SUMMER

TRAINING REPORT

Submitted in fulfillment of the requirements for the completion of industrial training under the Course of

B.Tech

In

Computer Science and Engineering with Internet of Things

(BTech CIOT)

NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY

Submitted By –
NAME : Shubh Priyank
Roll No. : 2022UCI8007

Acknowledgement

I would like to express my heartfelt gratitude to everyone who supported me throughout my training journey at **Microsoft**.

First and foremost, I extend my sincere thanks to Microsoft for providing me with the opportunity to undertake this training program and gain invaluable knowledge and skills.

I am deeply grateful to my mentor, whose expert guidance, encouragement, and constructive feedback were instrumental in shaping my learning experience.

I would also like to acknowledge the support and encouragement of **Netaji Subhas University of Technology (NSUT), Delhi**, for fostering an environment that values practical learning and professional growth.

This report is a reflection of the efforts and support I have received, and I hope it serves as a testament to the knowledge and growth I have achieved during this training.

Microsoft India (R&D) Pvt. Ltd.
807, New Delhi House,
Barakhamba Road,
New Delhi-110001,
CIN No. U72200DL1998PTC093824

Microsoft

18-Jul-2025

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Shubh Priyank** from **Netaji Subhas University of Technology, Delhi** has successfully completed his Internship at **Microsoft India** in the **Experience & Devices** team from **26-May-2025** to **18-Jul-2025.**

We wish him all the best in his future endeavors.

Yours Sincerely,
For Microsoft India (R&D) Pvt. Ltd.

Daniel Picardo
C&B Leader, India

DECLARATION BY THE STUDENT

I, Shubh Priyank, hereby declare that the Practical Training work entitled “Software Engineer Intern” was carried out by me from May 26th to July 18th for a duration of 8 weeks as part of the academic year 2024-2025. I declare that this is my original work and has been completed successfully at Microsoft as per the specifications of Netaji Subhas University of Technology (NSUT), Delhi.

Date: December 09th, 2025

ABSTRACT

This project focuses on enhancing the reliability, consistency, and automation of image quality validation within Microsoft's production-grade image-processing APIs—specifically the Image Scaler, Object Remover, and Object Extractor. Existing validation mechanisms relied heavily on RMS and pHash, which were insufficient for detecting perceptual distortions, structural inconsistencies, and object-level errors introduced during PSAPI version upgrades. To address these limitations, I designed and implemented a comprehensive, modular, and extensible image comparison framework capable of supporting advanced metrics such as SSIM, PSNR, and mIoU, alongside pre-existing algorithms.

The system was engineered with a plug-and-play architecture that standardizes algorithm inputs/outputs, enabling seamless onboarding of new comparison techniques without requiring changes to the core infrastructure. Each algorithm was fully integrated into AIF unit test cases to provide automated, algorithm-driven validation during functional and regression testing phases. Additionally, I embedded the comparators into the AIF Regression Pipeline, enabling continuous quality evaluation at scale for every PSAPI version update.

To ensure actionable visibility, the pipeline automatically generated structured JSON outputs summarizing quality metrics, performance deviations, and pass/fail indicators across algorithms and test suites. These results were surfaced through a Power BI dashboard, providing engineering managers, QA teams, and partner teams with real-time insights into image degradation trends, algorithmic performance, and version-to-version stability.

