



Nemak - Feasibility Study: Defect Recognition for Automotive Parts

Prototype development of an integrated object recognition system for quality control of engine blocks

DropsLab Technologies S. à r.l.
2, rue Jean Steichen
L-5868 Alzingen
Luxembourg
www.dropslab.com

Authors: Felix Saretzky, M.Sc.
Date: 19.03.2024

1. Motivation and Background

After several meetings and a visit by Nemark at Dropslab in Luxembourg, the following work packages were prepared. The working method was derived from the work packages listed in the table below.

Work Package Overview	Person Days
WP 1: Requirements Analysis and Conceptualization • Joint definition of requirements and detailed documentation of the current processes. Development and detailing of the target processes using assistance systems with suitable hardware such as smartphone and glasses. • Investigation of the classification process in practice and selection of appropriate reference objects (definition of class/item number) and definition of acceptance criteria. • Analysis of potential inspection processes for the reference objects and the occurring phenomena. • Specification of requirements and continuous alignment of work.	1
WP 2.a: Reduced ML Classification Pipeline • Analysis and evaluation of suitable ML classification algorithms / ML object detectors.	3
WP 2.b: Development of the reduced ML model • Creation of a reduced, trained ML model and evaluation of the system on a reference case using ML classifiers and laboratory testing.	2
WP 3: Prototype Implementation of Object Detection • Conceptualization and implementation of interfaces. • Evaluation of technology and workflow for building a semantic reference database (identifiers, serial numbers, labels) for workflow digitalization. • Evaluation of object detection approaches. • Integration of object detection and component classification into workflow support.	2
WP 4: Evaluation & Presentation of Results • Conceptualization of demonstrators (Recognition Demonstrator, Workflow Demonstrator, Decision Support Demonstrator) • Procurement of hardware and software and construction of the demonstrators. • Validation of AI-based recognition • Field study: Prototypical testing, technical optimization, and expansion of the overall system with defined reference parts and cases. • Meeting for presenting the proof of concepts and showcasing the implemented solution based on use cases. • Preparation of a report	1

Table 1: Work Packages

2. Objective

The project aims to explore the feasibility of implementing an artificial intelligence (AI)-based system for recognizing defects in engine block parts. This includes analyzing the requirements, designing the system, developing a machine learning (ML) pipeline, creating a reduced ML model, and integrating object recognition into Dropslab Plex to support operators for improved quality control in automotive production.

3. Criteria

Nemak provided Dropslab with 4 parts including, 2 structural parts and 2 engine blocks.

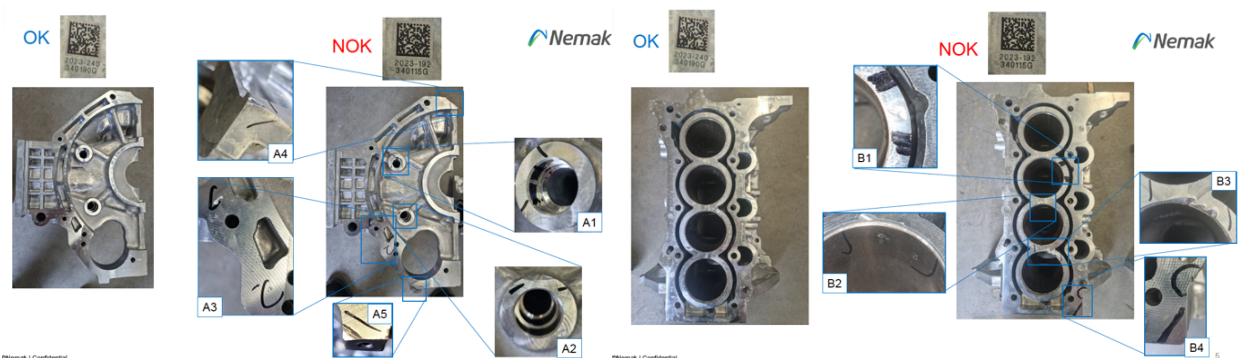


Figure 1: Requirements from Nemak

The aim is to develop a prototype that has a "smart tag" under realistic conditions on a hardware end device such as a smartphone (IOS, Android) or smart glasses (RealWare) within a workflow, which enables object detection to be carried out. The object detection should recognize the criteria described in the *Setup* chapter and mark them with a bounding box. The prototype is to be tested on the engine blocks provided by Nemak. The performance of the system will be validated via visual assessment by the Dropslab QC team.

4. Setup

To create a realistic production line environment, the engine blocks were placed on workbenches in our test laboratory. The hall has normal lighting conditions.



