TRANSPORTATION POOLED FUND PROGRAM
QUARTERLY PROGRESS REPORT

Lead Agency (FHWA or State DOT): Kansas DOT

INSTRUCTIONS:
Lead Agency contacts should complete a quarterly progress report for each calendar quarter during which the projects are active. Please provide a project schedule status of the research activities tied to each task that is defined in the proposal; a percentage completion of each task; a concise discussion (2 or 3 sentences) of the current status, including accomplishments and problems encountered, if any. List all tasks, even if no work was done during this period.

Transportation Pooled Fund Program Project # *(i.e, SPR-2(XXX), SPR-3(XXX) or TPF-5(XXX)* **TPF-5(535)**	**Transportation Pooled Fund Program - Report Period:** ☐Quarter 1 (January 1 – March 31) ☒Quarter 2 (April 1 – June 30) ☐Quarter 3 (July 1 – September 30) ☐Quarter 4 (October 1 – December 31)

TPF Study Number and Title:
TPF-5(535): Human-centered Steel Bridge Inspection enabled by Augmented Reality and Artificial Intelligence

Lead Agency Contact: **David Behzadpour**	**Lead Agency Phone Number:** **785-291-3847**	**Lead Agency E-Mail** David.Behzadpour@ks.gov
Lead Agency Project ID: Click or tap here to enter text.	**Other Project ID (i.e., contract #):** Click or tap here to enter text.	**Project Start Date:** 10/1/2024
Original Project Start Date: **10/1/2024**	**Original Project End Date:** **9/30/2027**	**If Extension has been requested, updated project End Date:** Click or tap to enter a date.

Project schedule status:

☒On schedule	☐On revised schedule	☐Ahead of schedule	☐ Behind schedule

Overall Project Statistics:

Total Project Budget	Total Funds Expended This Quarter	Percentage of Work Completed to Date
$600,000	$15,926	18%

Project Description:

The main objective of this proposed research is to provide state DOTs practical tools for supporting human-centered steel bridge inspection with real-time defect (e.g., fatigue cracks and corrosion) detection, documentation, tracking, and decision making. The proposed research will not only bridge the gaps identified in the IDEA project, but also expand the existing capability by developing AI algorithms for crack and corrosion detection. In addition to AR headsets, the project will also develop AR-based inspection capability using tablet devices. The tablet device can be used to perform AR-based inspection directly in a similar way to the AR headset. It can also leverage Unmanned Aerial Vehicles (UAV) for remote image and video acquisition during inspections, enabling bridge inspections from a distance in a human-centered manner.

Progress this Quarter
(includes meetings, work plan status, contract status, significant progress, etc.):

1. Task 1: CV and AI algorithms for crack and corrosion inspection

Building upon the hybrid framework introduced in the initial report—combining transfer learning and domain adaptation—we have implemented an active learning-based pipeline to further reduce dependency on large-scale manual annotation for crack and corrosion segmentation. In our earlier setup, corrosion was treated as a single class; however, in the current implementation, we have expanded the label taxonomy to include three severity levels: Fair, Poor, and Severe, enabling more granular damage assessment.

Additionally, while the previous approach relied exclusively on public datasets, we now incorporate a very small portion of labeled domain-specific data to better align the model with the visual feature distributions of the target dataset. This minimal supervision dataset facilitates partial transfer of domain-specific features into the training process and improves the model's generalization to field conditions.

The updated workflow (Figure 1) integrates a confidence-based active learning loop, where a model trained on public and minimally labeled domain data is iteratively fine-tuned using high-confidence pseudo-labels generated from unlabeled target-domain images (Texas bridge dataset).

As shown in the flowchart, the pipeline begins with a small, labeled set and public annotated data to fine-tune a YOLOv12 model pretrained on large-scale damage datasets. This initial model performs inference on a large portion of unlabeled images, and a confidence thresholding mechanism is applied to extract reliable pseudo-labels. These are merged into the training set and used to retrain the model in iterative cycles. The process continues until a predefined performance threshold (e.g., plateau in mAP50) is achieved on the validation data.

