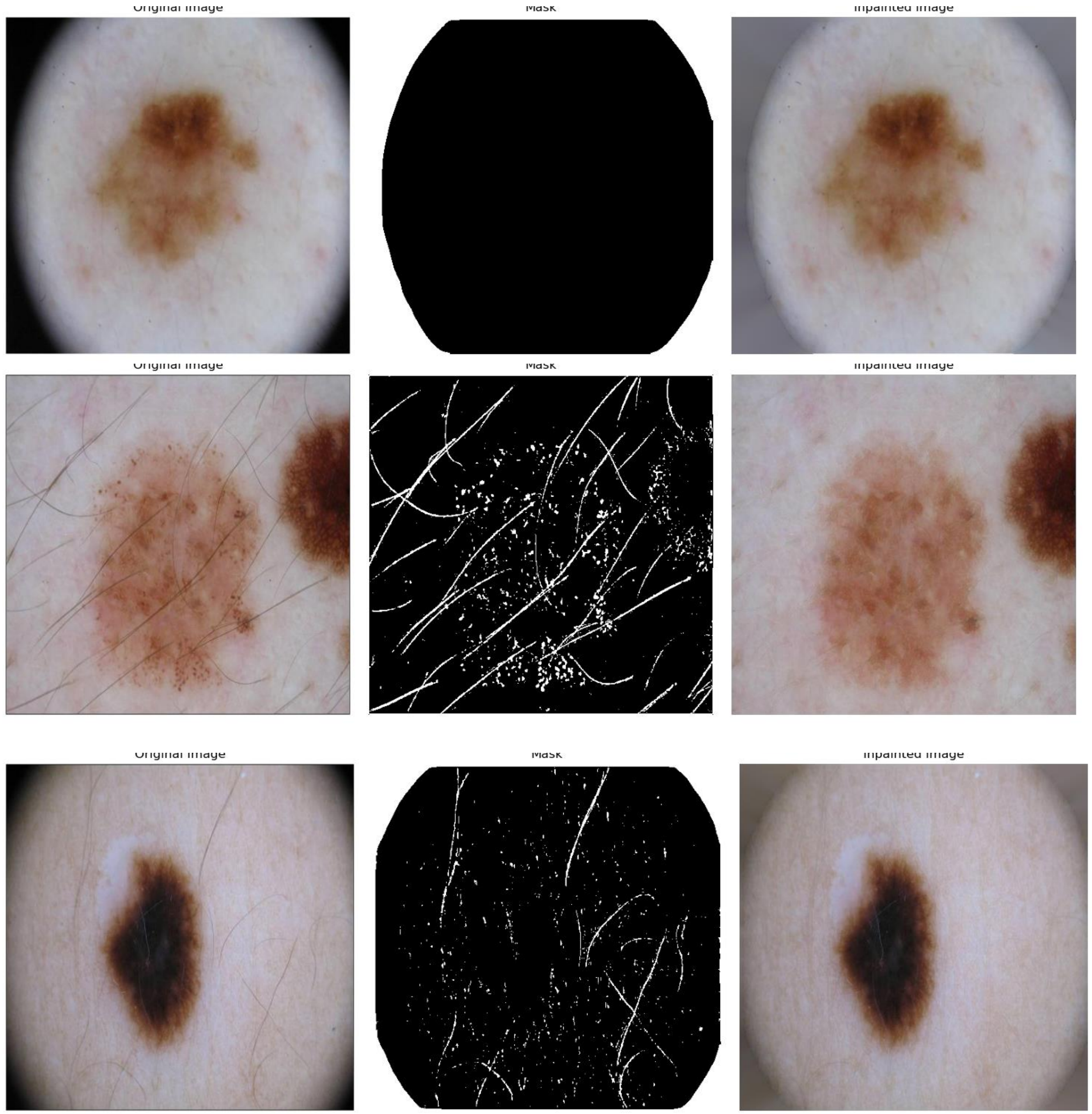


Phase 1
Image Preprocessing for Confounding Factors

- **DCA Removal (1):** Dark Corner Artifacts (DCA) have been identified as confounding factors in prior research on machine learning detection.

- **2 Hair Removal (2):** Hair can also be a confounding factor.

- **3 DCA + Hair Removal (3):** Consider scenarios where both factors exist in t he same image.



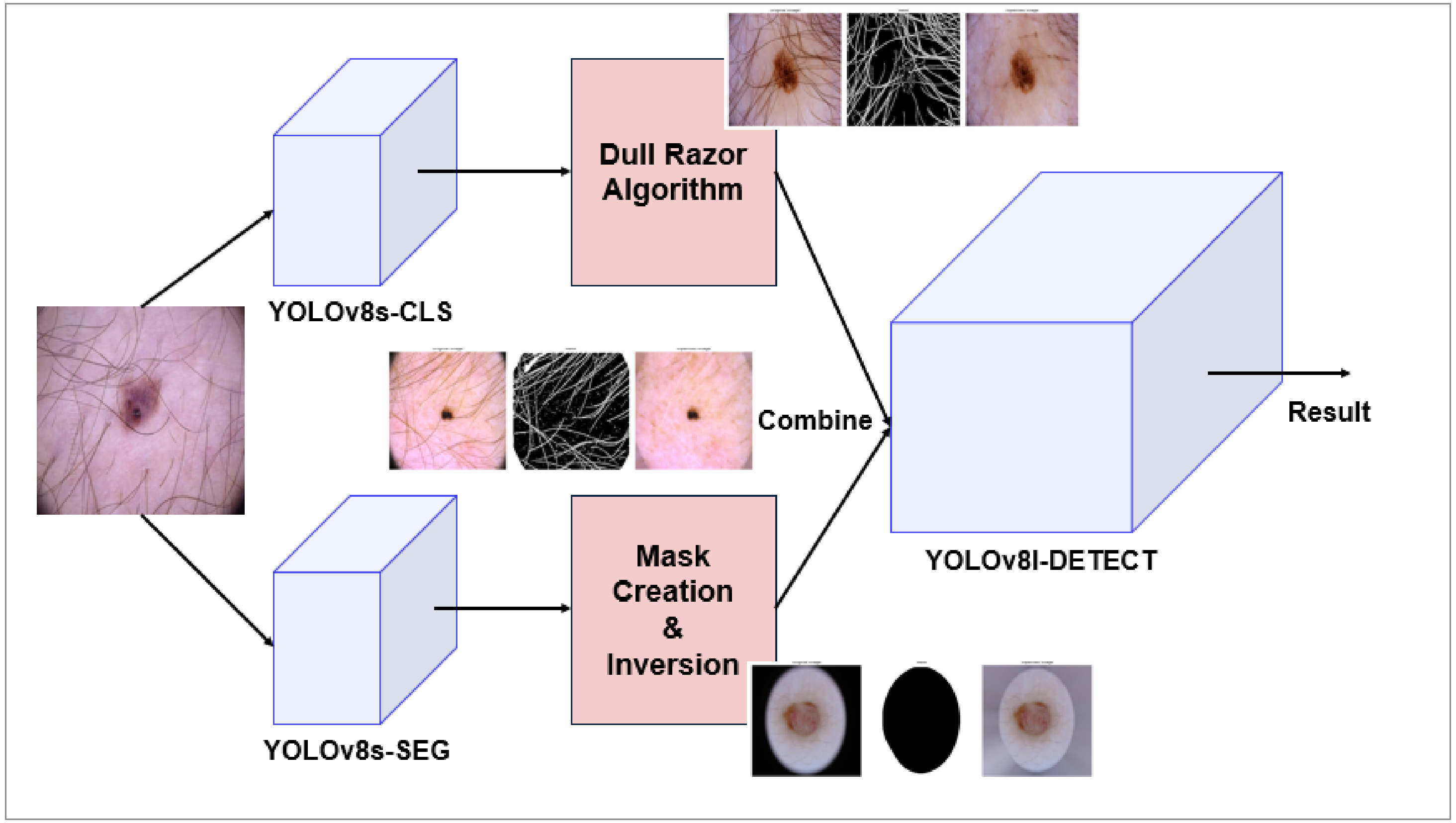
1

2

3

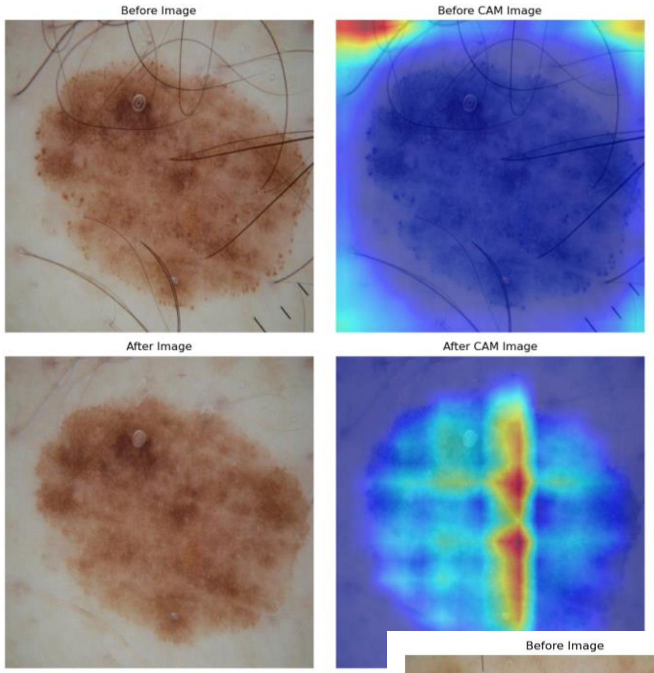
Phase 1 – Image Processing Structure

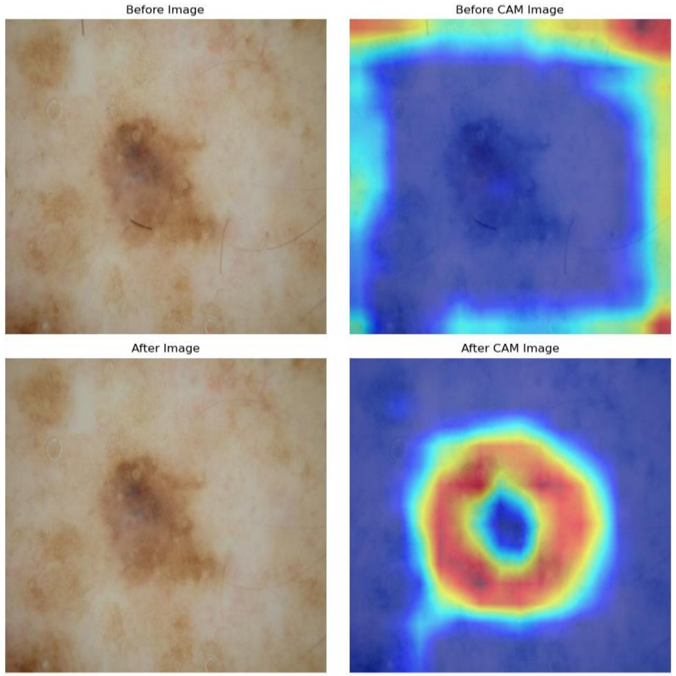
- **Pipeline.** Developed pipelines to process images and remove confounding factors.
- **Detection/Classification.** Use variations of the YOLOv8 model to identify images with confounding factors.
- **Post-Detection Processing.** Apply algorithms such as Dullrazor to create masks, then use inpainting algorithms to refine the masked sections.



Results

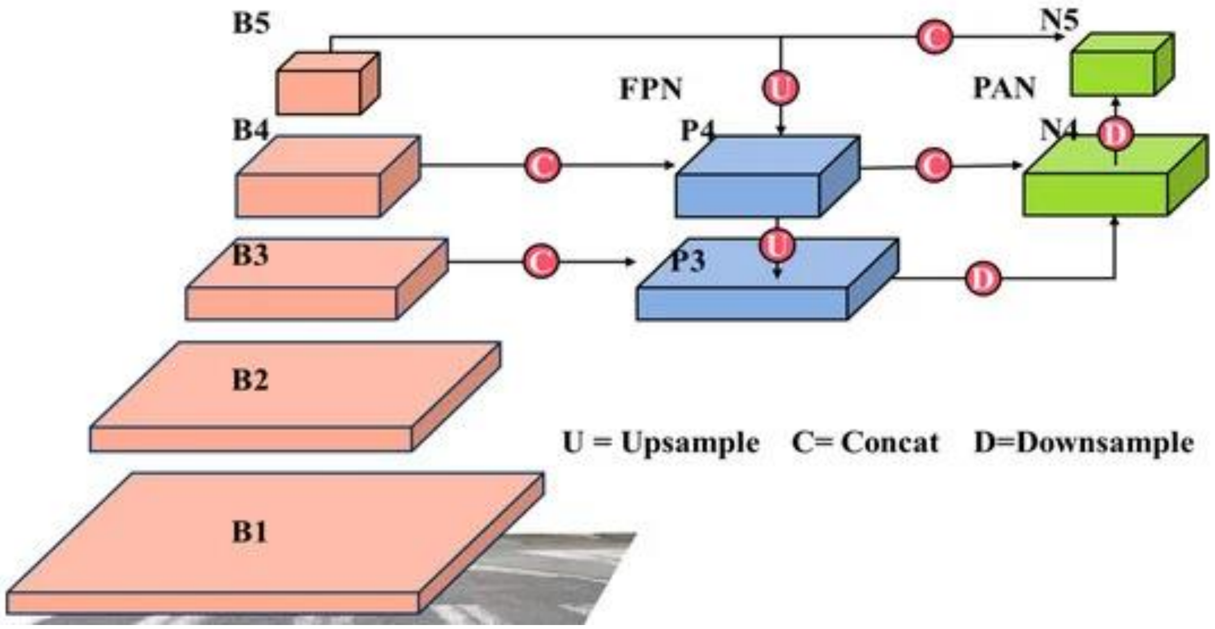
- **Interpretability Analysis.** Temporary EigenCAM to assess changes in the model's understanding of processed images.
- Some samples showed improvement, with focus shifting from unrelated parts to the center, while few deteriorated.
- **Accuracy Results.** Improvements in accuracy were modest, with only a 0.05 increase in the mAP@50 score. This indicates that **further improvements or replacements with better algorithms** are necessary for future enhancements.





Phase 2 – Model Modification

- On progress for the final publication
- **Integration of modules** that improve generalization and accuracy is expected in the second stage of the research
- Such as the **Bidirectional Feature Pyramid Network (BiFPN)** module. This module enhances feature integration across different scales.



https://www.mdpi.com/1424-8220/23/20/8361

Technical Platforms

- Framework: **Pytorch**
- Model: **Ultralytics**, **HuggingFace**
- Explanation: **EigenCAM**
- Human intervention data annotation tool: **Roboflow**

Future Plans

- **Complete the Second Stage of the Research.** Finish the ongoing second phase of the research project.
- **Reinforce the First Stage.** Enhance the effectiveness of the initial stage to improve the model's comprehension.
- **Expand into Related Research.** Develop additional research focused on enhancing model interpretability.

Thank you!