Figure 2: Engine Blocks OK and NOK

Within this setup, image data of the engine block was acquired.

The image below shows the integrated system architecture in the Dropslab Core and Sense Platform.

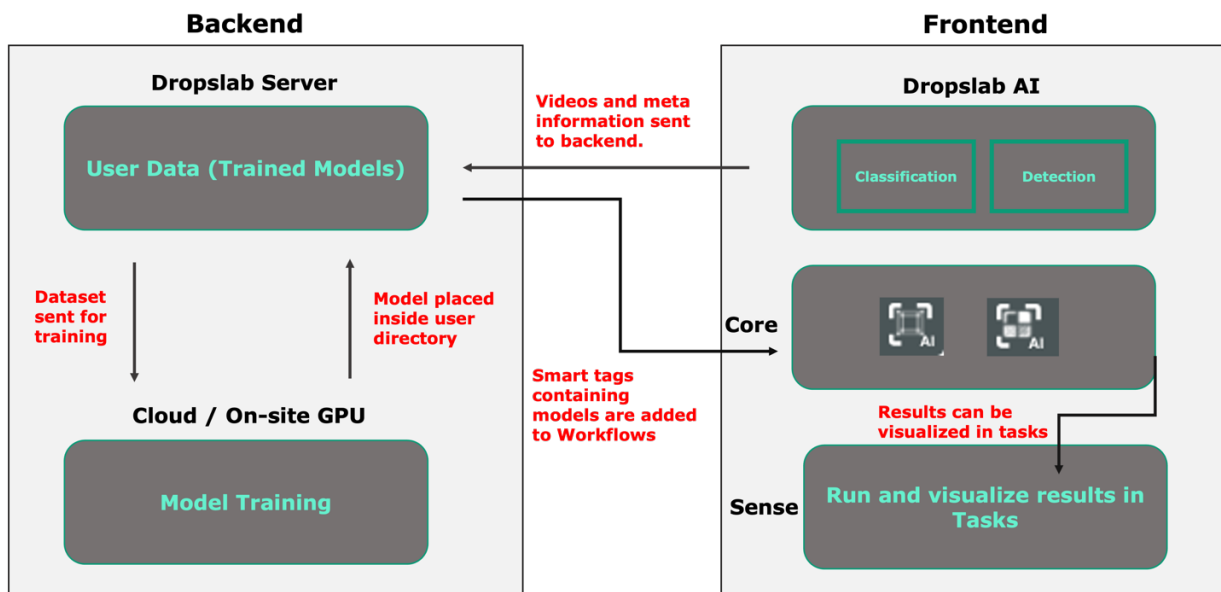
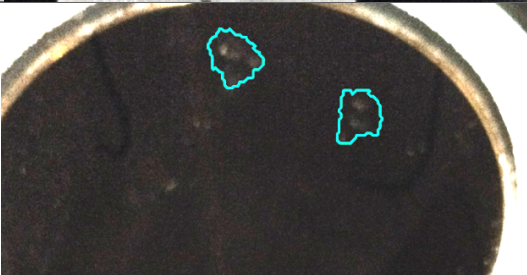


Figure 3: System Architecture

5. Implementation of AI-based Defect Detection

In this section, we outline the process of creating a dataset, manually annotating the data with polygons, applying data augmentation techniques, and training various object detection models for our specific use case. To develop an object detection system, we curated a dataset consisting of 200 images relevant to our task. The dataset was carefully collected to ensure diversity in object appearances, poses, and backgrounds. These images served as the foundation for training and evaluating our object detection models.