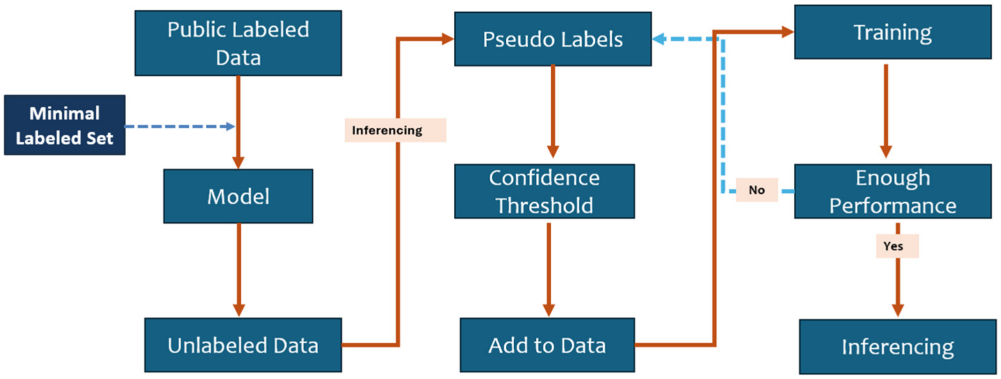


Fig 1. Active learning loop for crack and corrosion segmentation

This active learning approach substantially improves labeling efficiency, reduces reliance on fully annotated domain data, and progressively integrates domain-relevant features through pseudo-label refinement. The use of the YOLOv12 backbone ensures real-time inference capability while maintaining high segmentation accuracy, benefiting from recent architectural advancements such as R-ELAN, attention modules, and area-based feature aggregation.

2. **Task 2: AR-based software for human-centered bridge inspection**
Development of the AR infrastructure inspection tool has continued and progressed steadily. The transition from HoloLens to Magic Leap 2 platform has been going smoothly, with all features previously implemented working as expected on the new headset. In addition to preserving core functionality, a series of quality-of-life improvements have been integrated into our application to enhance the convenience of use during field inspections. Some progress has been made with spatial persistence, particularly in the anchorage of the Magic Leap's spatial mesh. However, further development is necessary to ensure that inspection inference results remain anchored and stable across sessions/inspections.

In terms of the second approach based on UAV and tablet devices, we have investigated the ability of an RBG-D camera to reconstruction the 3D environment. This ability is critical as the 3D environment allows accurate anchorage of holograms during the human-centered bridge inspection.

Anticipated work next quarter:

1. **Task 1: CV and AI algorithms for crack and corrosion inspection**
Integrating additional publicly available data and incorporating a small portion of labeled domain-specific samples has enhanced the model's ability to generalize to real-world field conditions. The active learning framework developed in this task enables the generation of high-confidence pseudo labels, which are iteratively added to the training set to improve model performance through self-supervised refinement.

In future stages, we will focus on automatically setting an optimal confidence threshold to select only the most reliable inferences such as pseudo labels. This thresholding strategy will help maximize training quality while minimizing the introduction of noisy labels, allowing the model to gradually improve its segmentation accuracy across successive learning loops. This pipeline ensures scalability, label efficiency, and improved adaptation to challenging visual conditions found in field inspections of cracks and corrosions.

2. **Task 2: AR-based software for human-centered bridge inspection**
During the current reporting period, we plan to complete our implementation of the Magic Leap's spatial persistence for all inference outputs and inspection results. We will focus efforts on resolving issues with Magic Leap video capture while demonstrating the applications' features. Additional research will be done to support the development of a handheld or drone-based inspection tool in the future. Once all headset features are completed, particularly spatial persistence, we will begin work on a database to store inspection results.

Significant Results:

1. **Task 1: CV and AI algorithms for crack and corrosion inspection**

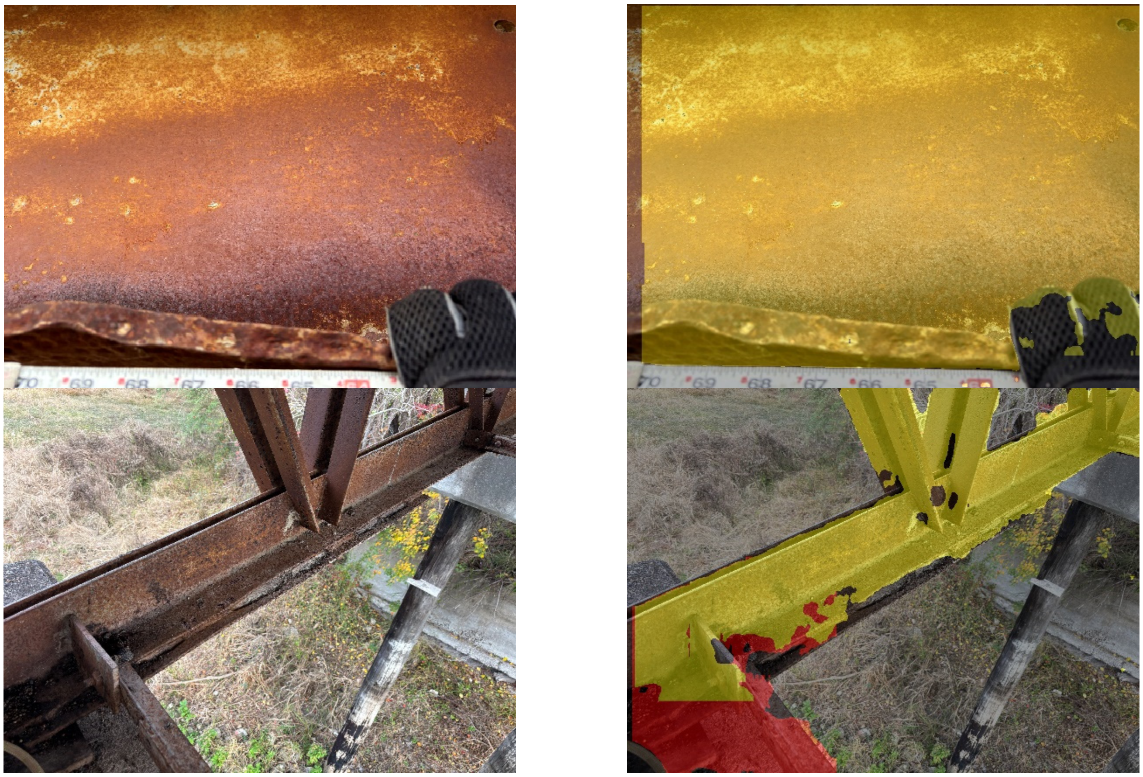
To initiate model development, we curated a training dataset comprising both publicly available and domain-specific labeled data. For crack segmentation, 1,040 annotated images were sourced from public datasets, supplemented with 5 manually labeled images from the Texas bridge dataset to inject domain-specific context. For corrosion detection, we used 430 publicly labeled images and incorporated an additional 10 labeled images from the Texas dataset to improve domain adaptation.

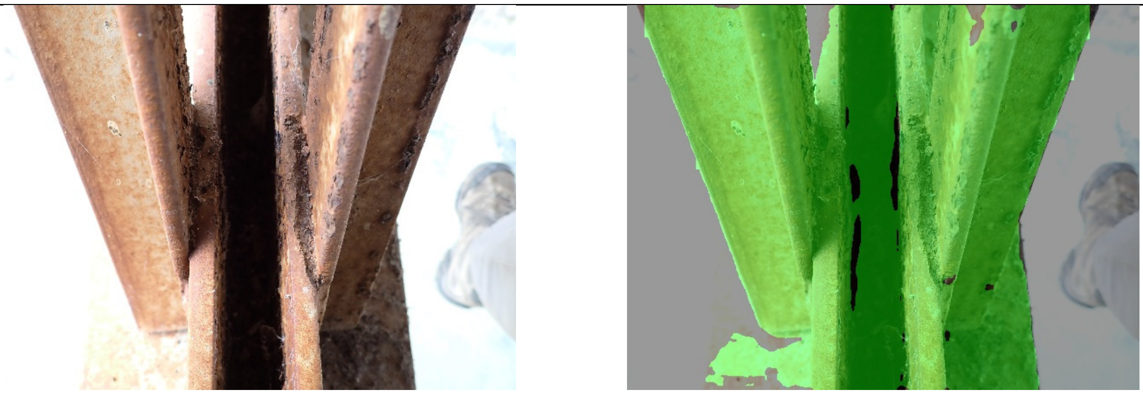
A standard 80/20 split was employed to partition the dataset into training and validation subsets to ensure unbiased performance evaluation. We trained and fine-tuned two separate YOLOv12 models independently for crack and corrosion segmentation tasks. This decoupled training approach allows the architecture to learn optimal features tailored to each type of damage. To further transfer structural and visual patterns specific to the Texas bridges, we manually labeled an additional 15 images and integrated them into the training set, enhancing the model's ability to generalize to real-world field conditions. The performance metric, mAP@0.5 for each damage type, is reported in Table 1.