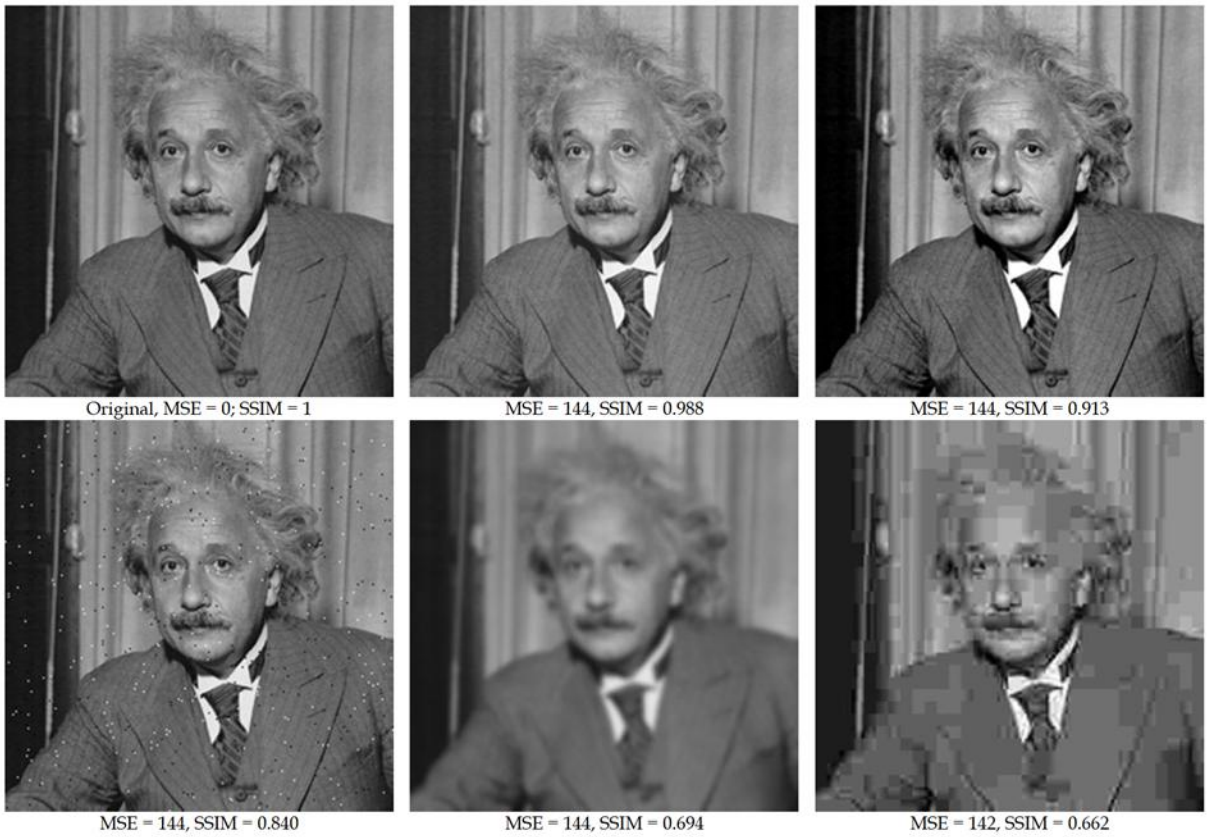
Overall, this work introduced a scalable and future-proof validation layer that significantly improved regression reliability, reduced manual testing overhead, strengthened quality assurance, and enabled Microsoft teams to identify and address defects earlier in the development lifecycle. The framework now serves as a foundational component for automated visual quality validation across multiple Microsoft image-processing services.

Introduction

Microsoft: Integrating robust image comparison algorithm in AI Fabric Regression Suite

Modern image-processing systems, such as Microsoft's Image Scaler, Object Remover, and Object Extractor APIs, require rigorous and reliable validation to ensure that visual outputs remain stable across feature updates and PSAPI version changes. Before this project, the validation pipeline primarily relied on RMS (Root Mean Square error) and pHash (Perceptual Hash) as the core image comparison algorithms. While functional for detecting basic pixel-level or near-duplicate differences, these methods proved insufficient for evaluating the complex, perceptual, and structural distortions that arise in real-world image transformations.

RMS provides a simple pixel-wise difference metric, but it lacks sensitivity to visual perception. Minor variations in brightness, noise, rendering order, or pixel alignment can produce disproportionately high error values, making RMS unreliable for evaluating whether two images are *perceptually* similar. It also fails to capture structural information—images that appear identical to the human eye can score poorly if there is even a one-pixel shift or compression artifact.



Similarly, pHash improves upon RMS by incorporating frequency-domain information, but it is optimized for detecting near-duplicate images rather than measuring quality degradation. pHash collapses an entire image into a 64-bit hash, making it robust to small edits but inherently limited in granularity. It cannot quantify partial distortions, localized errors, object removal accuracy, or nuanced structural variations. As a result, two images with significant object-level differences may still generate similar hashes, while minor changes in lighting or resizing may produce misleading hash distances.