Object Detection Classes based on the Nemak Requirements

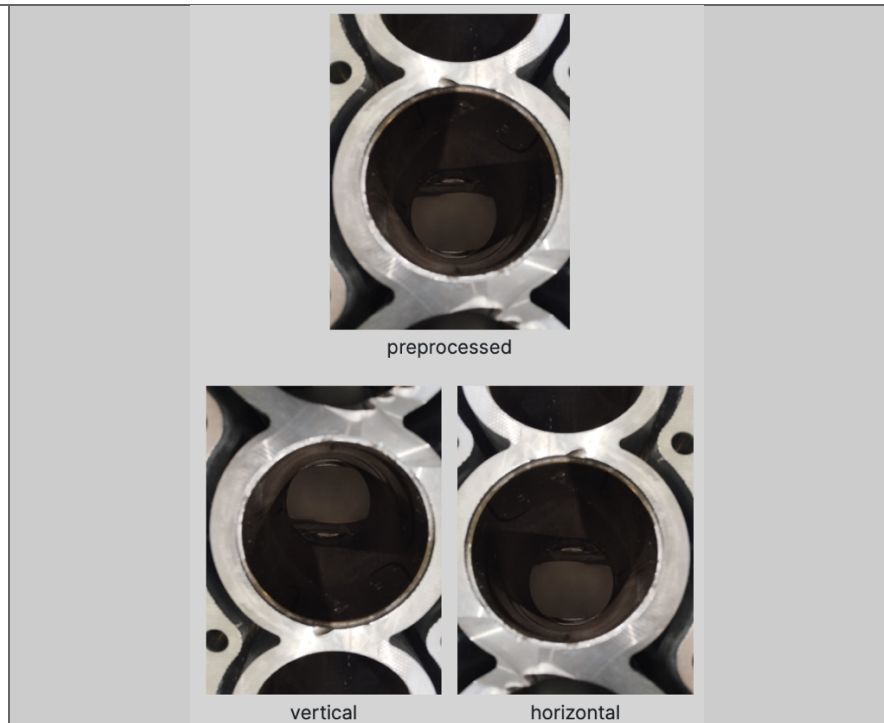
<i>Chipped Surface</i>		Portions of the metal's outer layer have been fragmented or broken off, leaving irregularities on the surface.
<i>Damaged Surface</i>		The metal's outer layer exhibits visible impairments or deformations, potentially affecting its structural integrity.
<i>Dented Edge</i>		The edge of the metal part shows concave or uneven depressions, likely caused by impact or pressure.
<i>Notched Edge</i>		The edge of the metal part displays irregular grooves or cuts, possibly resulting from cutting or abrasion processes.

We manually annotated the dataset using the polygon annotation method. This approach involves drawing precise polygonal shapes around each object of interest, ensuring that the annotation closely follows the object's contours. Polygon annotations offer several advantages over other annotation methods, such as bounding boxes. Given the limited size of our initial dataset, we employed data augmentation techniques to artificially increase the dataset size and introduce more

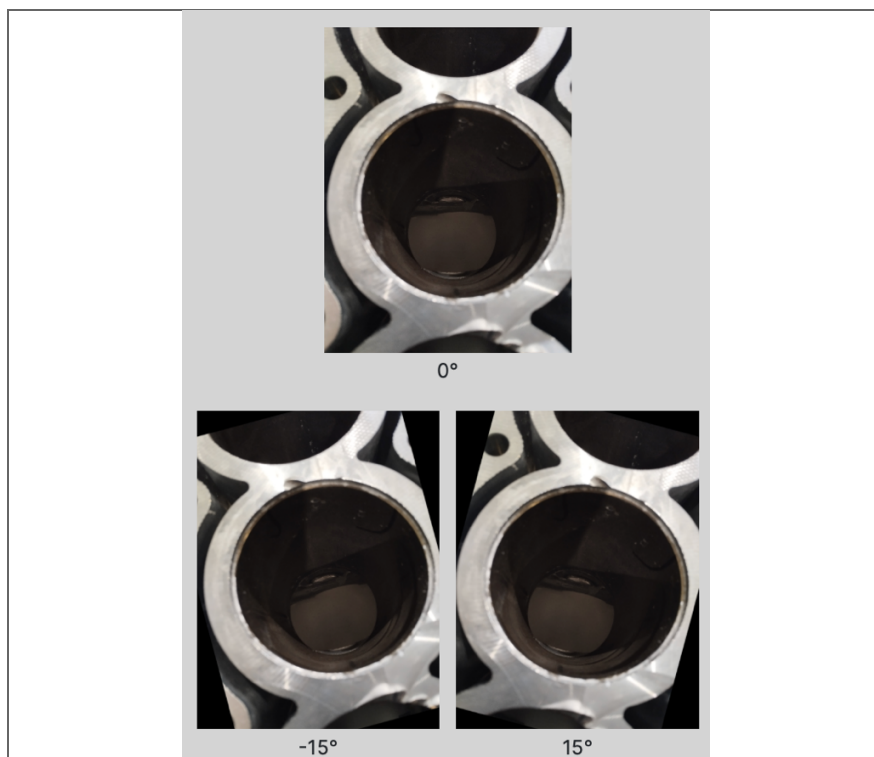
variability. Data augmentation generates new training samples by applying random, yet realistic, transformations to the original images. The applied transformations are explained in the table below. These augmentations significantly increased the dataset size and provided the models with a wider range of object appearances and contexts, helping them to generalize better and perform well on unseen data.