Table 1. Performance Index for YoloV12

Damage Type	mAp50
Fair corrosion	0.15
Poor corrosion	0.16
Severe corrosion	0.05
Crack	0.25

After training, the models were deployed to perform inference on the full unlabeled Texas dataset. Sample segmentation results for corrosion and crack identifications are illustrated in Figures 2 and 3, respectively. Here, fair, poor and severe corrosions are highlighted in green, yellow, and red colors, respectively.





Images

Inferences

Figure 1. Inference Results for Corrosion Detection

As demonstrated in the inference results, the trained model can identify both the location and severity level of corrosion on previously unseen data with reasonable accuracy. The multi-class segmentation (Fair, Poor, Severe) enables detailed damage assessment. However, the model occasionally produces false negative inferences, particularly for low-contrast or small-area corrosion regions. These errors can be mitigated by refining the training dataset through the active learning pipeline, where high-confidence pseudo-labeled outputs are manually reviewed, corrected if necessary, and incorporated into subsequent training iterations. This iterative process is expected to enhance the model's robustness and reduce misclassification in challenging scenarios.

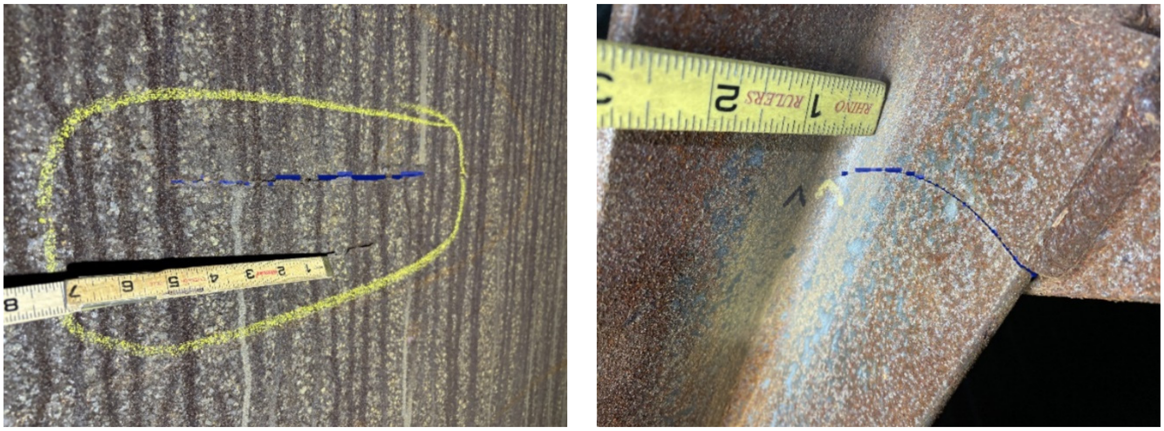


Figure 2. Crack Detection inferences

A similar trend was observed in the crack segmentation task, where some regions were left undetected due to false negative errors as shown in Fig 2. This issue arises primarily because crack pixels constitute a very small fraction of the total image area, making them harder to learn and detect reliably. To address this imbalance, it is necessary to augment the training data with additional crack examples and leverage pseudo-labeled samples generated from the unlabeled dataset.

At the current stage, pseudo labels have been generated; however, incorporating all of them into the training pipeline would introduce noise and degrade model performance. Therefore, a selection strategy is needed to filter out low-quality predictions. Specifically, we propose selecting pseudo-labeled samples that exhibit minimal false positives and false negatives by setting an optimal confidence threshold and evaluating the predictions using Intersection over Union (IoU). By prioritizing pseudo labels with high IoU and strong confidence scores, the training dataset can be refined iteratively to improve model accuracy and robustness.

2. Task 2: AR-based software for human-centered bridge inspection

Subtask 2.3: AR software environment for AR headset

Migration from the HoloLens 2 to the Magic Leap 2 platform has been completed with all features functioning as intended on the new hardware. With this, we can now turn to implementing new features into our AR inspection tool. Several changes were made to the functionality of the user interface panel seen in Fig. 3. Functionality has been given to the two buttons outlined in red, and the functionality of the exit button has been altered for the comfort of the AR application user.