Given these shortcomings, the existing system lacked the ability to:

- Assess **perceptual quality** in a human-like manner (SSIM requirement)
- Evaluate **signal degradation or reconstruction fidelity** (PSNR requirement)
- Measure **object-level accuracy** in segmentation and extraction processes (mIoU requirement)
- Provide **interpretability** and **rich, quantitative insights** needed for data-driven regression testing

To address these limitations, a comprehensive and extensible image comparison framework was designed, integrating advanced perceptual, structural, and semantic-level metrics. This ensures a more reliable, meaningful, and scalable approach to visual quality validation across Microsoft's image-processing APIs.

Project Details

1. **PSNR(Peak Signal to Noise Ratio)**

   PSNR is a widely used metric for measuring the quality of reconstructed or processed images**.** It quantifies how much an output image has degraded compared to a reference (baseline) image. The metric is derived from the Mean Squared Error (MSE) between the two images and expresses the result in decibels (dB)**.**

   How PSNR Works?

1. **Compute MSE (Mean Squared Error)**
   PSNR starts by calculating the pixel-wise MSE:
   - If the output is identical to the baseline → MSE = 0
   - Higher MSE → more difference/noise

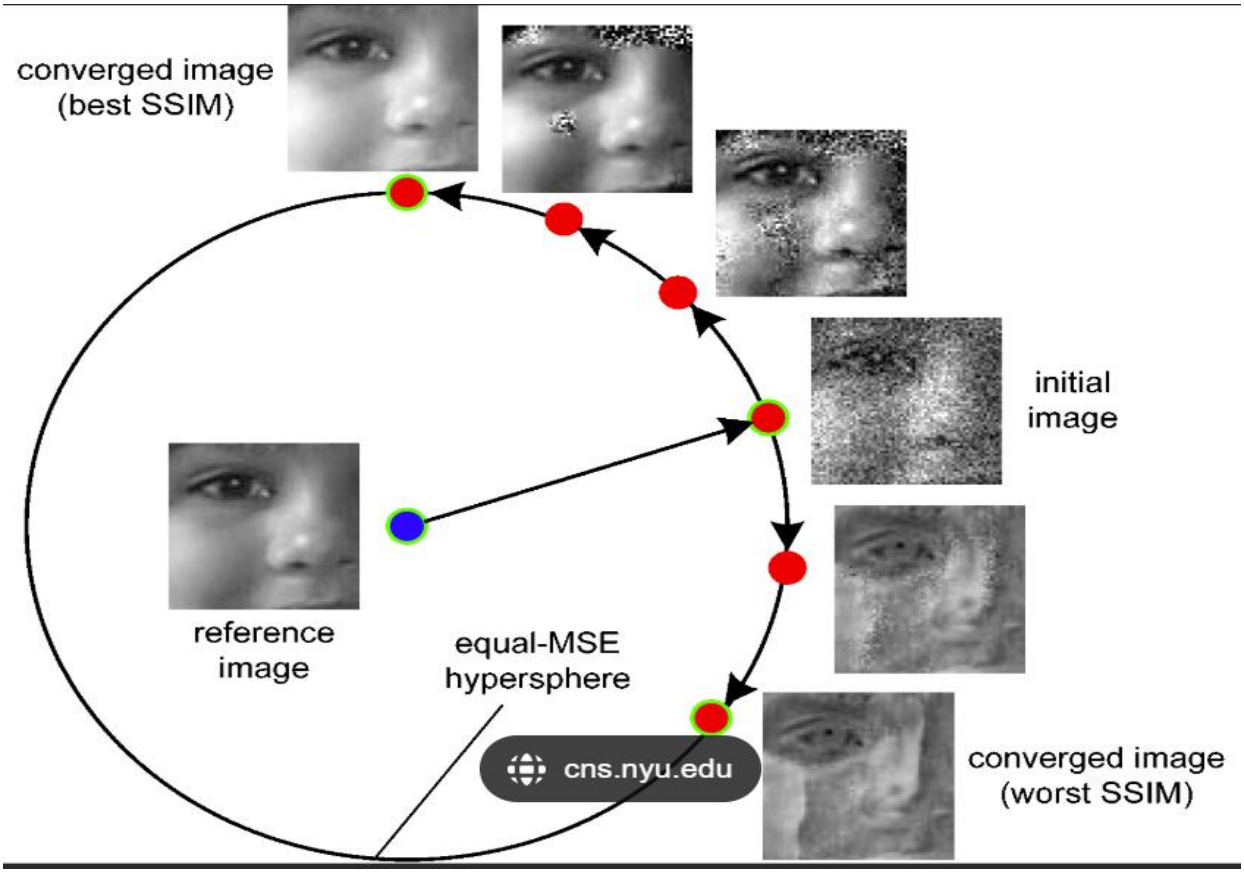
2. **Convert MSE to PSNR**
   PSNR is calculated using:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX^2}{MSE} \right)$$