Segmentation Methods

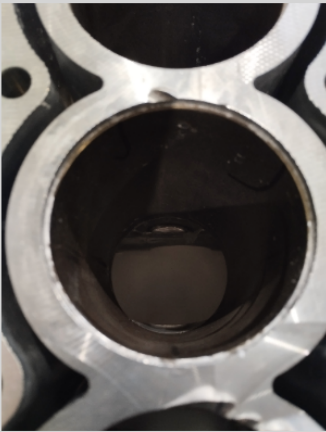
Flip (horizontal, vertical)



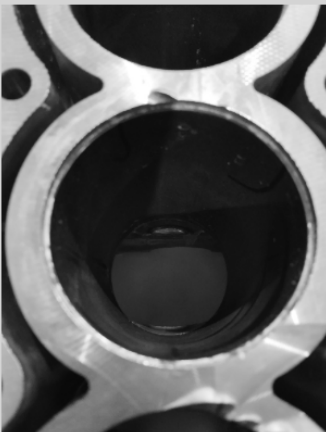
Rotation (-15°; + 15°)



Grayscale (applied to 12%
of all images)



original

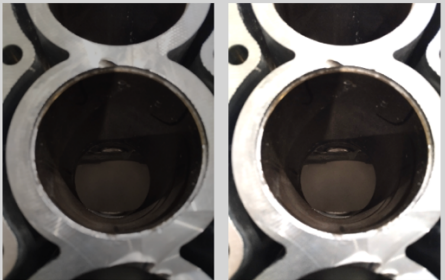


grayscale

Brightness (-25%; +25%)



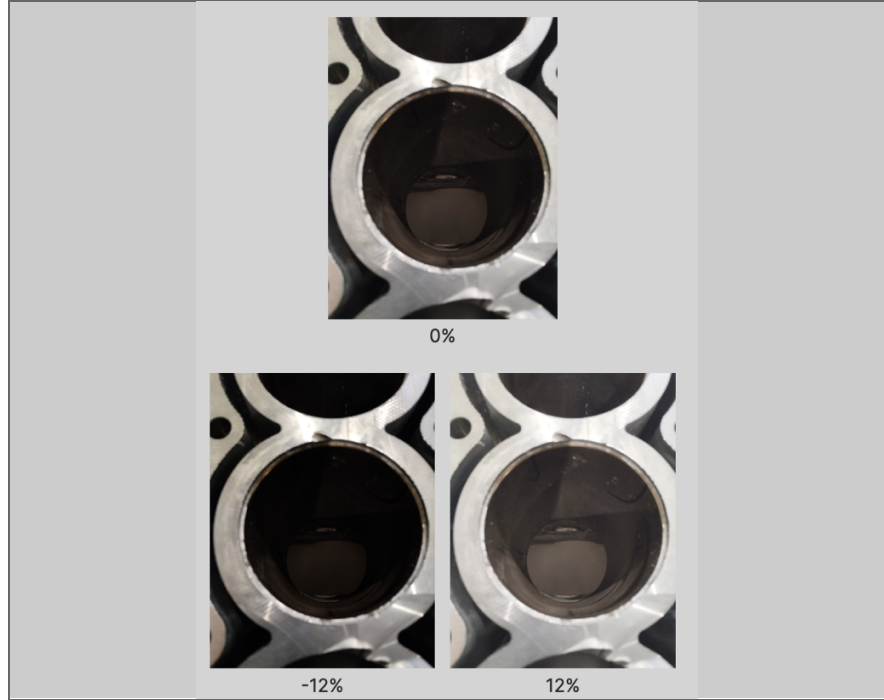
0%



-25%

25%

Exposure (-12%; +12%)



We trained multiple object detection models on our augmented and annotated dataset to identify the best-performing architecture for our specific use case. The models were selected based on their performance and inference speed. Each model was trained using appropriate hyperparameters and optimizers to ensure optimal performance. The trained models were then evaluated on a separate validation set to compare their accuracy, speed, and resource utilization. The following plot shows the training performance of the best model.