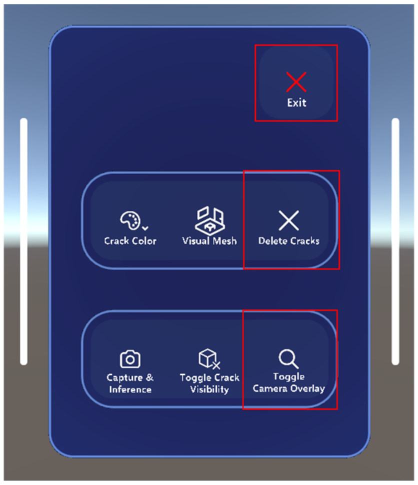


Figure 3. User interface with updated functionality

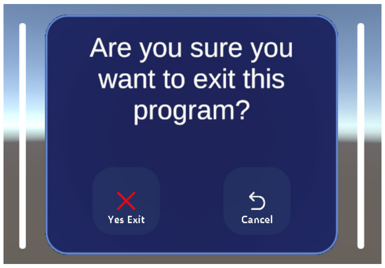

Are you sure you want to exit this program?

Figure 4. Confirmation of exit message

As we were developing the application, we found that it was very easy to accidentally press any button on the user interface, including the exit button. This was not only frustrating to development, as it would interrupt our tests as we work on the program, but this could be catastrophic to active human centered inspection, as a user could easily close the program unwillingly and without realizing. This oversight could cost several hours of work and inspection to be lost if not saved properly beforehand. To solve this, a simple addition was made. The exit button no longer immediately quits the program, instead it will prompt the user, as seen in Fig. 4, and ask them if they are certain they want to quit and allow them the option to cancel this decision.

Additionally, in Fig. 3, you can see that the ”delete cracks” and the ”toggle overlay” buttons are outlined in red. These are also new additions and have recently been given functionality. The “delete cracks” function was planned to be a tool to allow the inspector to delete projections of images/inferences of cracks that they had seen. This would serve useful if the inspector took a bad photo or wanted to redo inspection of a specific surface. We thought about how to properly implement this functionality, as a mere “delete all” button would not be very useful and would force users to start over on all inspections. Instead, inspectors needed to be able to delete specific inspections/projections so as to not have undesired deletions.

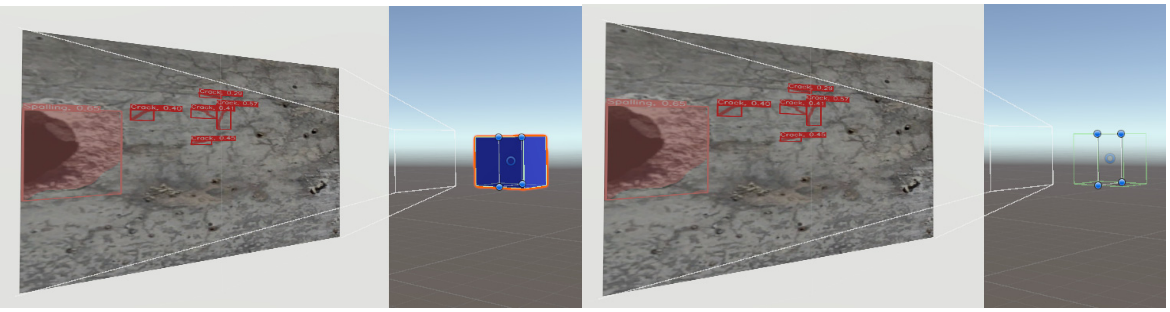


Figure 5. Left - Projectors before this update are visible and can be distracting. Right - Projectors when hidden.

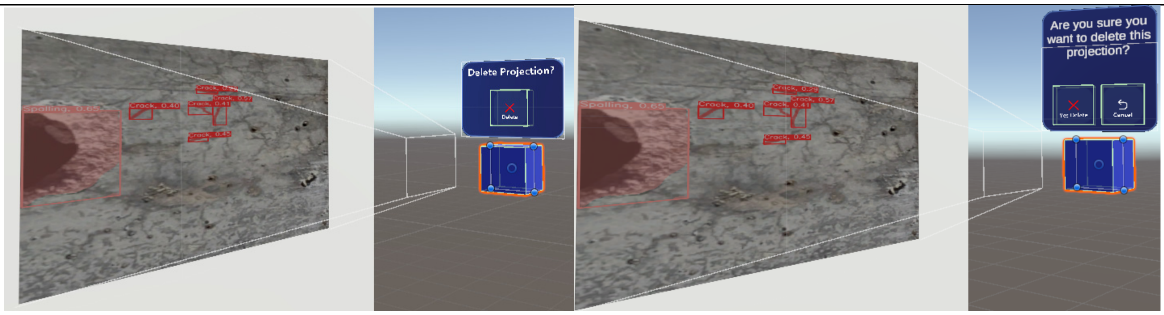


Figure 6. After pressing ”delete cracks” inspectors can now interact with projectors and are able to delete them.