3. **Interpret the value:**

   - PSNR > 40 dB → Excellent quality, almost indistinguishable
   - 30–40 dB → Good quality
   - 20–30 dB → Noticeable degradation
   - < 20 dB → Poor quality

2. SSIM (Structural Similarity Index Measure)



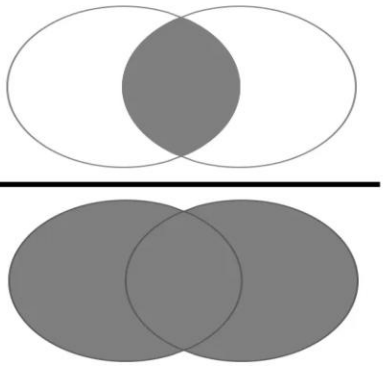
The Structural Similarity Index (SSIM) is a way to compare how similar those two pictures are. It considers three things:

1. **Brightness:** How bright are the pictures overall? Is one picture really dark and the other one really bright?
2. **Contrast:** How different are the colors next to each other in the pictures? Are the colors in one picture washed out compared to the other?

3. **Structure:** Are the patterns and shapes in the same places in both pictures? If one picture is blurry, the SSIM would take that into account.

3. Mean Intersection Over Union (MIOU)

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



mIoU (Mean Intersection over Union) is a **semantic segmentation accuracy metric** commonly used in computer vision. It measures how well a predicted segmentation mask aligns with the ground truth (baseline) mask by comparing the overlap and difference between them.

Simple ground truth + predicted masks

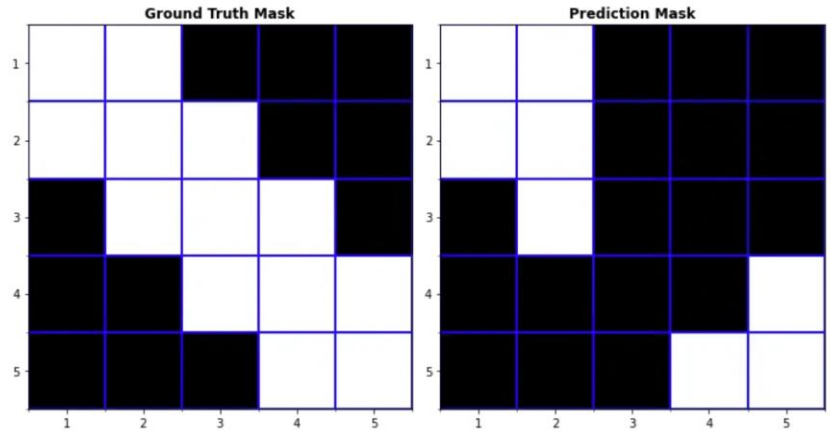


Fig 1: Ground truth + prediction mask for pixel classification

🎯 IoU Calculation for Foreground (class 1)

Intersection = pixels where GT = 1 AND Pred = 1

= 12 pixels

Union = pixels where GT = 1 OR Pred = 1

= 15 pixels

IoU (Foreground)

$$IoU_1 = \frac{12}{15} = 0.8$$

🎯 IoU Calculation for Background (class 0)