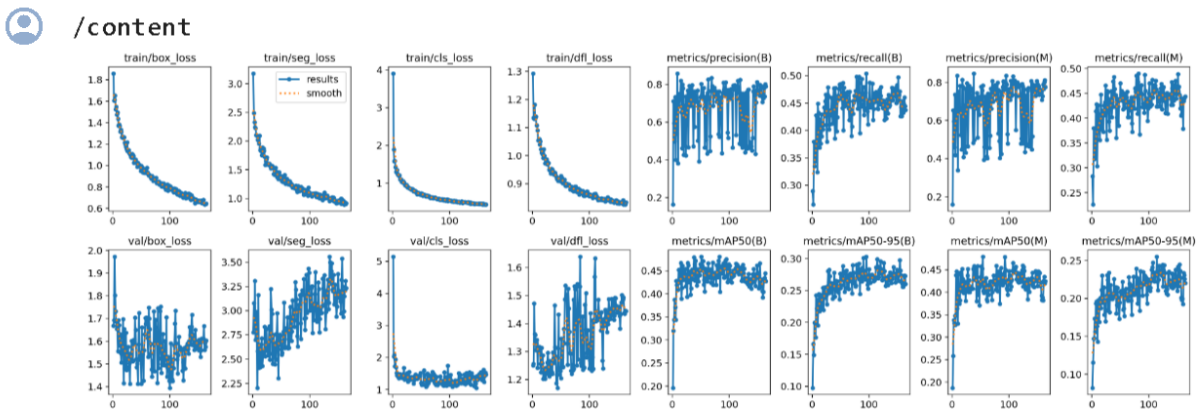


Figure 4: Summary of the Training Performance

Training Performance

Overall Performance	Precision (P)	Recall (R)	mAP50 (mean average precision at IoU 0.5)	mAP50-95 (mean average precision at IoU 0.5-0.95)
	92.9%	83.3%	92.7%	51.1%
Per Class Metrics	Chipped Surface	Damaged Surface	Dented Edge	Notched Edge
	P=90.0%, R=100.0%, mAP50=99.5%	P=96.0%, R=100.0%, mAP50=99.5%	P=85.5%, R=63.6%, mAP50=72.3%	P=100.0%, R=69.6%, mAP50=99.5%
Speed	Pre-process	Inference	NMS (non- maximum suppression)	Total
	2.5ms	38.6ms	110.3ms	151.4ms

Table 2: Performance of the Object Detection Training Process

6. Results

The results are split across 3 platforms. We have the application on an iPhone (IOS), Google Pixel (Android) and a smart glass (RealWear Navigator 500).

To execute a workflow on an end device, it must be created in the Dropslab Core Platform. To do this, a start and end point as well as steps are added in the "Workflow Designer" using the drag & drop function.