To implement this, we created a system that works as such and can be explained using Figs. 5 and 3. In Fig. 5, on the left you can see a projector and its projection as they were shown to the user/inspector before this new change. The projector is visible as a blue box from which the inference is projected onto a wall that the inspector can see. On the right, you can see that we have hidden the projector from the inspector's view by default. This helps declutter the inspection work and allows the user to get a better view so that the holograms do not end up hampering inspection. This is the default state of the program. Images can be taken, inferences can be made, and results are shown to the inspector, while projectors are hidden in the background. Fig. 6 illustrates what happens upon pressing the “delete cracks” button. Each individual projector is made visible to the inspector, and the inspector is then allowed to interact with the projector (the user may want to slightly alter the location of the projector if it has become slightly misaligned with the crack/fatigue that it is imaging) and also given the option to delete it. To make sure no accidental deletions are made, the user must confirm their decision to delete the projector.

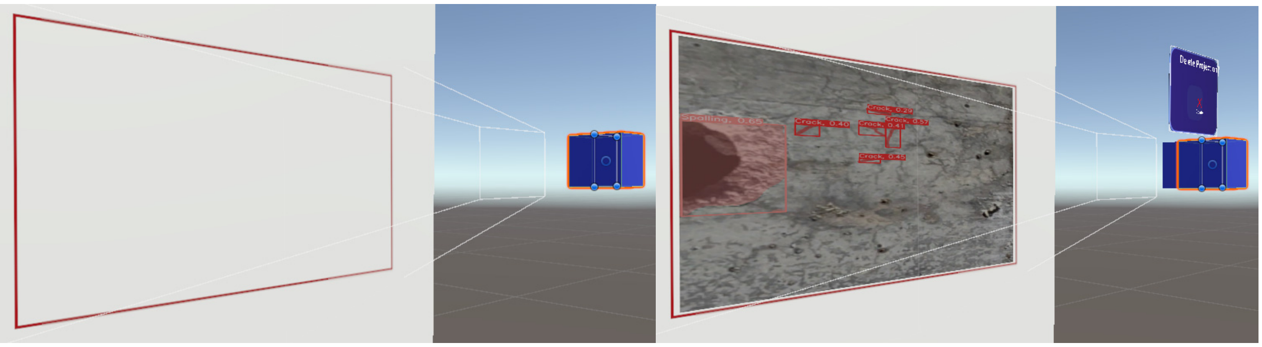


Figure 7. Image overlay demonstration. Left – projector attached to user’s head showing where images will be taken from and where they will be projected to. Right – after image is taken and inference is made projections are within bounds of the overlay.

As for the” toggle overlay” button, this is an option that was implemented to allow the inspectors more clarity over what they are imaging and inspecting. As we developed the program, we noticed that it was not easy to predict exactly where you were imaging simply by looking at it, and we sought to make it more obvious to the user what they were framing before they took an image for inference and inspection. To solve this issue, we implemented the ”toggle overlay” feature. To allow the user to see the dimensions and borders of the image they are going to take before they take it, we attach a projector to the main camera of the unity scene (which aligns with the headset, or inspector’s head in the real world) and project a red rectangle with the same dimensions and aspect ratio as the camera that will be taking images for inspection. This can be seen in Fig. 7, where on the left you can see the projector outlining where the projection will land once an inference is made on an image, and on the right, you can see how the projections align with each other's shape and size. Please note, the projector attached to the head of the inspector is not itself visible by default as this would impair the user’s vision. We have made all projectors visible for the sake of demonstration here.