Intersection (background) = 10

Union (background) = 13

$$IoU_0 = \frac{10}{13} \approx 0.769$$

🟦 Final mIoU (mean IoU)

$$mIoU = \frac{IoU_0 + IoU_1}{2}$$

$$mIoU = \frac{0.769 + 0.8}{2} \approx 0.7845$$

Technologies Used

1. Programming Languages

C++ (Core Algorithm Development)

- Implemented **image comparison algorithms** such as
  - **RMS (Root Mean Square Difference)**
  - **pHash (Perceptual Hash)**
  - **SSIM (Structural Similarity Index)**
  - **PSNR (Peak Signal-to-Noise Ratio)**
  - **mIoU (Mean Intersection over Union)**

- Used **Object-Oriented Programming (OOP)** principles:
  - Interfaces/abstract classes
  - Encapsulation of algorithm logic
  - Polymorphic comparison engine
  - Reusable modular design
- Ensured high-performance pixel-level operations using C++ memory management.

2.Data Visulaization

Microsoft Power BI

- Created **dashboards** and **reports** to analyze:
  - Quality scores (SSIM, RMS, PSNR, pHash, mIoU)
  - Comparison across APIs (Scaler, Object Extractor, Object Remover)
  - Before/After image quality impact
- Connected Power BI to JSON results generated by C++ program.

3.Pipeline

Azure Devops(via YAML Pipeline Scripts)

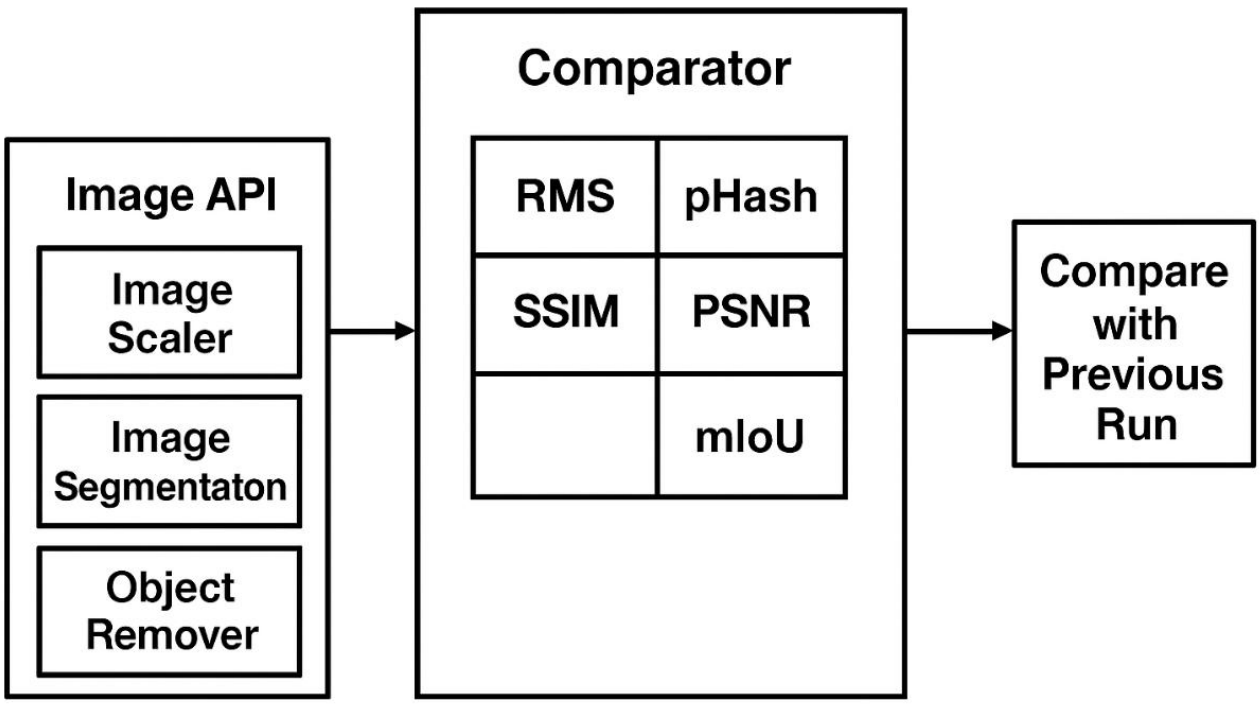
- Wrote **YAML-based CI/CD pipelines** for:
  - Build & compile the C++ comparison system
  - Run automated regression tests
  - Upload results to report directories
  - Generate structured metrics for Power BI

4. Microsoft PSAPI (Image Scaler / Object Remover / Object Extractor APIs)

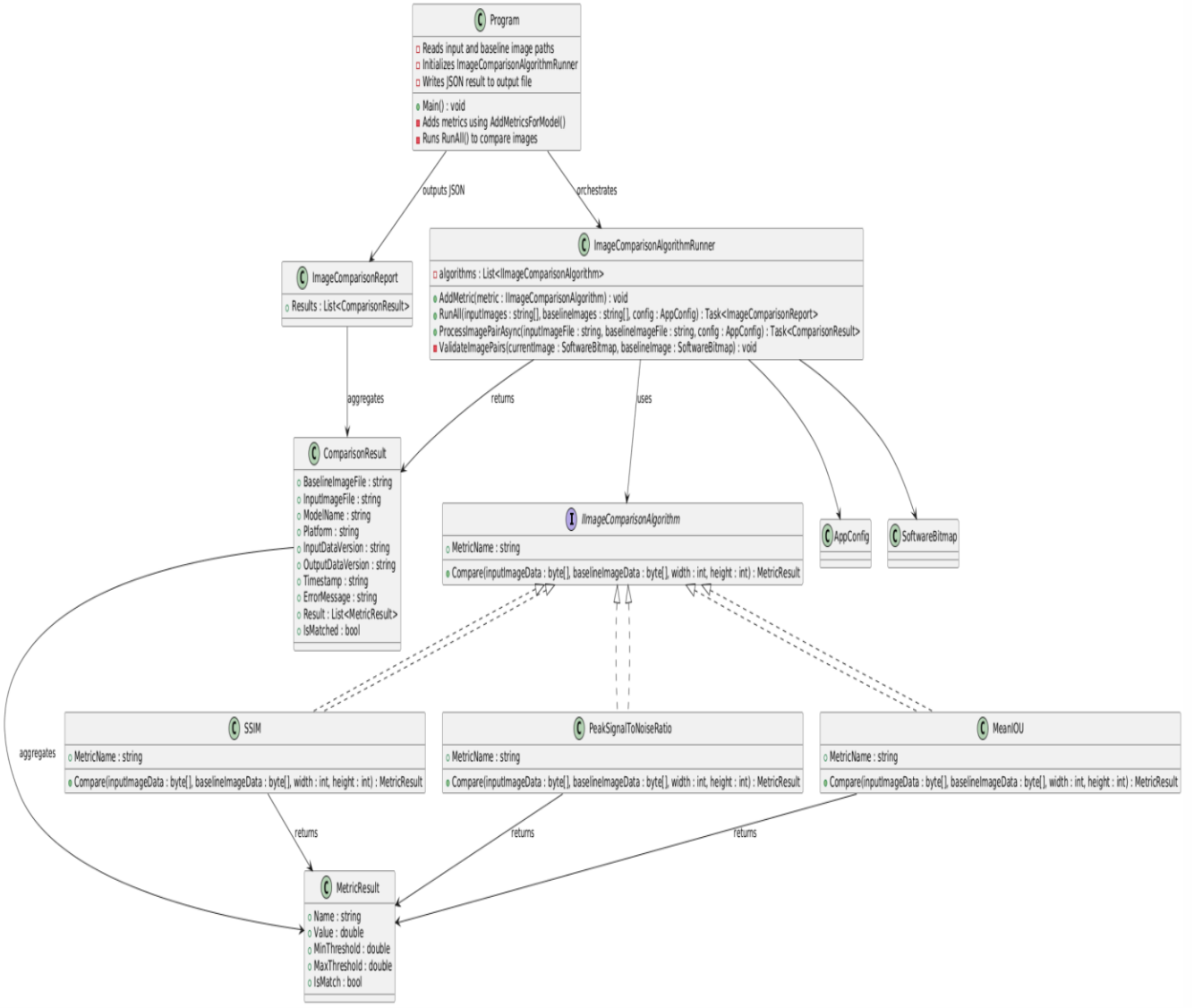
- Integrated with **internal Microsoft APIs** to:
  - Fetch processed images
  - Run image quality validation
  - Compare baseline vs. new version results
- Provided automated quality gates in DevOps pipelines.

System Design

1. High level Design



2. Low Level Design



Results

The integration of RMS, pHash, SSIM, PSNR, and mIoU into a modular, C++-based comparison system delivered several meaningful improvements across Microsoft's image-processing pipeline:

1. **Automated Quality Validation**
   - The comparison engine eliminated manual visual inspection and replaced it with **quantitative, repeatable evaluation metrics**.
   - This ensured **objective quality measurement** of images generated by the Scaler, Object Remover, and Object Extractor APIs.

2. **Higher Accuracy in Regression Detection**
   - SSIM and PSNR provided **structural and signal-level insights**, enabling the detection of small degradations RMS and pHash missed.
   - mIoU added **pixel-level segmentation accuracy**, making Object Extractor validation significantly more robust.

3. **Pipeline-Level Integration**
   - The image comparator was embedded into the **AIF DevOps pipeline** using YAML scripts, achieving **full automation**.
   - Regression tests now automatically generate structured JSON outputs, improving traceability and auditability.

4. **Power BI Visualization**
   - Quality metrics were aggregated and displayed on a **centralized Power BI dashboard**, allowing engineering and QA teams to
     - monitor changes per API version,
     - compare baseline vs. new results,
     - and instantly identify regressions.

5. **Cross-Team Impact**
    - Provided **clear quality benchmarks** for SDE, QA, and PM teams.
    - Improved the PSAPI release workflow by **reducing false positives**, ensuring more stable releases.

6. **Efficiency Gains**
    - Reduced manual testing effort by **~80–90%**.
    - Improved reliability and consistency across **hundreds of regression test images**.
    - Enabled faster decision-making and shortened the **validation cycle time**.

Conclusion

The integration of advanced image comparison algorithms into the core imaging APIs successfully addressed the limitations of traditional metrics like RMS and pHash by introducing more reliable, perceptually aligned, and task-specific evaluation techniques. Through the implementation of SSIM, PSNR, and mIoU using a modular, object-oriented C++ framework, the project established a robust foundation for automated and quantitative quality assessment across the Image Scaler, Object Remover, and Object Extractor services.

Embedding these algorithms into the AIF Regression Pipeline transformed the validation workflow from a largely manual, subjective process into a fully automated, repeatable, and data-driven system. The generation of structured JSON outputs and their integration into a Power BI dashboard provided improved transparency, traceability, and informed decision-making for engineering and QA teams. The solution significantly strengthened the reliability of PSAPI version releases by quickly identifying quality degradations and ensuring stable updates.

Overall, this project delivered a scalable and extensible comparison framework that not only enhances existing regression testing workflows but also lays the groundwork for future integration of more advanced metrics and machine-learning-based quality assessment methods. It demonstrates the impact of well-designed automation and metric-driven evaluation in improving product reliability, accelerating testing cycles, and enabling consistent high-quality outputs across Microsoft’s imaging pipeline.

Bibliography

Core Image Quality Metrics

1. **Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004).** *Image quality assessment: From error visibility to structural similarity*. *IEEE Transactions on Image Processing, 13*(4), 600–612. (Original SSIM paper)
2. **Hore, A., & Ziou, D. (2010).** *Image quality metrics: PSNR vs. SSIM.* *20th International Conference on Pattern Recognition (ICPR)*, 2366–2369.
3. **Huynh-Thu, Q., & Ghanbari, M. (2008).** *Scope of validity of PSNR in image/video quality assessment.* *Electronics Letters, 44*(13), 800–801.

Segmentation Metrics (IoU / mIoU)

4. **Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010).** *The Pascal Visual Object Classes (VOC) Challenge.* *International Journal of Computer Vision, 88*(2), 303–338. (mIoU originated from VOC challenge)
5. **Jaccard, P. (1912).** *The distribution of the flora in the alpine zone*. *The New Phytologist, 11*(2), 37–50. (Original Jaccard Index – basis for IoU)