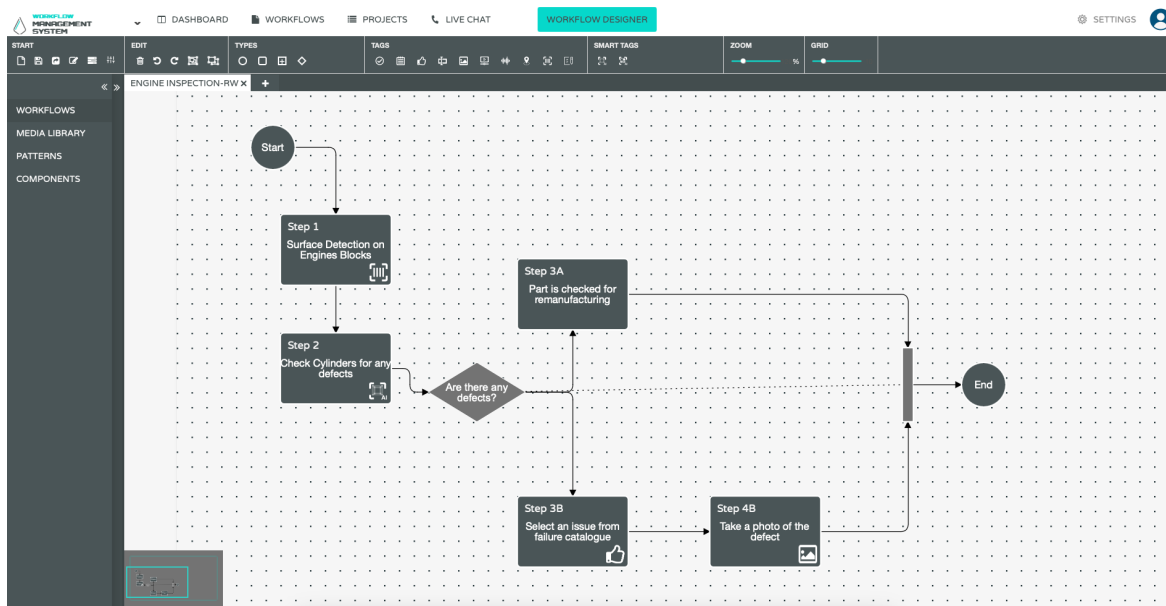


Figure 5: Workflow Designer Overview

The image *Figure 5: Workflow Designer Overview* shows how a workflow was created to perform object recognition. In the first step, a QR code is added as a tag. This ensures that the correct engine block workflow is executed, and the data results can be categorized. Step 2 contains the trained object recognition model. The result of the object recognition is used as input in a decision node. This means that if the quality control is OK, the "upper path" of the decision tree is automatically selected. If the quality control is NOK, the "lower path" is automatically selected. In the lower path are further steps like selecting defects from a catalog or taking a picture of the defect. The data recorded within a project is saved. These can be used to create an automated PDF report or to visualize the data in a graph.

All these settings are freely configurable in our Dropslab Core Platform. This makes the creation and execution of tasks flexible, modifiable, and quick to implement.