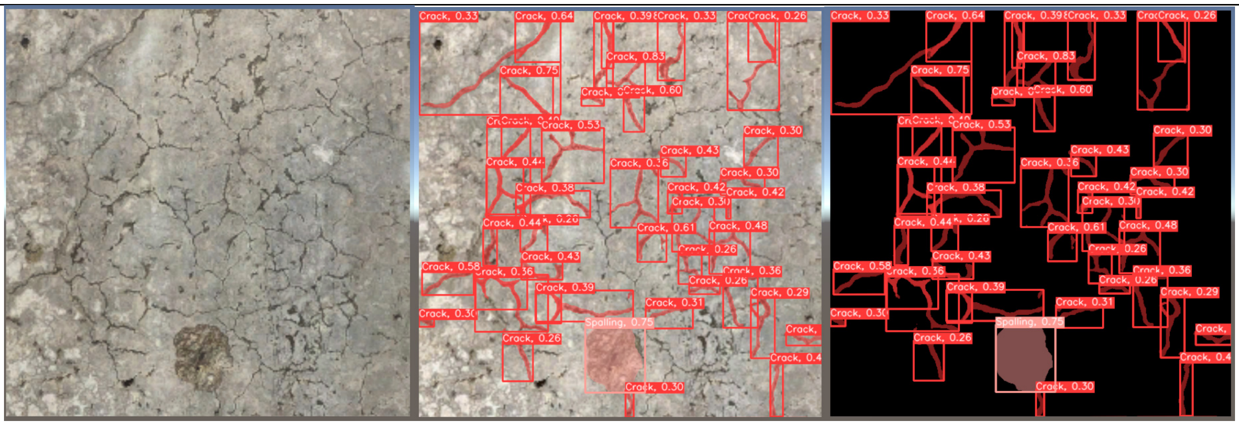


Figure 8. Left – example crack image. Middle – example of output with background image. Right – example of output with only inference mask.

Another alteration has been made to the inferences and their projections. Previously, when images were taken and inferences were made, the inspection software would return an image that contained both the background image of what was taken by the camera, as well as the inference of the crack overlayed as a mask on the image. This can be seen in the middle image of Fig. 8. This could sometimes be disorienting to the user, as they would see both the physical object in front of them, as well as a projection of an image of that object overlayed on top of the object itself. To fix this we have altered the inference/projection pipeline to instead allow us to only project the mask of the cracks and fatigue themselves (as seen on the right in Fig. 8), and place those directly on top of the physical structure so the user isn't forced to see features and objects duplicated on top of one another.

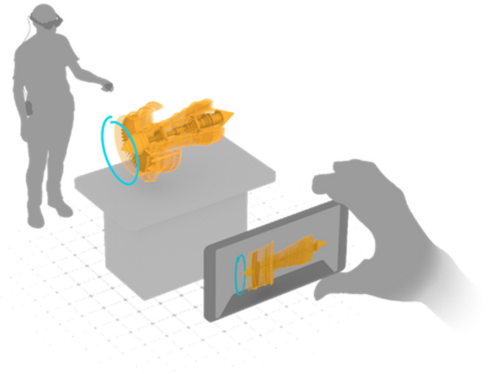


Figure 9. Magic Leap Spectator illustration
https://developer-docs.magicleap.cloud/docs/guides/features/magic-leap-spectator/ml-spectator/

While developing our tool for Magic Leap 2, we ran into some issues with documentation of our application. It seems it is quite complicated for the magic leap to allow you to record video of the user's view while that user is currently in an application that uses the headset camera. This issue is not unresolvable but is still currently being worked on, as we have been communicating back and forth with the development support team at Magic Leap. However, while looking into solutions for this issue, our team discovered a feature known as Magic leap Spectator. Magic Leap Spectator is a plugin for the Magic Leap headset that allows users to pair a mobile device to the headset and allow mobile users to view the hologram content produced by the headset as seen in Fig. 9. Our team will be exploring this option not only for demonstration purposes but also to allow multiple inspectors to view inspection inferences and results in real time using only one headset.

Subtask 2.4: AR software environment for tablet device and UAV

To enable 3D awareness of the bridge environment for UAV-based inspections, we investigated an Intel D455 RealSense Depth Camera. The RGB-D camera captures synchronized and spatially aligned color and depth information, providing a depth value for each pixel in the image. This capability makes it highly suitable for bridge structural monitoring, where both surface appearance and geometric information are critical. From the captured data, surface and structural features can be extracted to support the detection of damage, deformation, or other anomalies.

The motivation for using an RGB-D camera, such as the Intel RealSense D455, lies in its ability to simultaneously acquire high-resolution visual and depth data in real time. This is particularly valuable for drone-enabled, AR-based bridge inspections, where accurate 3D models of the structure are essential for anchoring holographic overlays. By generating dense and geometrically consistent 3D meshes from aerial RGB-D scans, virtual annotations or damage markers can be precisely aligned with real-world bridge components in augmented reality applications. The D455 offers several advantages for this context, including a depth range of up to 6 meters, a global shutter for both RGB and depth sensors to reduce motion blur during drone flight, and a field of view of 86° × 57°. Additionally, its depth accuracy of less than 2% at 4 meters ensures that the reconstructed 3D models maintain sufficient fidelity for visualization and measurement tasks. These specifications, combined with its compact form factor and robust hardware synchronization, make the D455 an effective choice for onboard drone inspection and real-time AR anchoring.