Listed below are 2 tables that display the step by step the results of object recognition on the end devices.

a. Smart Glasses – RealWear Navigator 500



Figure 6: Employee who wears smart glasses and sees the task in front of him.

Explanation

After the employee puts on the smart glasses and selects a task, they see this frame. They can navigate through the application using voice commands.

The employee selects the first step and receives instructions on what to do. If necessary, additional media such as images, technical drawings, 3D CAD files, and videos can be included for further assistance.

After the installation is understood, the employee navigates to object detection.

The employee takes a picture of the engine block.

The employee receives the result of the object detection.

RealWear Navigator 500

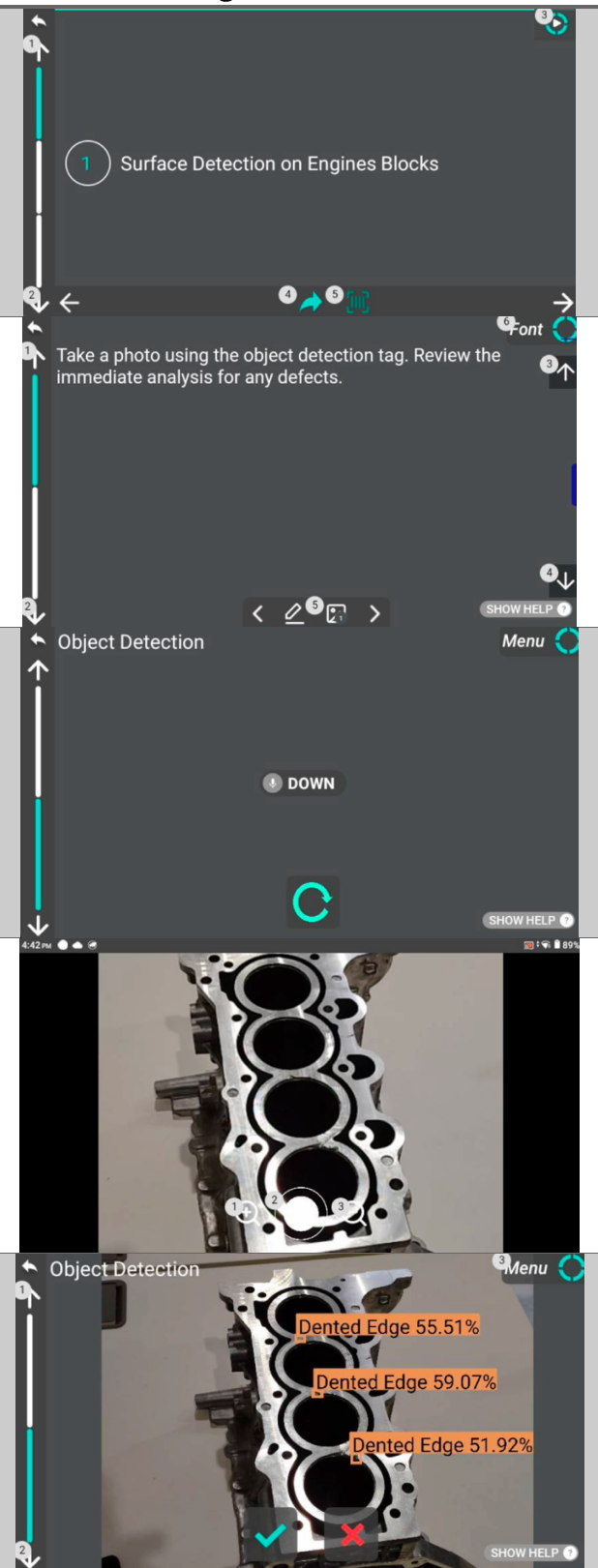
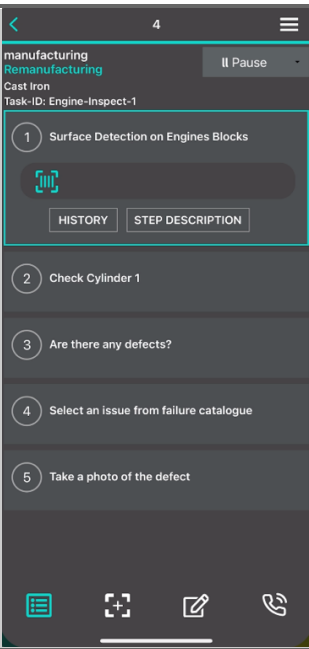
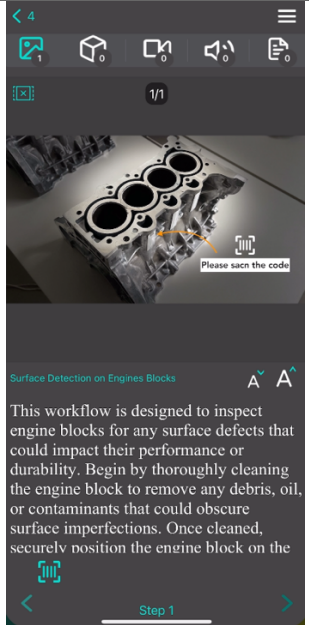
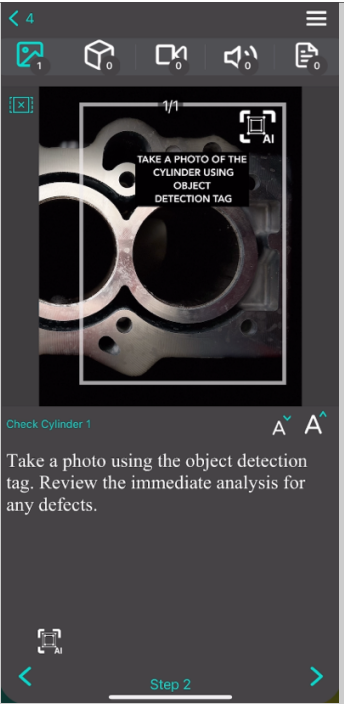


Table 3: Frame by Frame explanation of the object detection workflow executed on a RealWear Navigator 500

b. iPhone (IOS)

7. Explanation	iPhone 14 Pro
Once the employee has opened the application on their smartphone and selected a task, they will see this frame. In the first step, the "Engine-Inspect-1" task is selected (last one).	
The employee selects the first step and receives instructions on what to do. If necessary, additional media such as images, technical drawings, 3D CAD files, videos can be added as aids	

After the installation process is understood, the employee navigates to object detection.



The employee takes a picture of the engine block.



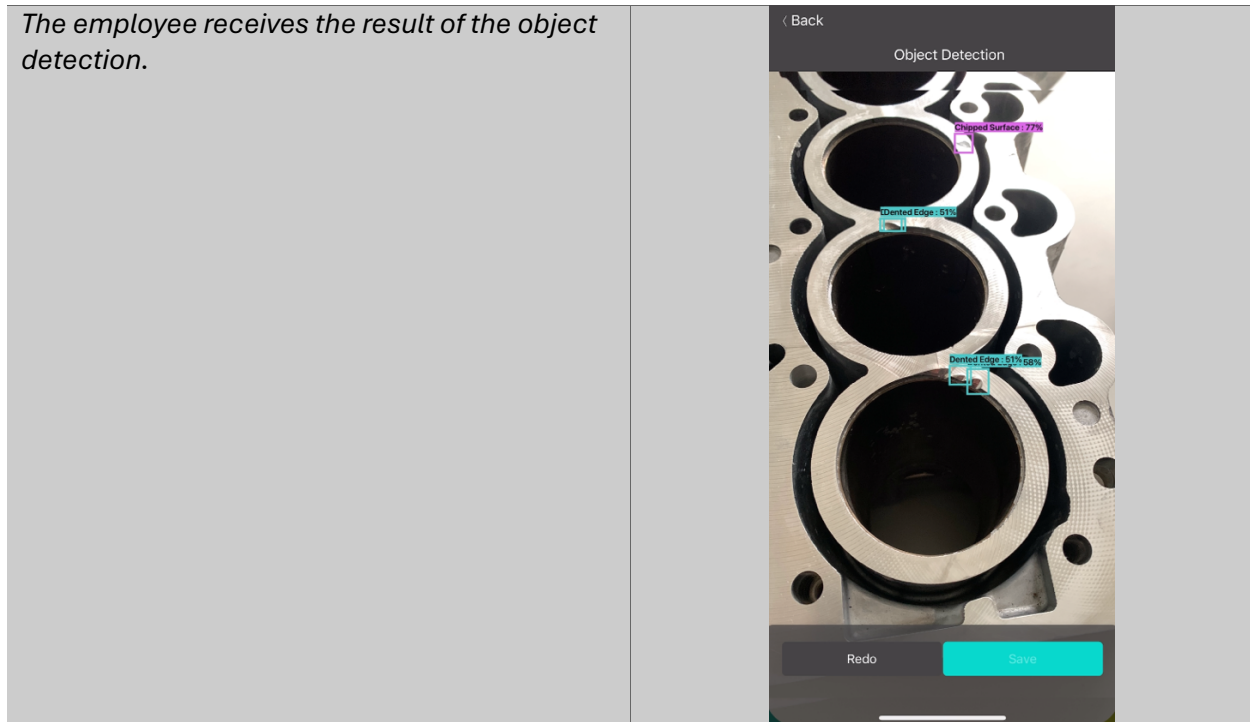


Table 4: Frame by Frame explanation of the object detection workflow executed on iPhone (To keep the report short, the table for Android is omitted, as it is visually and functionally the same).

Here you can find both tests as video link:

- [RealWear](#)
- [iPhone](#)

Future Work and Conclusion

To ensure the deployment of an object detection system in the production line at Nematik, a generalization of the object detection model must be validated. A large additional data set must be created which contain data from the production environment. Preferably 2 approaches are proposed:

- Mobile Solution (smartphone or smart glasses)
- Fixed camera setup with inference PC solution

After the creation of a larger data set, a new deep learning model is trained and evaluated. This is then tested on both hardware solutions with real products. Once positive generalization has been achieved, the product can be deployed in production. Work packages for a follow-up project are listed below.

Work Package Overview Future Project	Person Days
WP 1: Requirements Analysis and Conceptualization for Production Deployment at Nematik • Work together to determine requirements and document Nematik's current production line processes. Develop and improve target processes using	3

assistance systems with appropriate hardware such as smartphones, glasses, or fixed camera setups. - Investigate the object detection process in Nematik's production environment and choose appropriate reference objects (defining class/item number) along with acceptance criteria. - Identify potential inspection processes for the reference objects and phenomena occurring in Nematik's production line. Establish requirements and continuously coordinate tasks for the project.	
WP 2.a: Deep Learning Classification System for Nematik Production Line Generalization - Research and assess suitable deep learning object detection algorithm for wide-ranging implementation in Nematik's production line - Assemble a large additional dataset containing data from Nematik's production environment	6
WP 2.b: Creation of a Generalized Deep Learning Model - Train and evaluate a new deep learning model using the large additional dataset. - Test the model on both mobile (smartphone or smart glasses) and fixed camera setup with inference PC solutions using actual products from Nematik's production line.	8
WP 3: Object Detection Prototype Implementation for Nematik Production Line - Design and set up interfaces tailored to Nematik's production line. - Evaluate technology and workflow for creating a semantic reference database (identifiers, serial numbers, labels) to digitalize workflow in Nematik's production environment. - Assess object detection methods suitable for Nematik's production line. - Integrate object detection and component classification into workflow support in Nematik's production line.	4
WP 4: Evaluation and Presentation of Nematik Production Line Deployment Results - Develop demonstrators (Recognition, Workflow, and Decision Support) specific to Nematik's production line. - Obtain hardware and software and build the demonstrators for Nematik's production line. - Verify AI-based recognition in Nematik's production environment. - Perform a field study: prototype testing, technical optimization, and system expansion with defined reference parts and cases in Nematik's production line. - Meeting to present proof of concepts and demonstrate the implemented solution based on Nematik's use cases. - Create a report summarizing the deployment in Nematik's production line.	5

Table 5: Future Work Packages

In conclusion, it can be summarized that a prototype has been developed for three platforms, showcasing high customizability. The results obtained from testing on an engine block in our Dropslab laboratory have led to approval. However, it's noteworthy that while the system demonstrates efficiency in the ideal research environment, it may pose challenges in other operational settings as many other variables and factors might affect its efficiency. Nevertheless, our system will be able to overcome these challenges due to its flexible nature. Furthermore, additional enhancements are warranted to address potential limitations and optimize performance.