An example of the raw data collected by the camera is illustrated in the top two figures in Fig. 10: the top-left shows the RGB image, while the top-right displays the corresponding depth map. Using the camera's intrinsic parameters, such as focal length and principal point, a colored 3D point cloud can be reconstructed, as shown in the bottom-left figure. Furthermore, a 3D textured mesh can be generated from the RGB image and depth information, as demonstrated in the bottom-right figure, offering a more complete and interpretable representation of the structure.

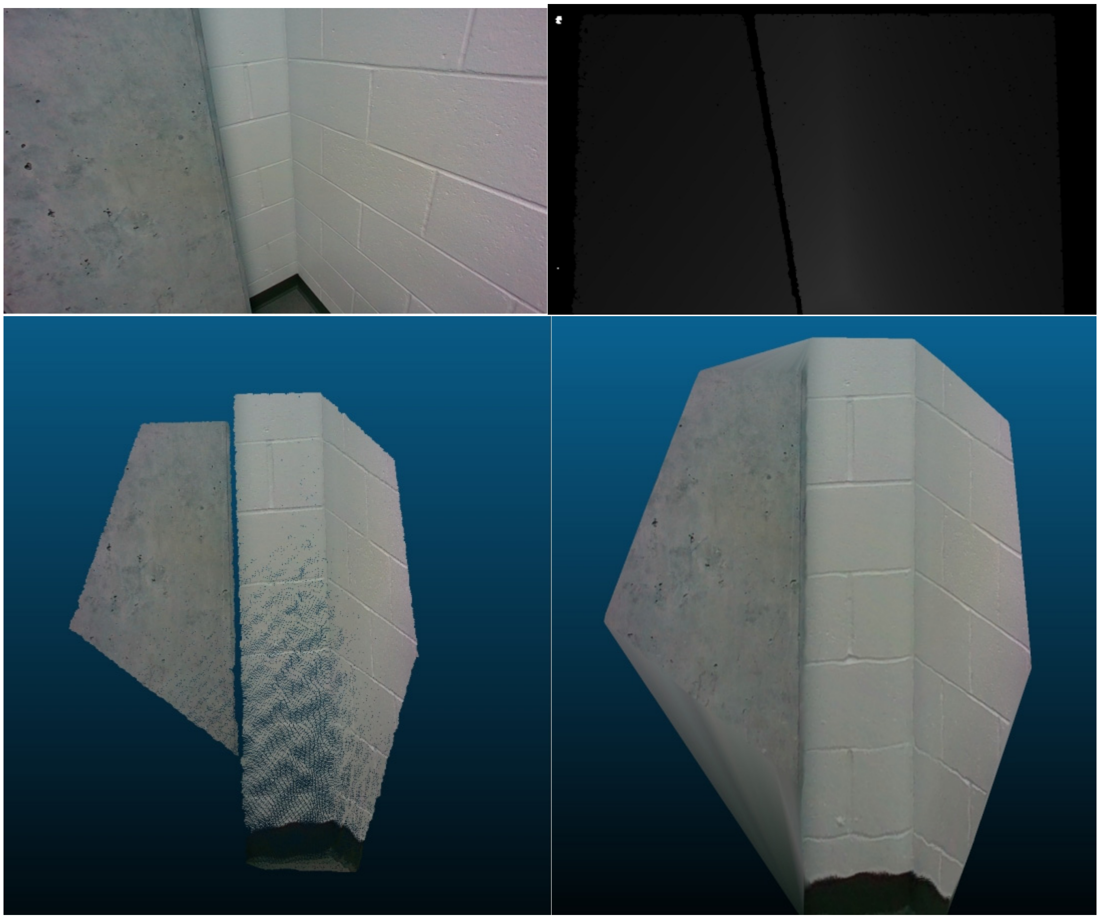


Figure 10. An example of an RGB image (top left) and a depth map (top right) with constructed point cloud (bottom left) and 3D mesh (bottom right) in a lab room

Circumstance affecting project or budget. (Please describe any challenges encountered or anticipated that might affect the completion of the project within the time, scope and fiscal constraints set forth in the agreement, along with recommended solutions to those problems).

N/A

Potential Implementation: