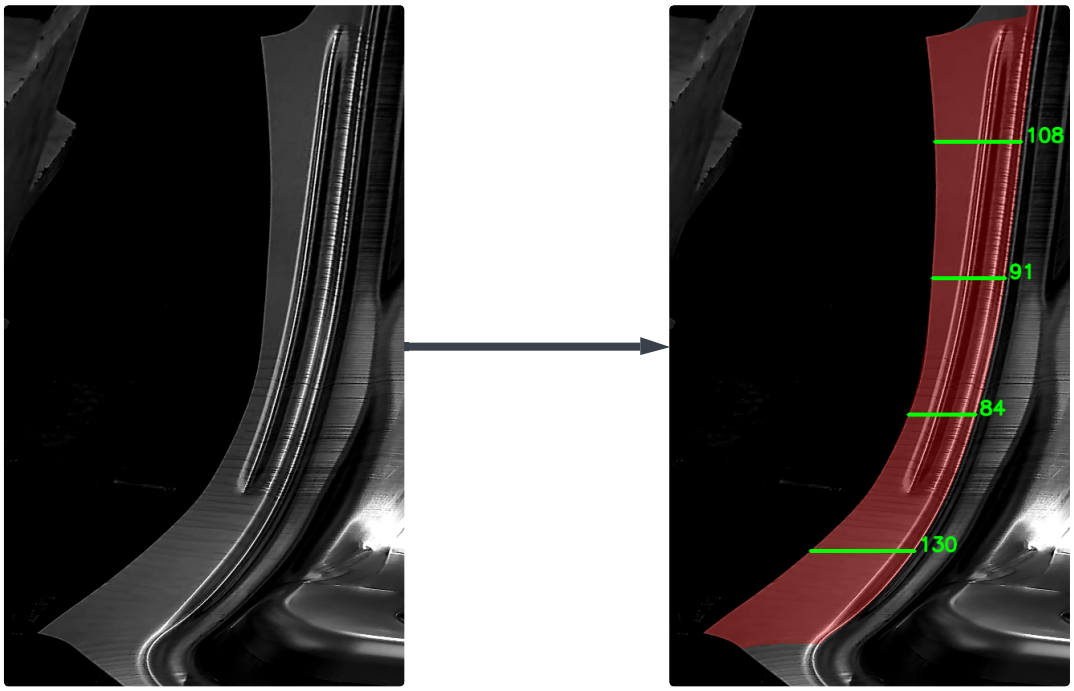
Implementation of Optical Draw-In Measurement System in an Automotive Press Line

- Machine Learning and Computer Vision -



Project Report

VT4

Aalborg University
Materials And Production

Materials And Production
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Implementation of Optical Draw-In Measurement System in an Automotive Press Line

Theme:
VOLVO project

Project Period:
Spring Semester 2024

Project Group:
VT4

Participant(s):
Yaroslav Semenyuk

Supervisor(s):
Benny Endelt
Chen Li

Copies: 1

Page Numbers: 88

Date of Completion:
May 31, 2024

Abstract:

In pursuit of product personalization, climate goals, and safety, Volvo actively invests in research on cutting-edge technologies. Leveraging validated learning and product architecture, a VT4 student from Aalborg University is dedicated to structurally developing and implementing a vision-based machine learning system. This system aims to assist in production quality control and data collection for further research. Concurrently, the student investigates whether the metal sheet draw-in during active production can be estimated using flange measurements. This report details the implementation of U-Net and YOLOv8 models, with U-Net also deployed and tested on the NVIDIA Jetson-Orin Nano edge device. The outcome is a modular product designed to aid in production quality control, although there is limited evidence supporting the capability of estimating draw-in through flange measurements.

Contents

Acknowledgement

Gratitude to my university supervisors: Benny Endelt and Chen Li

Gratitude to Volvo for making this project available and for support from Johan Pilthammar and the team at Olofström.

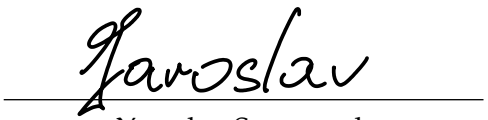
Gratitude to the group of scientists from TATA Steel for help in setup of the camera and data acquisition.

This study was funded by VINNOVA in the EUREKA Smart program (grant number 2021-03144)

Preface

This student project was made available by Volvo, and was divided into the project for fall semester 2023, and master thesis project for spring semester 2024. The student of this project is inspired to utilize the *Lean Startup* methodology by Eric Ries. The focus will be on the Validated Learning approach, to soonest possible travel to Volvo site, and start to gather relevant data to test product hypothesis for higher product viability and waste minimization.

Aalborg University, May 31, 2024



Yaroslav Semenyuk
ysemen19@student.aau.dk

Thesis Summary

Since 1927 Volvo Cars were producing personal cars with the deep interest in environment, safety, and personalization. This led to Volvo Cars being a global car manufacturer by 2024. The pursuit of these three areas is well aligned with the current global climate goals. Being among the global car manufacturers proves that the company has a well-established product portfolio, which may indicate that the company's strategy should shift from radical product innovations, over to incremental process innovation to maintain the competitive success as pointed out by *Patterns of Industrial Innovation*.

This thesis is an indication that the company indeed emphasizes the incremental process innovation, as the student of this project is asked to develop a concept for the product that would bring a production process optimization. Hence, the thesis starts with *Problem Analysis*, where the student throughout physical stay at the company learns about the current production state to reveal possible shortcomings of the existing production setup. In order to achieve highest efficiency, the student utilizes the *Lean Startup* methodology called *Validated Learning*, which provide tools as pivoting meetings to bring light upon which product ideas to preserve and which to pivot from. As the result, the student collects enough data to form a list of requirements for a future viable solution concept. Then, these requirements for a desired product are turned into a product concept through use of the *Product Architecture* by Steven Eppinger, which visually conveys the information about the product's future functional capabilities, geometric shape, but also general prospects and limitations.

The developed concept is then implemented following the general machine learning pipeline, where the student covers data acquisition, data processing, model design, testing and validation, and lastly deploys the model onto an edge-device. With a product demo, the core functionality of a desired product as established in the requirements table, are tested and validated. Lastly the product concept is discussed and concluded. The product concept deemed by Volvo as a valuable beginning for their future research and work in the area of machine learning.

Chapter 1

Introduction

Volvo Cars were producing its cars since 1927 and from the very beginning had the goal to be a brand that cares for people and the world. The main three goals of Volvo is to deliver cars that would provide personalized cars with relevant services. To not only deliver climate neutral cars, but also redesign the production processes to become climate neutral. And lastly make a car safe both for the passengers and for its environment. [6] With these goals in mind, in 2023 the company reached the mark of using 74% climate neutral energy in their operations, where 98% of electricity used is climate neutral. Also Volvo was able to reuse 92% of all waste within all its operations. These and other conscious decisions has led the company to personal records of SEK 399 billion in revenue for 2023 through record retail sales of 708,716 cars. [4]

Since 2018, Volvo has become a global car manufacturer with plants in Europe, Asia, and North America [5], which, along with record growth in retail sales, indicates that the company has one or several established products in its portfolio. This raises a question about whether the company emphasizes incremental process innovation over radical product innovation. Given the increasing production numbers, investing in incremental process innovation, as suggested by *Patterns of Industrial Innovation* [1], becomes crucial for maintaining competitive success. Volvo recognizes this and has directed investments towards research in process innovation, of which this report is a part of.
With the heightened demand for sustainable solutions in the manufacturing industry among which is EU regulations [9], Volvo is exploring smart technologies such as vision systems integrated with Machine Learning (ML). These technologies are particularly relevant in the automotive sector for the manufacturing of body components via the sheet metal forming process. During this process, the measurement of draw-in can be used as a quality assessment measure. This project will evaluate the possibility to use segmentation based ML models for estimation of draw-in.

1.1 Process Method

The group of this project was inspired by The Lean Startup by Eric Ries [26] methodology. Hence the project will strive to follow the principles outlined in the book, especially that are related to the Validated Learning. The core of this methodology is the *Build-Measure-Learn Feedback Loop*, which can be see in Figure 1.1.

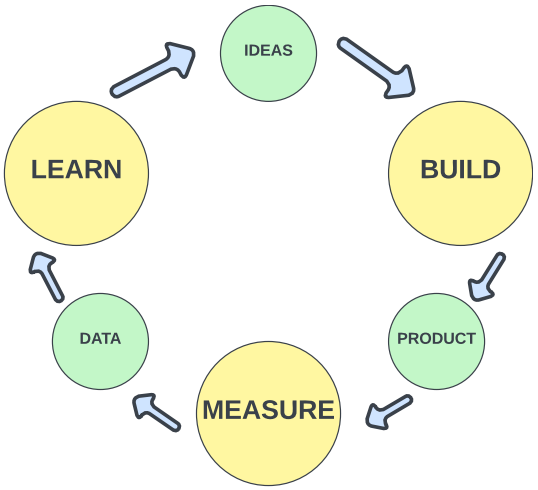


Figure 1.1: Recreation of the Build Measure Learn Loop in [26]

The methodology is designed primarily for startups, but will also be highly beneficial for this project, as it will provide a structure on how to turn ideas into meaningful products for this thesis, where it will serve as a tool to learn whether to pivot or preserve a product or feature idea. This is done through preparation of a *minimum viable product* (MVP), which will be presented to the Volvo representatives, whereafter a feedback will be gained for the student of the project to learn from. Through this, the group will learn, which features of the product are crucial to preserve and which to abandon. This will reduce wasted time on detailed development of unnecessary features, hence increase the likelihood of success by focusing on what truly matters to the company.

Chapter 2

Problem Analysis

This chapter will contain detailed description of the given case, with the underlying topics as the description of draw-in, given press line setup at Volvo site in Olofström, and quality control procedure. Finally the feasibility of the solution will be considered through use of existing products.

2.1 Case Analysis

As the quality standards and requirements for sustainable solutions increase in the automotive industry, a need for utilization of smart technologies arises as stated in Appendix A. Smart technologies as machine learning (ML) are interesting, since they may provide tools for improvement of current production processes, or provide a possibility to track and analyze quality faults that previously were impossible to detect, or increase the speed of existing analysis methods. Considering the given case of manufacturing car body components through the sheet metal forming process, the production rate of up to 15 parts per minute makes it impossible to perform a thorough quality inspection on each produced part. The measurement of the material draw-in during main forming process is said to reflect the general quality of the produced part, and if tracked automatically through a vision system, later it could be used for closed loop control system of a press machine. The draw-in can be seen in Figure 2.1, where the blue area around the rightmost part illustrate the material travel distance that can occur during the main forming process.

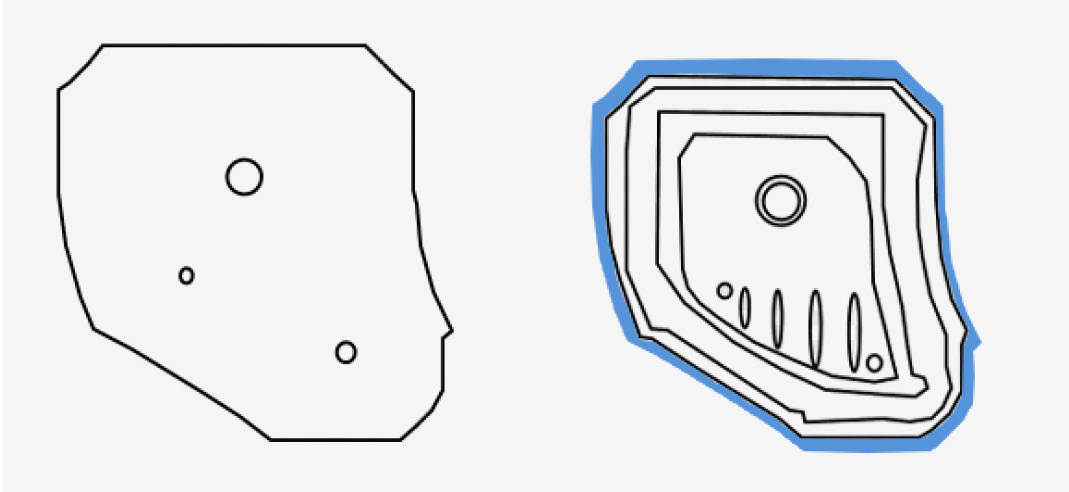


Figure 2.1: Blank metal sheet on the left, and formed car door part on the right.

2.1.1 Forming Process and Quality

The forming process considered in this project report is a process, where a blank sheet of metal is formed into a desired geometry through application of forces using a press machine. Such forming process can be seen in Figure 2.2, where on the left is a blank metal sheet pre-cut to the needed size, while on the right is a car door panel after the forming process. Depending on the complexity of the part geometry, varying number of forming and cutting processes can take place at one press line in order to get a desired car body part.

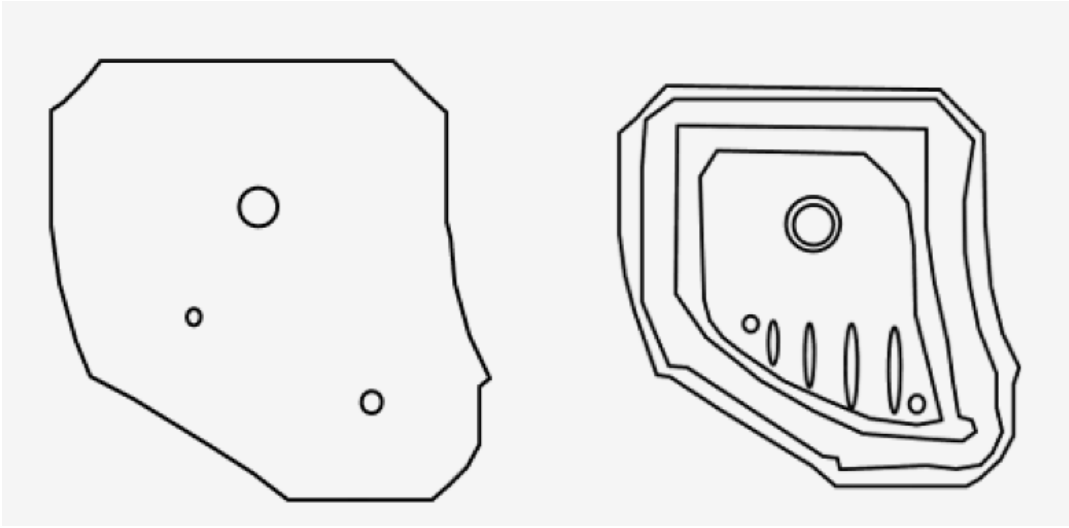


Figure 2.2: Forming process of a blank metal sheet into a car door panel.

Regarding the part quality, there are many known factors that influence the forming process, such as amount of lubrication and machine parameters that among others include blank holder force, cushion force, stamp speed, draw-bead height and shape etc. But also unknown factors such as material purity, metal corn alignment, temperature, surface finish etc. The former and the latter are important to track and be in control of to ensure the proper quality of produced parts. Without control of these factors quality issues will arise, where in best case the issue will be detected in production and part will be removed, while in worst case the flawed part will be detected by the customer as mentioned in Appendix B. There are several types of common quality defects of produced parts that were presented to the group of this project during the physical meeting with quality responsible expert at the Volvo plant. These defects are listed below, where images have been collected during the same meeting:

1. **Stretching** is a quality defect that is also referred to as the *orange peel*, as it visually resembles the peel of an orange. Minor material stretching can be seen in Figure 2.3a, while a severe stretching that affects greater area of the produced part can be seen in Figure 2.3b. The severity and size of this defect affect the difficulty for detection during the production. Alternatively cross section of stretching can be seen in Figure 2.4, which is identified by the fact that the material curves inwards. This type of defect is not only cosmetic defect, but it also affects the part's structural integrity. Therefore, if this defect is detected in production, the risk assessment is made, where it is considered, if this part will be under load in a complete car or not. If such defect occurs in a part that is going to be under load, the part will be scrapped, since it will eventually lead to part failure. Otherwise the severity of a cosmetic defect will be assessed and the part may be kept.

(a) Minor material stretching defect.

(b) Severe material stretching.

Figure 2.3: Two examples of material stretching defect.

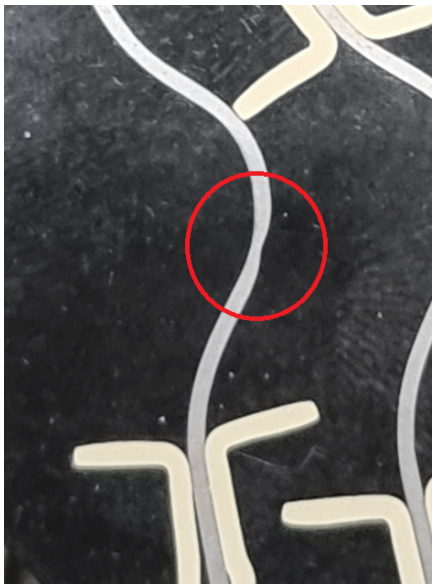


Figure 2.4: Cross section of material stretching defect.

2. **Thinning** is another common defect that affects the part both cosmetically and structurally as in stretching. Upon detection same procedure will be applied as with stretching. This defect can be seen in Figure 2.5, where

contrary to stretching, the material curves inwards from both sides. Such defect affects the structure of the part more than stretching, but is impossible to be differentiated from stretching without cross section analysis.

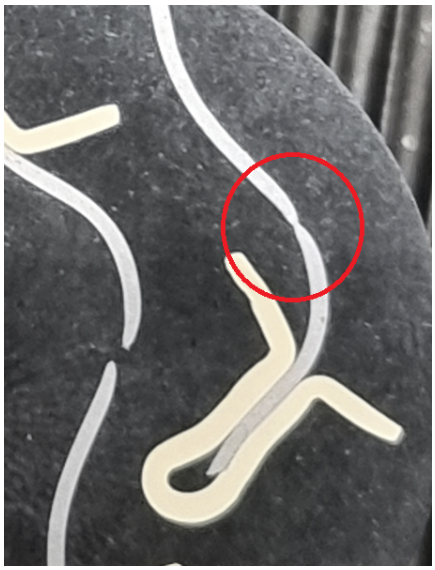
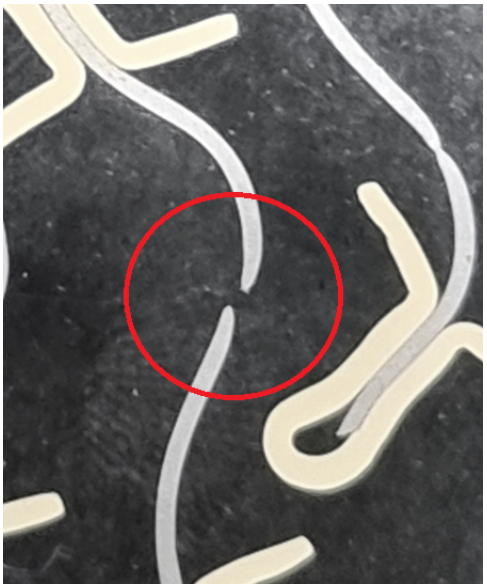


Figure 2.5: A cross section example of thinning defect.

3. **Cracking or Tearing** is a critical defect in production, where the part is always scrapped. Cracks in material can be microscopic, but also be large enough to be detected with an untrained eye as in Figure 2.6a, such defect in cross section can be seen in Figure 2.6b.

(a) A large crack on a part.



(b) Cross section of a crack or a material tear.

Figure 2.6: Example of a crack and a cross-section of another crack.

4. **Wrinkling** is a defect, where due to excess material in certain areas of a part result in uneven surface, where the material buckles or folds on itself.

Summary

According to the quality responsible expert in Appendix B, most of these defects occur due to improper press-machine parameters, and rarely due to the material quality fail. Therefore it is crucial to be able to track and control as many of the factors and parameters as possible, to ensure the quality and to reduce the scrap rate within production. Still the amount of factors that are tracked and fully controlled is limited, where the amount of tracked factors or parameters vary between press lines as it is dependent on the shop floor team experience, motivation and dedication of a given press line.

2.2 Current Setup

Based on the information gathered through contact to the Volvo representatives, but also through the physical stay at production plant in Olofström, given press line T9 will be briefly described alongside current approach of car panels production.

2.2.1 Press-Line T9

The group of this project was presented for several press lines during physical visits in fall semester 2023 and spring semester 2024, but was asked to use press line T9 as the baseline for analysis and further solution development, while keeping in mind possibility for expansion of the solution to the other press lines.

The press-line consist of several production steps of forming and cutting. Since the goal of this project is the draw-in estimation, a potential solution can only be placed at the idle station right after the main forming process. Subsequent steps will cut away some of the material, which will make it impossible to estimate draw-in on. During the visit it was determined that for the solution to be viable, it would need to provide measurements under 4 seconds.

2.2.2 Quality Control at T9

Another crucial part of the current setup is the quality control at the press line. In order to gain needed data for this section, a two-hour physical meeting was arranged with several shop floor workers and managers. Additionally, later another shorter physical meeting was arranged with the worker responsible for quality control education, to validate and cover remaining parts of this topic. Since both meetings covered same topic, it was decided to keep both meeting notes as one summarized transcript, which can be found in Appendix B.

Procedure

According to the meeting with shop floor managers and the individual responsible for quality standards education, there is an educational course that is regularly updated and provided to new workers before they commence work in production, or to established workers upon request. Although according to the shop floor managers, established workers rarely do request courses for update as mentioned in Appendix B. Consequently, a knowledge gap exists between the organization's specified quality control procedures and the practices of individual workers on the shop floor. This results in somewhat different approaches from one press line to another, a fact that was also confirmed during the meeting with the shop floor managers. Therefore, describing the general guidelines at Volvo would not fully capture the current procedural state on the shop floor, nor would it be feasible to address procedures across every press line. Thus, the outcomes of the interview will serve as a baseline for further analysis specifically focused on the T9.

Quality related data that is being collected at T9 can be divided into two parts, that are the press machine parameters and physical assessment of the produced

car door components.

- **Machine Parameters.**
  By the beginning of a new batch of material, the press machine parameters are given to the shop floor workers either by the supplier, or by the production managers. These parameters are guidelines, thus several test stamps are necessary to determine, if the given parameters fit. In case some of the parameters need to be adjusted, the change will be noted physically or verbally within the press line team. For the next batch the parameters may be kept or discarded, but they rarely leave the boundaries of the press line.

- **Physical Assessment** Usually during the aforementioned test stamps, but also during beginning of the batch, mid, and in the end of the batch the production is halted, where press line team will check the produced parts visually for presence of defects. Also, the amount of draw-in is estimated through size measurements of the flanges of the produced parts. The area that is being measured can be seen in Figure 2.7, where the blue area is the flange, and the red arrows illustrate potential area for measurement. According to the shop floor managers, the measured area of the flange changes with respect to which part is produced.

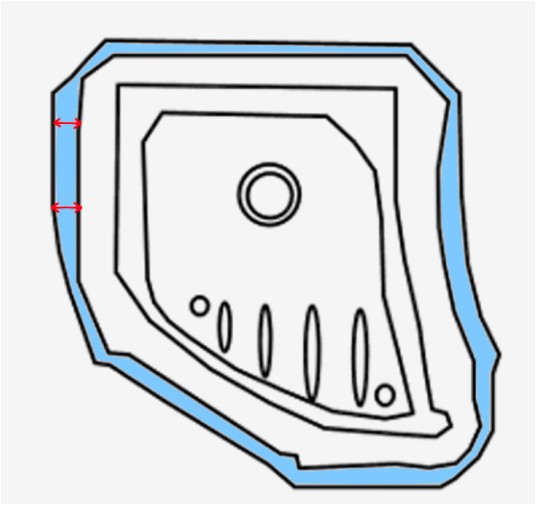


Figure 2.7: Car door part, where blue area is the flange, which is measured.

The above presented data is collected manually, where the exact approach is partially based of the aforementioned course on quality, and partially based of experience of an individual worker. Besides thorough visual inspections, every produced part is briefly assessed visually on its quality, but the production is not

halted, which leaves approximately 4 seconds for the shop floor workers to assess the quality of produced parts. During the test stamps, where the production is halted, flanges are manually measured often using a ruler and a pencil. This flange measurement can take up to 10 minutes for each part in test stamp as mentioned in Appendix B.

What Slips Through

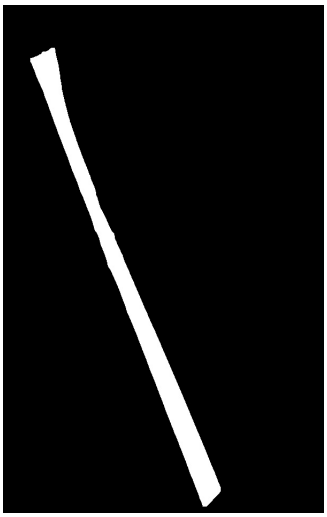
There are some defects that are known to the press team, but it is almost impossible for them to detect these defects during the production. As mentioned earlier, besides test stamps, and random samples, the press team briefly track every produced part, but have severe time limitation due to the high production rate at the press line. Hence, the material stretching and micro cracks are mostly impossible to be detected during active production. While material thinning can only be detected through cross section analysis.

2.3 Thesis Objective

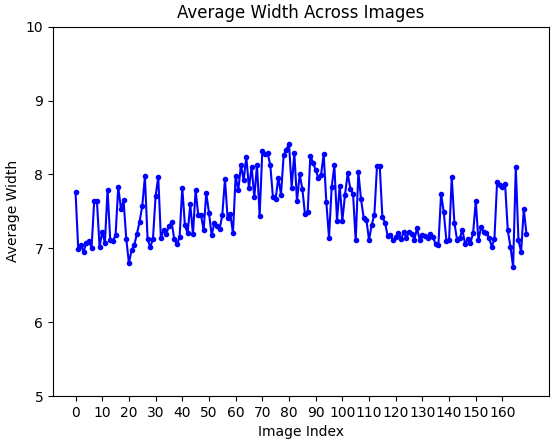
As the known defects are introduced alongside the current setup at the T9 press line, the main objective is as mentioned in section 2.1, to implement and evaluate ML as an automatic tool to assist the press team perform the draw-in estimation as a part of quality control procedure. This will presumably shorten the task time, but also decrease the probability of a defected part slip through, as flange of each produced part will be measured and displayed to the press team immediately after each stamp for evaluation.

2.3.1 Previously Achieved

Feasibility of using ML for dimensional analysis was shown in the fall semester 2023 project at Aalborg University. Where the U-Net model was trained on the image dataset of a production sequence containing car door parts with visible flanges. For each test image, the model was able to detect the area with the flange and accurately segment it, meaning that the model produced a separate binary image exclusively containing the flange, which was then used for dimensioning. The respective results of that project can be seen in Figure 2.8, where in Figure 2.8a is a segmentation mask containing the detected flange, while in Figure 2.8b are the dimensioning results of the whole test set measured in pixels.



(a) Segmentation mask containing flange.



(b) Cross section of a crack or a material tear.

Figure 2.8: Results of fall semester project 2023.

2.3.2 This Thesis

As outlined above during the fall semester 2023, the project used pre-acquired dataset of images, trained 1 model, and omitted the model deployment. Hence the previous scope can be seen in Figure 2.9, while for this thesis, the scope is expanded to cover whole general machine learning pipeline. In this manner, the group of this project will participate in data acquisition, but also deploy the model on an edge-device to test the viability of proposed solution concept. Additionally several relevant ML models will be implemented and tested to determine the base for further development of this concept.

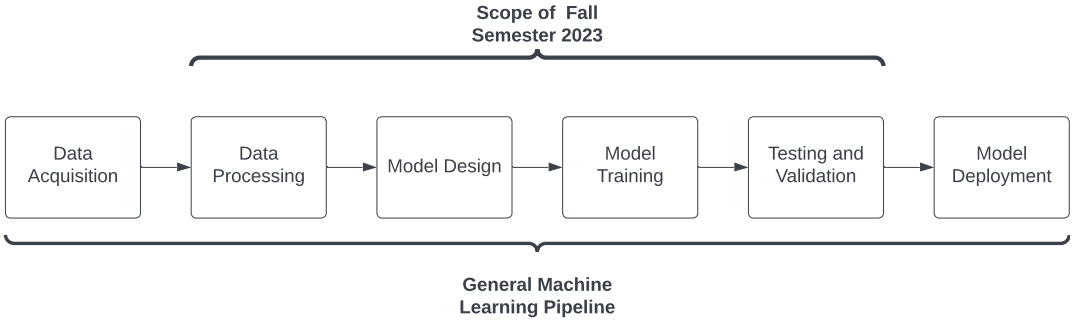


Figure 2.9: Scope of this thesis work.

2.4 Initial Problem Statement

Since the current production state is defined, and the objective is described, initial problem statement can be formulated as follows:

Can a machine-learning-based vision system automate manual methods to measure flanges and estimate draw-in?

2.5 Existing Products

This section will briefly evaluate solution potential to the mentioned initial problem formulation section 2.4. By examining the existing products available on the market, this thesis will contribute to the decision making for the Volvo as which directions for development are possible with external suppliers.

Smart Cameras

There are several suppliers of smart cameras, which are designed to perform automatically many types of visual inspections within a production environment. This type of products is directly related to the task given in this thesis. Compared to standard cameras, the scope of the smart cameras is extended to also perform tasks that normally would require an external processing device. Hence, based on the programmed task, the camera can directly communicate with production machinery, PLCs, and robots. This is achieved by addition of machine learning capable processing components to the camera's structure. [33]

Still, there is no product that will be applicable to all production applications and scenarios. Therefore each of the suppliers has a portfolio of different smart cameras that each is designed for a respective general task. When the customer's task fits one of the available products, the product is mostly ready out of the box, and require minimal efforts for integration within the production environment. Naturally, the amount of pre-programmed tasks can not facilitate all the possible production scenarios. To accommodate this, the suppliers provide a possibility to develop a vision system for the exact task required by the customer. This will require further model tuning and training by the supplier, thus the product development increases alongside its price. [8] [2]

Reflecting upon the topic of this thesis, the described task of predicting the material draw-in based on the flange measurement removes the possibility to apply any of the pre-programmed products by existing suppliers, and would require the external supplier for custom attendance for a personalized product.

2.6 Summary of Problem Analysis

During the group's visit to the production plant site, insights were gained into the production of car door components through metal stamping, along with the various types of production defects and the methods used for their detection and mitigation. It was learned that such tasks require thorough quality control, yet there remains a significant risk of missing some defects. Available products as smart cameras were analyzed to determine their applicability for the task of this project. As the result, any existing pre-programmed product can not directly solve the task of this project, and would require custom product development by the supplier. Therefore, this project aims to evaluate whether a machine learning solution can be developed cheaper compared to that from external suppliers. Thus a machine learning model will be trained using a set of production images collected from cameras mounted on the press line at the Volvo production plant in Olofström. To determine if a ML model can automate the existing task of flange measurement for draw-in estimation on each produced car door part.

Chapter 3

Solution

As the case and objectives are defined in chapter 2 together with stakeholder feedback on the desired product in Appendix C, an ideal product proposal can be made given unlimited time and resources with the respective requirements based on the MoSCoW method [10]. This chapter will present such ideal goal with consideration of realistic limitations of this thesis scope.

3.1 Requirements

Considering the outcome of the problem analysis and the outcome of pivotal meetings as in Appendix C, a list of functional requirements can be formed that will aim to satisfy the needs discovered throughout the analysis and meetings, but also to answer the initial problem statement as in section 2.4. It was decided by the group of this project to set up the requirements using the MoSCoW method due to the student's familiarity with the method. This method will also allow to set up all requirements as one table, for both a solution delimited with this thesis' time frame, but also for an ideal solution given unlimited resources. Separately a list of performance requirements will be formed as the baseline for the viability of the product, which was specified by the Volvo representative.

Firstly, the *Must-Have* requirements (MR) in Table 3.1 outline the most critical function for the solution to posses in order to be considered as a successful solution. These functions are directly the functions that were coveted by the Volvo representatives, where the interest is in detecting and measuring the flanges of produced car door panels, but also to store the measurements for later research purposes. Without these core functions, there is no purpose of further implementation of the solution. Secondly, the *Should-Have* requirements (SR) are important, but are not critical and primarily serve as the validation tool for easier analysis of the solution's performance. Thirdly, the *Could-Have* requirements (CR) are potentially desirable

functions for future iteration of the solution implementation, but currently does not affect the success of this thesis work. Lastly, *Won't Have* requirements (WR) are the functions with lowest priority for the current iteration of the solution, and deemed as not achievable during this thesis' time frame.

Must-Have	
MR1	Capture images.
MR2	Detect flange in a given image.
MR3	Produce segmentation mask of the detected flange.
MR4	Measure the flange' width.
MR5	Plot the width measurement.
MR6	Store the width measurements.
Should-Have	
SR1	Overlay segmentation mask on the input image.
SR2	Measure the time taken to provide the output.
SR3	Store the input images with overlays.
Could-Have	
CR1	Detect flange(s) in a video.
CR2	Produce segmentation mask of the flange in video.
CR3	Overlay segmentation mask on the video.
Won't-Have	
WR1	Measure the flange width in user defined area.
WR2	Detect surface defects.
WR3	Store press-machine parameters with respective output.
WR4	Compensate for increase or decrease of the flange width.
WR5	Display Relevant Production Data.

Table 3.1: Requirements for the solution of given problem.

3.1.1 Solution Design

With given set of requirements, the product design can be established. For structural approach in development of the product design, the method for establishing product architecture by Steven Eppinger [17] was chosen.

Method on establishing the product architecture consist of four steps as follows:

1. Create schematic of the product.
2. Cluster the elements of schematic.
3. Create a rough geometric layout.

4. Identify the fundamental and incidental interactions.

Following this method, **the first step** implies to create a schematic of the product based on its functional elements e.g. the functions that the product should be capable of. As seen in Figure 3.1 the required functions listed in Table 3.1 are used as functional elements for the product design and displayed based on color coding form the requirement table. Several data storing requirements were shorten to one general data storing function to remove repetition. Yielding in 14 functional elements.

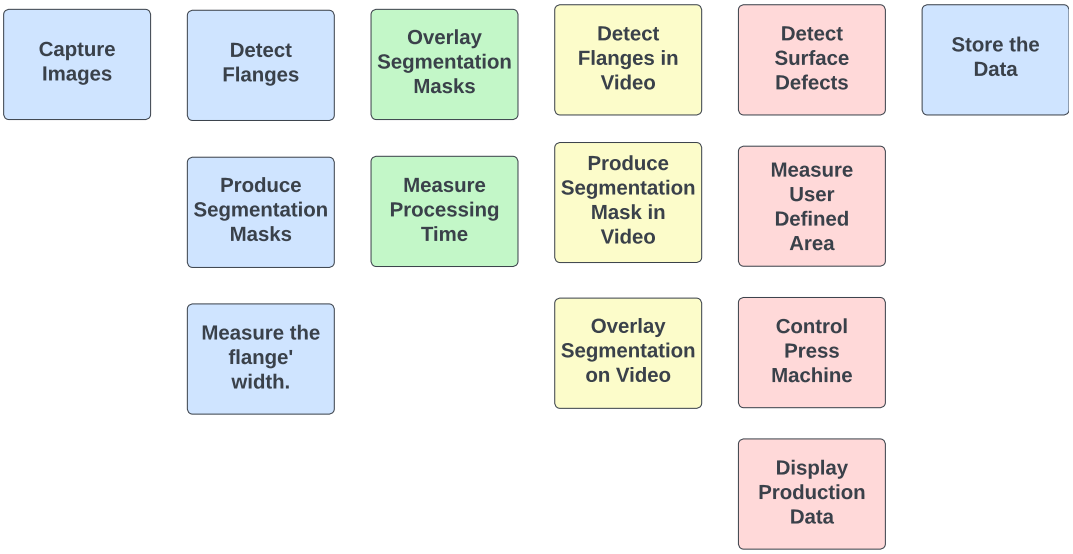


Figure 3.1: Functional elements of the product.

The second step of this method implies to cluster the functional elements of the schematic to chunks (components), which should represent the physical elements that will ensure given functionality. Also there should be a notation of how these chunks are related. The clustered functional elements can be seen in Figure 3.2, where the required functions were divided between four components; a camera, an edge device, a monitor, and a memory storage.

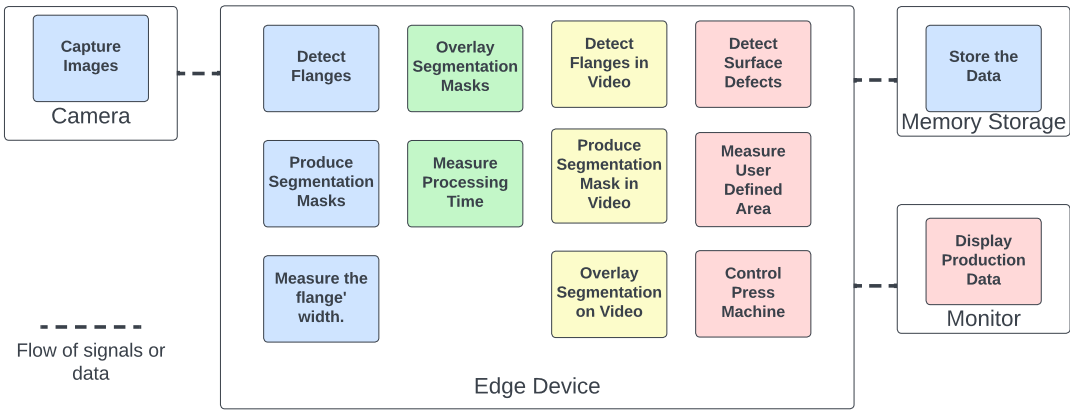


Figure 3.2: Clustered functional elements into physical component.

The third step is to create a rough geometric layout, which will demonstrate a rough dimensional layout of the main components. According to this method explained in the book [17], ideally the creation of geometric layout should be an iterative process in coordination with industrial designers to reveal physical feasibility of the given setup. Thus this activity is simplified to fit the scope of this thesis. The rough geometric layout can be seen in Figure 3.3, where previously identified physical components are now displayed in an approximate setup. From left, the camera is mounted to the press-machine wall observing the press station. Next is the edge device, which processes images from the camera and provides output to the monitor, but also controls the press-machine. Color coding follows the same approach as previously, where connection between the camera, the edge device, and the memory storage is blue, which is a must-have functionality for this thesis work. Contrary the red colored connection represents won't have functionality that is left for future implementation.

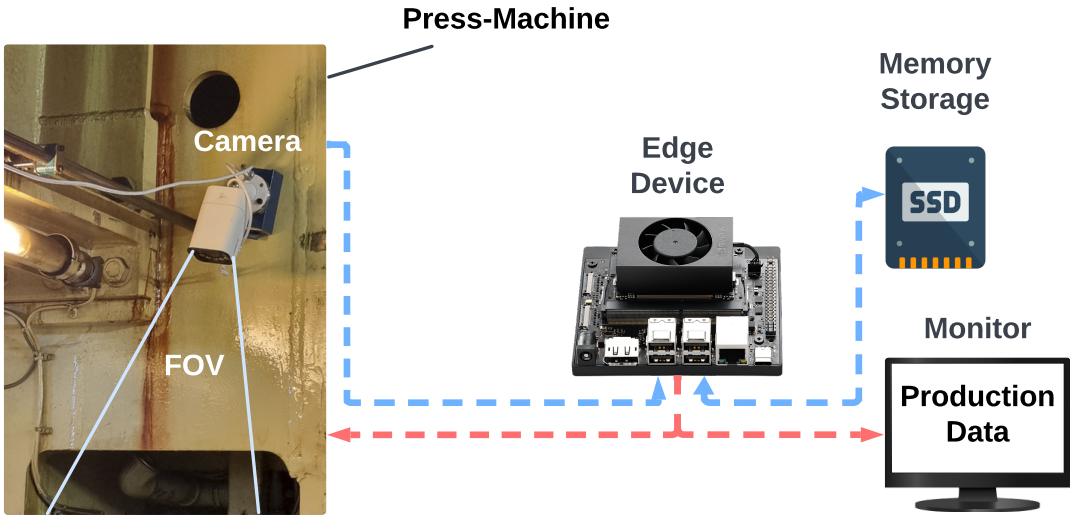


Figure 3.3: Visualization of the rough geometric layout.

Lastly, **the fourth step** is to identify fundamental and incidental interactions. During the physical visit at Volvo plant in the spring semester, the camera was set up to gather production data for this thesis project. As the result, two incidental interactions were discovered. The first being the intense vibrations from the press-machine, which were visible during each machine stroke during production on the recorded video by the camera. The second incidental interaction is induced by the heat from the production process, which creates vapor of oil or water that smudges the walls of the press-machine. The vibration can potentially cause the drift of the camera, or drift of the internal components of the camera. While the vapor will eventually smudge the camera, and some regular maintenance will be required.

3.2 Cost-Benefit Analysis

In this section the proposed solution will be evaluated from the cost-benefit perspective. The associated benefit were discovered throughout the meetings at Volvo plant in Olofström, while the direct cost of the solution will be roughly estimated.

3.2.1 Cost and Scalability

This subsection will contain two estimations of the product cost, the cost of a prototype developed in this thesis work, and an estimation for future implementation to satisfy all the requirements listed in Table 3.1.

Prototype Cost
The direct cost of implementing working prototype to function as a proof of concept is based of student's available equipment, but also through the ordered components at the end of fall semester 2023. The first part of components can be deducted from the solution design as in subsection 3.1.1, such as a camera, an edge device, and a memory storage drive. While the second part is a computer that can perform a ML model training. The approximate cost of the solution of this thesis can be seen in Table 3.2, where price origin can be found in Appendix D.

Direct Cost			
Type	**Name**	**Qty.**	**Price**
Camera	Reolink RLC-810A	1	EUR 105.04
Edge Device	Jetson-Orin-Nano/NX	1	EUR 635.93
Storage	Reolink RLN16-410	1	EUR 419.99
Computer	ASUS ROG Strix G15	1	EUR 1,689.21
		Total:	**EUR 2,850.17**

Table 3.2: Cost of delimited solution.

3.2.2 Machine Learning Scalability

A proof of concept solution for flange measurement of one type of product, can be trained with a higher-end laptop computer designed for gaming as one used in this thesis. Regarding the scalability of a machine learning solution, several stages of the general machine pipeline as in Figure 2.9 should be considered.

Starting with the *data acquisition* and *data processing* the CNN based models require a large data availability for model training to draw conclusions on the model performance. In an established industrial setting, like one in this project, many different products are produced, which has a great potential for large data collection. Contrary, a large data collection is not necessarily a large data availability, as models require consistent data and consistent data acquisition techniques. This may oppose a problem, if geometry of products is changed to fast to be able adequately collect and properly process required amount of data for the model to learn from. [21]

Given required data, the *model training* can be performed, which has other aspects for considerations, when the application of the model is scaled up. When trained on bigger amounts of data and with increase of model parameters, the model training becomes more time consuming as the computational demands rise. [21] This yield an investment in a more powerful hardware for model training, especially the graphical processing unit (GPU) or similar technology that were proven to be crucial hardware elements for model training. [32]. Large amounts

of data, and rapid evolution of product geometry, will require additional consideration of training techniques, where the model would presumably require to be often retrained to facilitate the changes. [21]

Lastly, the increase of the model training and model size, will also increase the computational demand for the *model deployment*. This is particularly applicable on edge-devices with fairly limited processing possibilities. This will in return require extensive research in model optimization, or the hardware upgrade for the model inference. [7].

Addressing these challenges in ML scalability, the next step for further technology validation in given context of this thesis would require in an investment in better hardware. A more powerful hardware would make it possible to conduct a more extensive research on larger amount of data, and further specify the hardware demands for edge-devices. As a guiding proposal for the hardware improvement would be the workstation similar to one at Aalborg University, which was assembled for the purpose of machine learning researching. The cost of such system would be DKK 50,000 or approximately EUR 6,704.75, which can be further inspected in Appendix D. Further research and model development may also arise a need in an additional investment a more powerful edge-device. A candidate for that could be an upgrade from Orin Jetson-Orin-Nano/NX to Jetson AGX Orin, which can be seen in Figure 3.4 that is designed for more advanced machine learning projects and comes at a price around EUR 1,869.34.

Figure 3.4: Jetson AGX Orin [14]

Such edge-device would need to be mounted by the press-line together with camera to observe the production process. Depending on the ML model complexity,

one edge-device could process input from several production stations. Most pessimistic scenario would require one edge-device at each press-line, but still one powerful workstation to function as a server computer to store data and for model training. Such future setup can be seen in Figure 3.5.

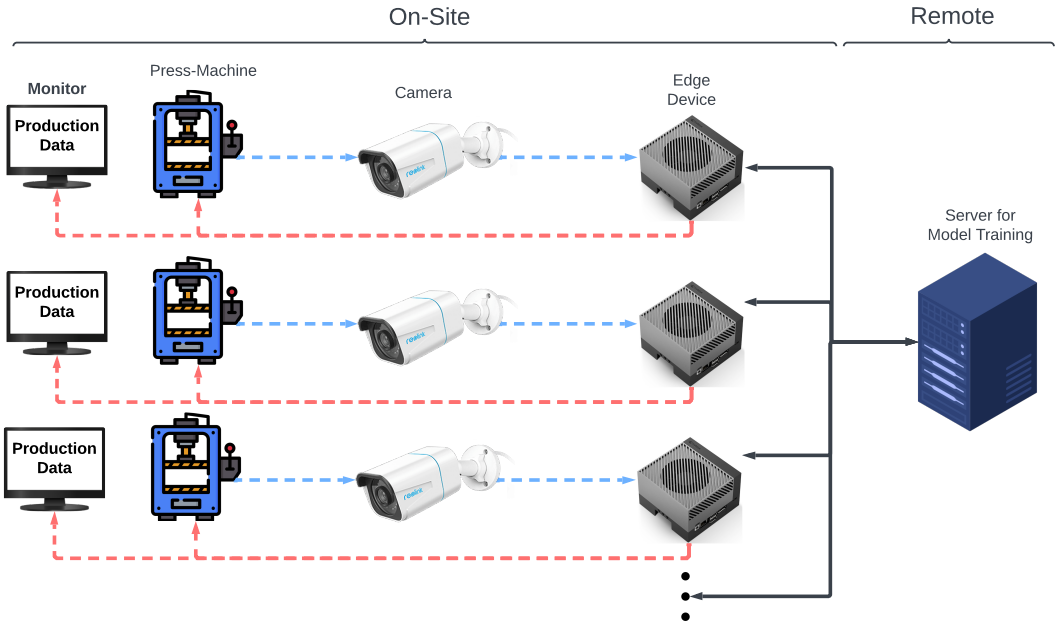


Figure 3.5: Proposed setup for future implementation of the solution.

The predicted cost of future solution implementation can be achieved through two guesses. The optimistic guess, where a cheaper edge-device as Jetson-Orin-Nano is used at each press-line. The pessimistic guess, where the model is complex and a more powerful edge-device as Jetson AGX Orin would be required at each station. Both of the guesses will require the same starting price of a server for model training, and only one camera as Reolink 810A for each station. In addition, there will be adaptive cost depending on the type and capacity of memory storage. Assuming the price of average 1TB of NVMe SSD of EUR 46.71 [18], this can be added to form a general cost equation as in Equation 3.2.2, where n is the amount of press-lines, and x is the amount of TB storage desired.

$$server + n(camera + edge - device) + x \cdot memory = price \tag{3.1}$$

Assuming that there are 10 press-lines at the factory that would have implemented the solution, and that 5TB of storage is enough, the optimistic cost guess can be calculated in Equation 3.2.2, where price of camera originates from Table 3.2 and

server price from the Aalborg University workstation as in Appendix D.

$$6,704.75 + 10(105.04 + 635.93) + 5 \cdot 46.71 = 14,348 \tag{3.2}$$

With the same assumptions for press-lines and memory storage, the pessimistic cost guess could be calculated in Equation 3.2.2

$$6,704.75 + 10(105.04 + 1,869.34) + 5 \cdot 46.71 = 26,682.1 \tag{3.3}$$

This yield the EUR 14,358 to EUR 26,682.1 range of direct component cost for future implementation of given solution, assumed the solution will be implemented on 10 press-lines.

3.2.3 Solution Benefit

As the design and cost of the solution are now established, it is possible to discuss its benefits prior to implementation. In Figure 3.6, a possible drawback of the solution is the implementation time that the company would need to allocate a team of developers to build the proposed solution from scratch, potentially taking longer than employing outsourced teams, or established companies with ready-made products as in section 2.5. However, there is a strong evidence to suggest that the overall benefits of the solution will outweigh the costs and the potential time-related drawbacks of its implementation. The main benefits of the solution can be seen in Figure 3.6, while in depth description follows below.

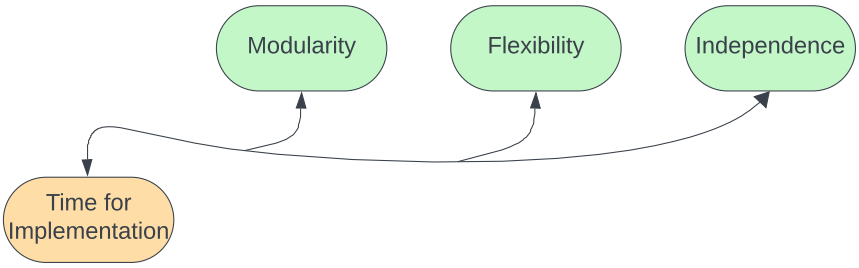


Figure 3.6: The benefit chart of the proposed solution.

Modularity
As a core activity upon establishing the solution design, *product architecture* method was chosen, which purpose is to create the product with highest possible modularity. The benefit of a highly modular product is that the clearly defined blocks of physical components as in Figure 3.2 can be assigned to teams, individuals, or suppliers, which will allow to develop different parts of the solution simultaneously. [17]. The modular architecture also allows to easily change any of the

physical components independently from other components, which allow an independent upgrade of the components as needed without redesigning the whole solution. But also cheaper and faster repair, as a broken component can be easily identified and exchanged for shortest possible up-time. These make the design for manufacturability, design for upgradability, and design for repair.

Flexibility
Comparing the proposed solution to existing products, such as smart cameras, as discussed in Section 2.5, the proposed solution is considered more flexible. As every functional element is intended to be developed in-house, it will facilitate optimal communication with the company, ensuring that the functionality of the solution aligns precisely with the company's true needs. Since the solution is not outsourced, any desired future changes to the design or functionality can be implemented as quickly as possible. This allows for the implementation of new discovered functions without the need to contact an external solution provider and wait for a redesign and a recalculation of costs.

Independence
Lastly, developing the solution in-house enhances independence, allowing for unrestricted experimentation and the addition of any desired functions without external supervision or control. Thus, the course of the future solution development is not dictated by an external supplier.

3.3 Summary of Solution

The requirements for the solution were established throughout the problem-analysis based on the knowledge gained from the close dialogue with the Volvo representatives, and structured with the MoSCoW method, to delimit this thesis work for feasible solution implementation. Then solution has been designed structurally through use of the *product architecture* method [17], which allowed to highlight the direct cost and benefits of the solution. Thus, the solution can now be implemented and tested.

3.4 Final Problem Statement

As the solution is designed, where technical feasibility is outlined, the final problem statement can be formulated as follows:

Can the change in flange width of produced parts be used for draw-in estimation?

Chapter 4

Implementation

This chapter will go through the implementation of the proposed prototype solution of this thesis following the aforementioned general machine learning pipeline as in Figure 4.1, which begins with data acquisition.

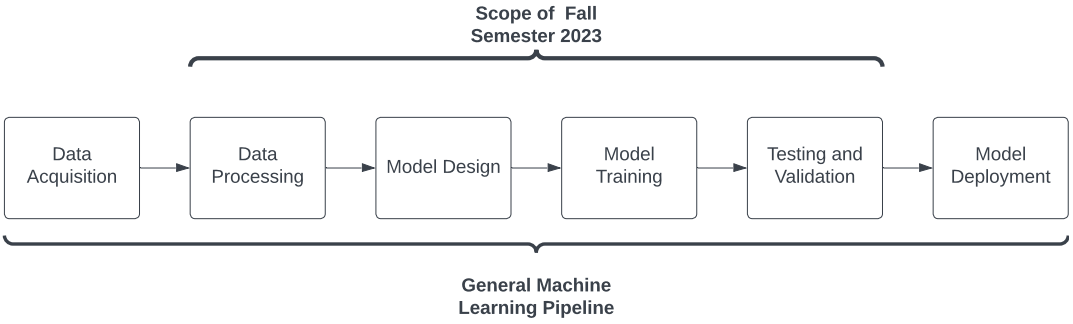


Figure 4.1: Scope of this thesis work.

4.1 Data Acquisition

In order to train a ML model, the set of relevant data is needed. To gather the needed data, the student of this project traveled to Volvo plant in Olofström Sweden. The student together with the team of scientists from TATA Steel mounted the set of four cameras to each of the 4 corner walls of the press-machine, and directed towards the first idle station right after the main forming station. One of the mounted cameras can be seen in Figure 4.2.

Figure 4.2: One of the 4 mounted cameras.

Then the 4 cameras were connected to the storage hub mentioned in Appendix D and set to actively video record the whole test production in a 4K resolution. As the result, approximately 4 hours of the test production were recorded, distributed over 7 separate video recordings with the active production.

4.2 Data Processing

4 Hours of video recording in the state as is, may be hardly used by the model to provide the desired output. For this a proper problem framing should be applied in order to visualize the goal and outline the success criteria. [15] As found during the problem analysis in chapter 2 the goal is to train a machine learning model to detect and segment flanges of the produced car door panels, to later be used for measurement and draw-in estimation. Thus, framing the problem in terms of ML yields the following:

Detect the area of interest and predict its segmentation mask.

Now as the problem is formulated in proper ML terms, the data can be processed to provide exactly the information needed for the model to learn. This will enable it to deliver the desired results later on.

4.2.1 Frame Extraction

Since the task is to learn to predict segmentation mask of the car door flanges during production, all the images containing car door panels should be extracted from the video recordings. The videos were recorded at 25 frames/second, but the production cycle time is between 4 to 5 seconds. Hence, if video converted

to a set of images, there will be many images containing the same door panels. If the ML model is trained on many identical images, the model may get overfitted, which could worsen the model's performance on unseen images during testing [19]. Therefore only frames containing unique produced car door panels should be extracted from the 7 videos.

By observing the production videos, it can be concluded that there are no scene or direction changes in the video, but there is a strong cyclic movement pattern. Therefore, a *Python* script was developed to generate a graph based on the changes in movement within the video. Figure Figure 4.3 presents a plot of the movement in a short 9-second section of the production video. In this plot, peaks represent the moments of most movement, while valleys indicate the least movement. The movements are plotted with respect to the video frames on the x-axis. Observing the plot reveals the cyclic nature of the production, where peaks correspond to product movement and valleys to product idling. Consequently, each peak is followed by a new unique product.

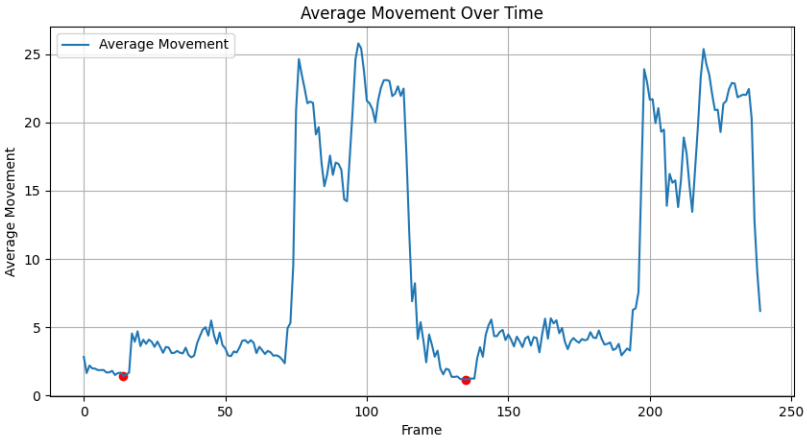


Figure 4.3: Change of movement in the recorded production video.

By knowing that a peak followed by a valley, marked with a red dot seen in Figure 4.3, represent a change to a new product, the frames corresponding to the valleys can be extracted, which will result in a frame with an idling unique car door panel with a flange. Main functionality of this script was achieved through use of the **spicy.signal** library containing function **find_peaks** [11]. During the extraction of the relavant frames, each frame received an ID number for later identification. The developed python scripts as **FrameEx** can be viewed at the Github of the author [30].

4.2.2 Image Transformation and Cropping

The team of scientist provided 4 transformation matrices that would yield an image transformation to a specific area of the flange as can be seen in Figure 4.4.

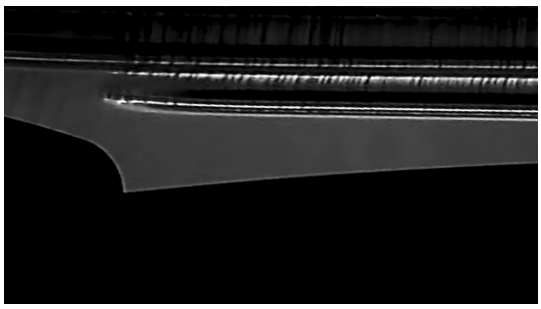


Figure 4.4: Example of one of the flange areas after image transformation.

But the goal of this thesis is detection and segmentation of the whole flange for later analysis. Therefore only values responsible for perspective correction in the transformation were used for image processing, result of which can be seen in Figure 4.5. This transformation makes the image appear as though it was taken from above rather than at an angle, thereby minimizing shape distortion of the flange caused by perspective.

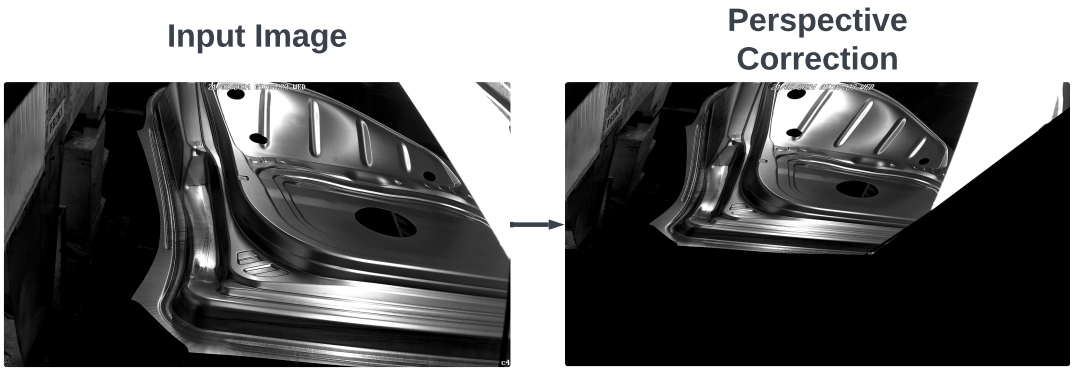


Figure 4.5: Perspective transformation applied to the image.

Due to limited computational resources, the images needed to be reduced in size. This could be done in two ways: cropping each extracted frame to the approximate position of the flange, or scaling the images down without cropping. In the latter method, the entire image of the produced car part would be used, but the fine details of the flanges would be lost. Therefore, the cropping approach was chosen, as it would decrease the computational load during model training while preserving the original detail of the images. Cropping to content was also

straightforward because the position of the flanges did not deviate much relative to the whole frame.
A python script that would take all extracted frames and crop the was developed and can be viewed at the Github repository of this project [31]. Through testing it was discovered that sufficient cropping size was achieved with 850 H by 500 W ratio. This can be seen in Figure 4.6.

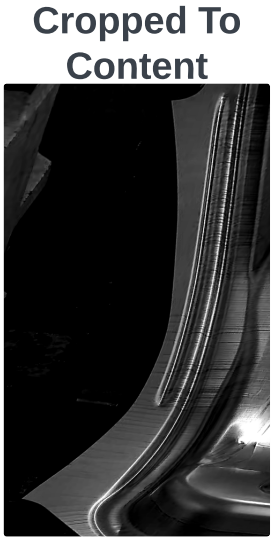


Figure 4.6: Example of image after cropping processing.

4.3 Model Design

As the data required for training acquired and processed, it is now needed to establish a model design for implementation. Instance segmentation by U-Net was proven to be of interest to detect highly reflective and homogeneous automotive parts like during fall semester 2023. As the technical feasibility is established, in this thesis there will be a further evaluation of the previously implemented U-Net model, but also comparison to other state-of-the-art models designed for instance segmentation. Due to limited time of this thesis, there will chosen 3 different models.

4.3.1 Literature Review

As the task for the model is defined by analysis of production images gathered in section 4.1 the search for a suitable model delimits the search to CNN-based models. CNN's are designed for image processing, where they automatically can learn features, patterns, and spatial relations from images [13].

U-net

Originally this model was proven to be efficient in biomedical image segmentation applications, where segmentation accuracy was more important than prediction speed. This model's architecture consist of a encoder-decoder structure. The encoder is for capturing of the image context, while decoder is for precise localization in the image. [28]

Performance

U-net achieved success in biomedical context due to crucial precision in segmentation of tiny cell structures from noisy biological environment. During fall semester 2023 it was discovered that U-Net is a suitable model for the segmentation of automotive parts, with near perfect segmentation of region of interest.

YOLOv8

YOLOv8 or You Only Look Once Version 8, is among the most popular CNN-based models due to its one-stage detector architecture that can predict bounding box and class probability without separate region proposal step. [29] Due to its popularity, there exist almost complete elaborated application programming interface (API), which makes it highly easy to use the model and adapt it for the exact needs of the task. [27]

Performance

YOLOv8 and YOLO framework in general gained popularity due to high inference speeds that could be achieved through this model, which was proven to be applicable in real-time inference for live video feeds in various applications like autonomous vehicles and surveillance. [29].

Mask-RCNN

This model is built on top of the Faster-RCNN framework that was designed for object detection. Mask-RCNN adds a branch to Faster-RCNN that makes it possible to predict masks of the detected objects.

Performance

This model is similar to YOLOv8, where it predicts a bounding box around an object and predicts a mask of the object within the bounding box. Due to similarity, these two models are often compared, where newest versions of YOLO tend to have overall better performance compared to Mask-RCNN as in object segmentation in complex orchard environments. [27]

Summary of Literature Review

Popular models for segmentation were discovered to be YOLO and Mask-RCNN due to their fast inference speeds that make them suitable for real-time applications. Contrary, the slower U-net was proven to suit well the context of this project. Thus, the U-net model should be compared to YOLO and Mask-RCNN, to determine if it is possible to achieve faster inference and still provide near perfect segmentation masks of the metal part flanges for further geometry analysis.

4.4 Model Training

As the architectures of the relevant models are established, the model training can be described in this section. Since there are three models, this section will be divided respectively.

System Specification
Described model training below was conducted on a laptop mentioned in Table 3.2, equipped with a NVIDIA GPU model **RTX 3070 Ti 8GB laptop**, an AMD CPU model **Ryzen 7 6800H 3.20 GHz**, and 16 Gb of CPU RAM.

4.4.1 U-Net Training

The model design and implementation of the training script was based of the tutorial [24], while additional functionality for training validation was added by the student of this project. The implementation given by the tutorial provided necessary functions for model training, also not required, but valuable functions as the *checkpoint* functionality that would allow to pause and continue the model training process for convenience. The other valuable function is the implementation of auto-augmentation of the data through use of *albumentations* open-source library, where images for the model training would automatically be transformed to virtually expand the dataset and reduce over-fitting.

Image Annotation
As the images were prepared, they needed to be annotated accrodingly to the model's requirements. For training, U-net require two set of images, a normal image and a corresponding binary mask for that image. **Computer Vision Annotation Tool** (CVAT) was chosen for this task, ask it could facilitate all the necessary tools for the given task, but also could be deployed locally on personal computer for free. This ensured the privacy of the dataset, compared to other annotation tools as **Roboflow** where the dataset will be public until payed for privacy. CVAT was also a familiar tool to the student of this project from the previous semesters.

The example of the interface can be seen in Figure 4.7, where the flange in the first

image of the set is annotated with a polygon tool. Then this manual process is repeated on the rest of the dataset, where each flange received own polygon selection. The whole annotation process took approximately 12 hours, or 25 seconds for annotation of each individual flange.

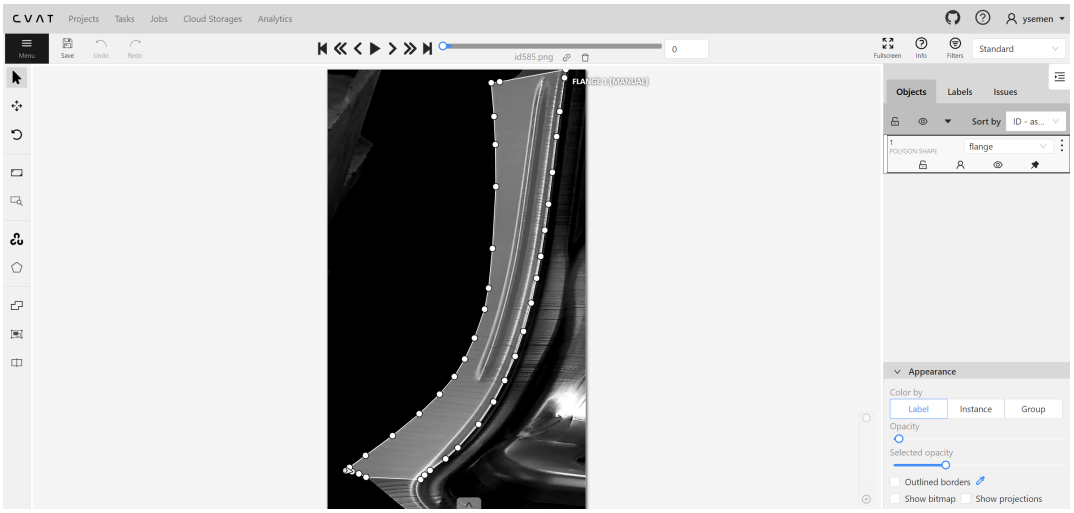


Figure 4.7: Interface of the CVAT with annotation example.

After the annotation process, the dataset could be exported into required format, which resulted in a set of binary images containing only respective masks for each individual flange as seen in Figure 4.8.

Figure 4.8: Snippet of the folder with binary masks.

For better model training validation, and later model performance testing, a sta-

tistical method as **k-fold cross-validation** [25] was chosen. In the given case of training CNN-based models, this method will prolong heavily the total training and testing time. Therefore smaller number "3" was chosen as "k" for number of fold for this method. This yield the total data set being shuffled into 3 different combinations of a testing, training, and validation sets. A python script was developed for this purpose and can be found on Github repository of this project [31]. Through k-fold cross-validation model performance is virtually trained and tested on the whole dataset, providing better representation of model's performance.

Training Parameters
For repeatability of the training process, a summary of training parameters can be seen in Table 4.1, which is similar to the parameters during fall semester 2023 project, but with prolonged training and different image size.

Parameter	**Value/Description**
Learning Rate	1e-4
Batch Size	2
Number of Epochs	30 for each fold
Number of Workers	2
Image Height	850
Image Width	500
Device	CUDA (GPU) RTX 3070 Ti laptop
Input Channels	3
Output Channels	1
Data Augmentation Techniques	Rotation, Horizontal Flip, Vertical Flip
Normalization	Mean, Std, Max Pixel Value=255.0
Loss Function	BCEWithLogitsLoss
Optimizer	Adam
Activation Function	Sigmoid

Table 4.1: Hyperparameters of the U-Net training.

U-Net Training Results
As an additional training activity, it was chosen to perform the model training on 3 aforementioned folds without auto-augmentation, and 3 trainings on same 3 folds but with auto-augmentation. This would asses the image augmentation effect on training process.

The chosen metrics for training validation were dice score, loss, precision, and recall. In this section only results of first fold without augmentation, and results of first fold with augmentation will be presented, while results for all folds can be found in Appendix E.

Training and Validation Loss

Loss curves mathematically represent the learning process of a given model over time. Training loss represent the error that the model have, and indicates represent how well the model learns to fit to the training data. Respectively validation loss indicates how well the model learns to fit to new data. If the global trend of both graphs show the decrease of loss, the model's performance becomes better. [3].

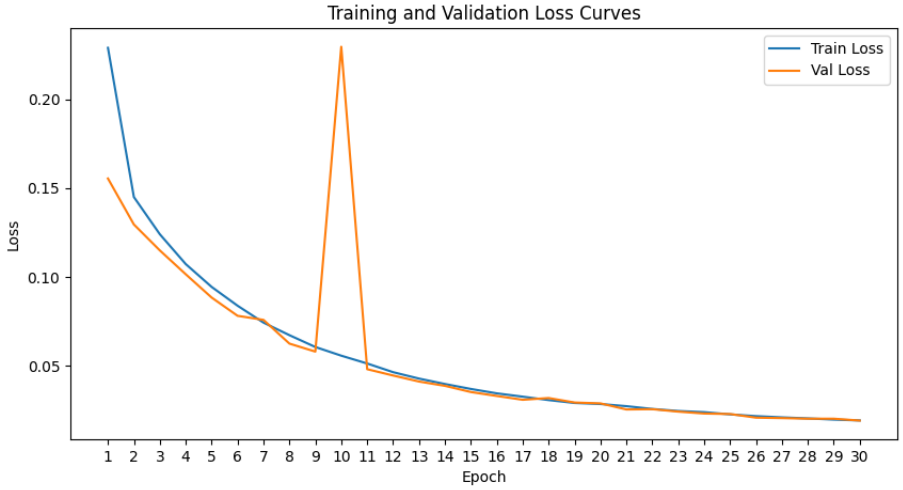


Figure 4.9: Loss curves without augmentation.

The plot in Figure 4.9 show the loss decrease without use of image auto-augmentation over the course of 30 epochs of training, where the blue graph is training loss and orange graph is validation loss. There is a spike on 10th epoch in validation loss, which may indicate that the model struggles to generalize on new data, but overall continues to converge. On the other hand plot in Figure 4.10 show training on the same fold of data, but with image auto-augmentation. With augmentation the model seems to learn better, as both graphs converge faster, but also because the validation spike is removed. In Figure 4.10 can be seen, that after the 10th epoch of training, both graphs seem flat, which idicates that the model does not learn much between 10th and 30th epoch. Thus prolonged training may provide insignificant model improvement compared to additional time spent on training.

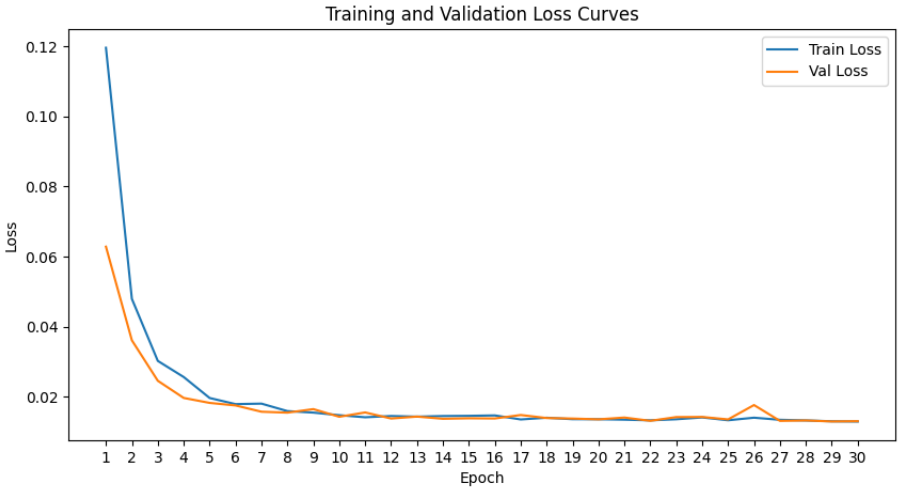


Figure 4.10: Loss curves with augmentation.

Precision

This metric represent the success of prediction by determining the false-positive rate, where high precision is low false-positive rate. Vice versa low precision indicate high false-positive rate meaning that background pixels were labeled by the model as instance class (flange). The precision P can be calculated as follows in Equation 4.4.1, where T_p is number of true positives, and F_p is the number of false positives. [16]

$$P = \frac{T_p}{T_p + F_p} \tag{4.1}$$

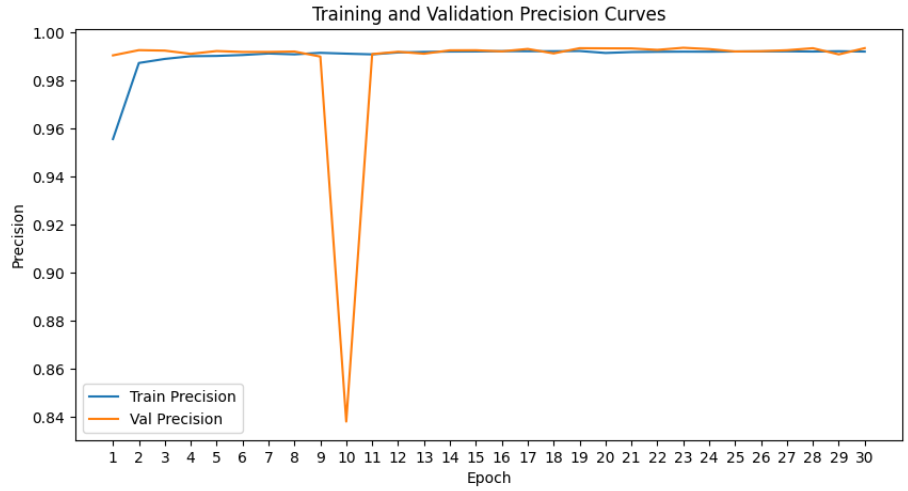


Figure 4.11: Precision curves without augmentation.

The plot in Figure 4.11 show high precision on non augmented data, but with a validation spike again on 10th epoch, while overall the precision maintain high. Contrary the plot in Figure 4.12 show as high precision at around 99%, but also managed to remove validation spike at 10th epoch, indicating theoretically more robust model performance.

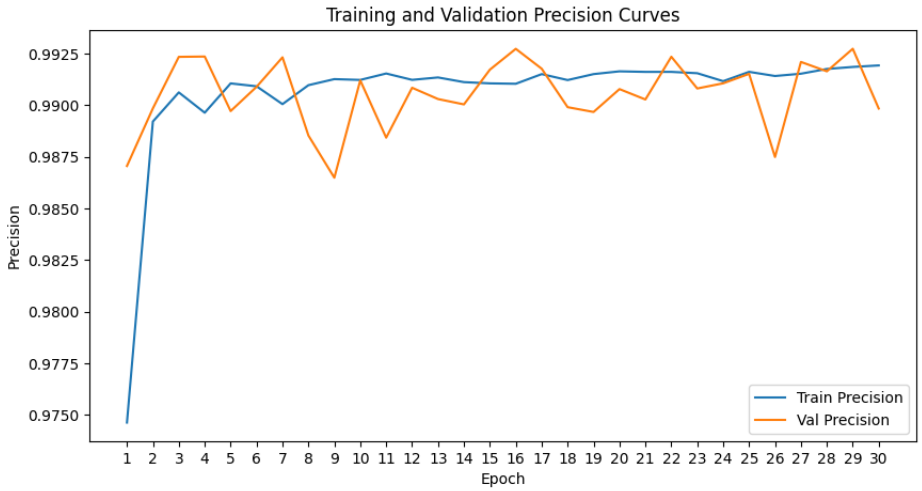


Figure 4.12: Precision curves with augmentation.

Recall

This metric is similar to precision, but instead of false-positives it represents the

false-negative rate. High recall represents low false-negative rate, meaning how many pixels of the instance class (flange) were falsely labeled as background class. The recall R is given in Equation 4.4.1, where T_p is number of true positives, and F_n is a number of false negatives.

$$R = \frac{T_p}{T_p + F_n} \tag{4.2}$$

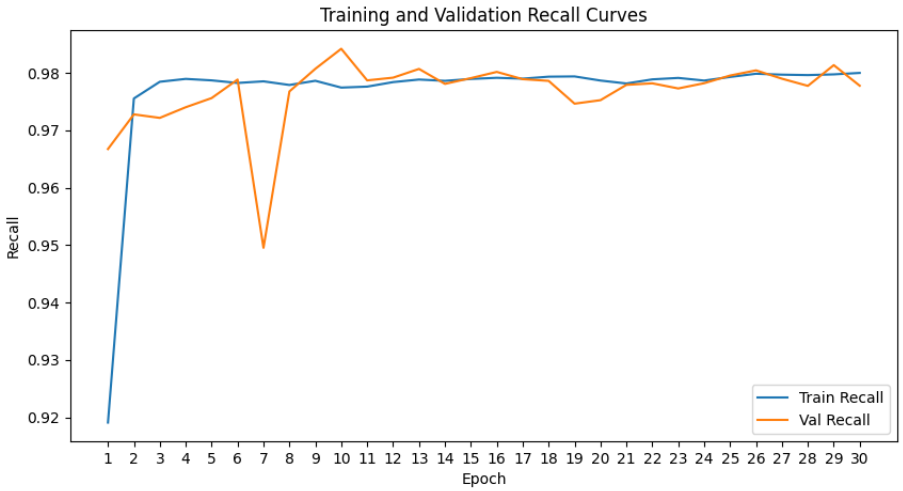


Figure 4.13: Recall curves without augmentation.

The plot in Figure 4.13 show high recall on non augmented data, but with a validation spike on 7th epoch. Again, the augmented data as in Figure 4.14 removes the spike.

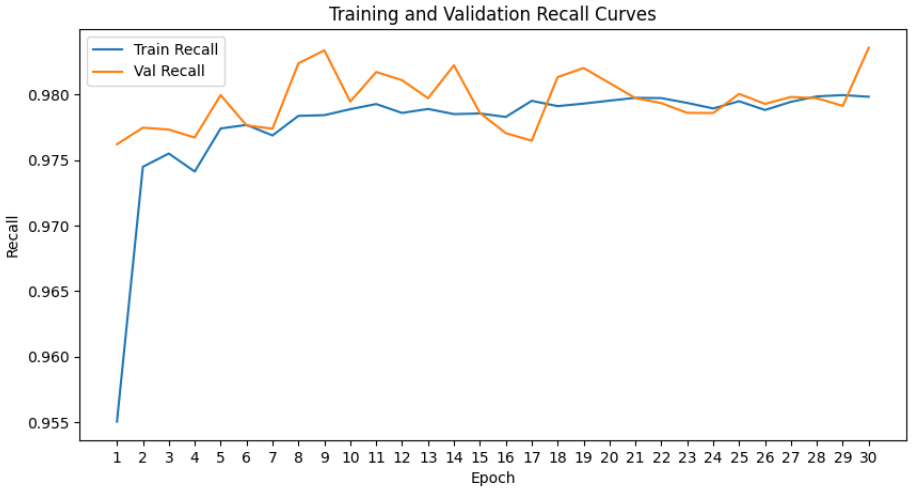


Figure 4.14: Recall curves with augmentation.

Intersection Over Union (IoU)

Jaccard index, or IoU is the metric that describe how well does the predicted mask by the model overlap the ground truth masks from manual labeling of the data. It is given in Equation 4.4.1, where the numerator part is the area of overlap between A and B, and the denominator is the area of union between A and B. [34]

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{4.3}$$

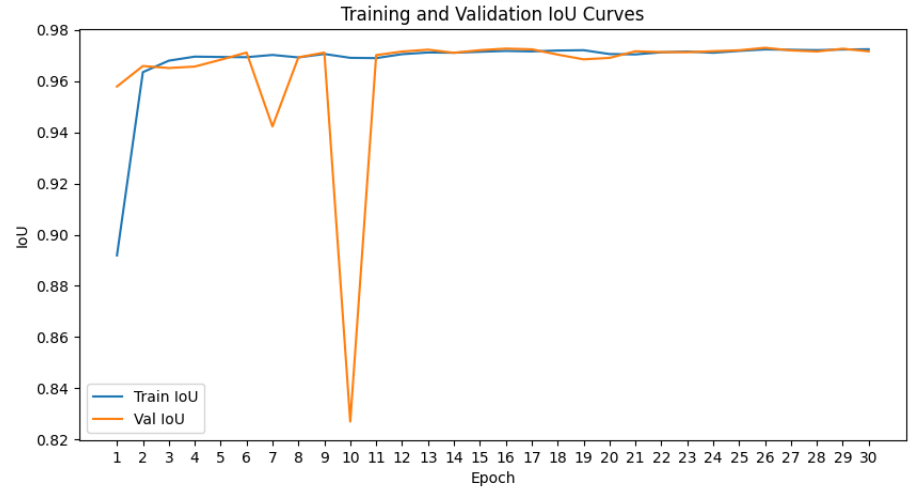


Figure 4.15: IoU curves without augmentation.

The plot in Figure 4.15 show the IoU for non augmented data, with the same spike on 10th epoch like previously, with 96% - 97% overlap between ground truth masks and the predicted masks. The plot in Figure 4.16 shows IoU for augmented data, where the overall overlap was improved due to data augmentation.

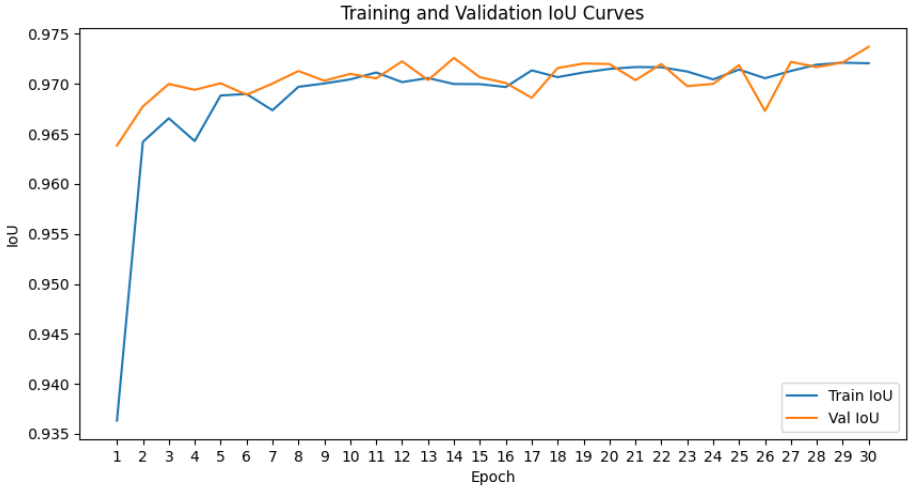


Figure 4.16: IoU curves with augmentation.

Summary of U-net Training The results of the training process were presented for first fold of the data, and compared to the a respective training on the same fold of data, but with auto-augmentation. The plots as above, can be found for all other folds in Appendix E.

4.4.2 YOLOv8 Training

The model design, implementation of the training script, and inference script was based of the tutorial [12]. Differently from the U-net implementation, YOLOv8 has a complete API, which allowed with minimal efforts to adapt and train the model on given dataset of this project.

Image Annotation
Same dataset of previously annotated images in CVAT could be used for this model as well, but the binary masks needed to be converted to set of polygon coordinates within a text file. The same tutorial for implementation of YOLOv8 model also provided the *masks_to_polygons.py* script for this purpose. [12]

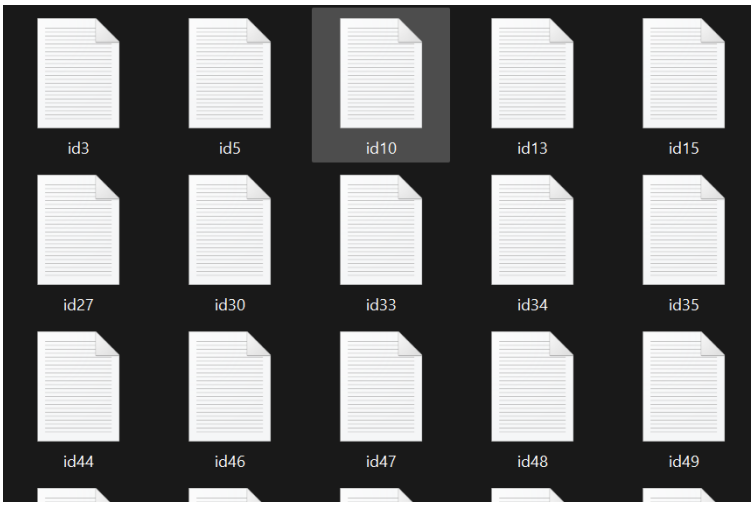


Figure 4.17: Snippet of the folder with text files with coordinates.

Subsequently the dataset needed to be divided into three folds as previously for 3-fold cross-validation. Since the structure of the dataset for YOLOv8 model is the same e.g. each image has a respective text file, the same script for fold division could be used for this data set as well.

Training Parameters

For repeatability of the training process, a summary of training parameters can be seen in Table 4.2, which were kept similar to the one used for training of the U-net model for easier comparison.

Parameter	Value/Description
Learning Rate	1e-4
Batch Size	2
Number of Epochs	30 for each fold
Number of Workers	2
Image Height	850
Image Width	500
Device	CUDA (GPU) RTX 3070 Ti laptop
Input Channels	3
Output Channels	1
Data Augmentation Techniques	Same Albumentations library
Normalization	Mean, Std
Loss Function	BCEWithLogitsLoss + more
Optimizer	Adam
Activation Function	Sigmoid Linear Unit

Table 4.2: Hyperparameters of the YOLOv8 training.

YOLOv8 Training Results

As mentioned above, due to YOLO's popularity, the model's script is elaborated for easier adaptation by the user to personal needs. Therefore, the model after complete training provides standard validation metrics without the need to explicitly code them.

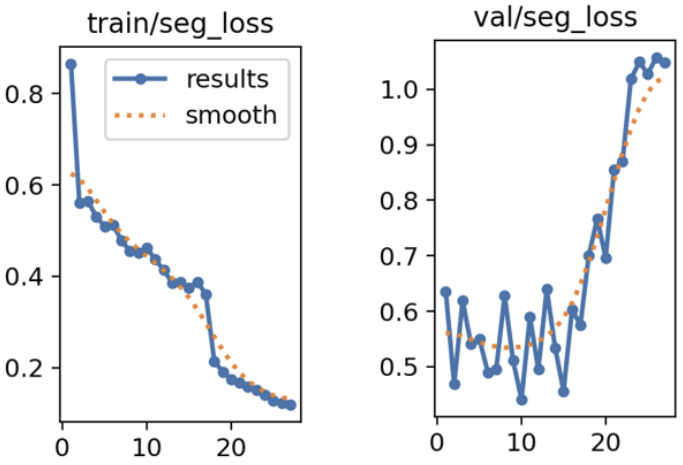


Figure 4.18: Segmentation loss curves of 30 epoch training of YOLOv8.

The plot in Figure 4.18 show the training and validation loss for the segmentation masks, where the training on the left seem as a normal training process with convergence. The validation loss on the right plot show rapid increase, which

may indicate the model's tendency for over-fitting, or generally bad segmentation performance on new images.

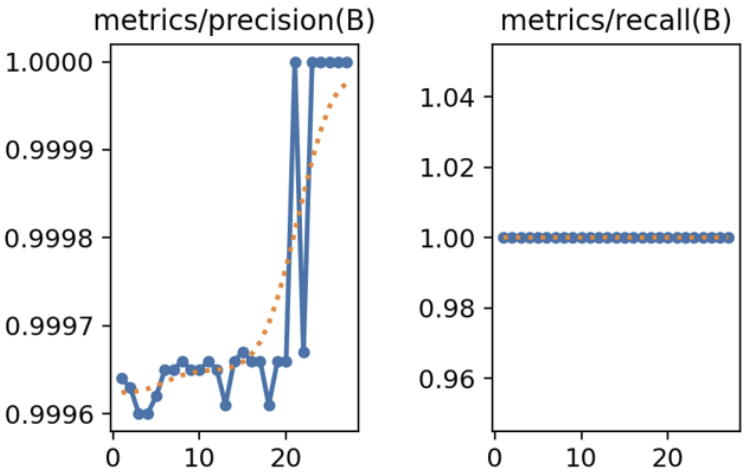


Figure 4.19: Segmentation precision and recall of 30 epoch training of YOLOv8.

Contrary to loss curves, the plots in Figure 4.19 for precision and recall show model's perfect performance regrading true positives and false negatives. The model also provided images showcasing models segmentation performance during training which can be seen in Figure 4.20. The model oversegments the flange accepting considerable amount of background pixels as the object class, which contradicts the precision and recall reseults.

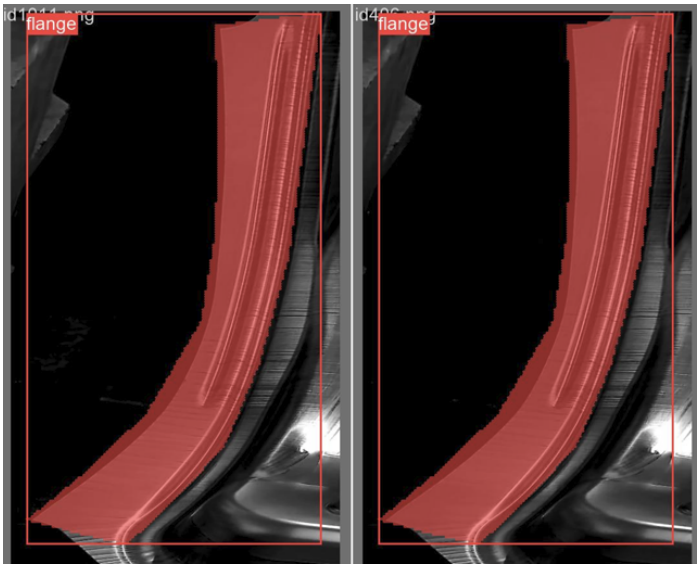


Figure 4.20: Segmentation performance of 30 epoch training of YOLOv8.

Summary of YOLOv8 Training
The results of the YOLOv8 training process were presented for the first fold of data, which were of questionable performance. The cross-validation method helped to reveal that this over-segmentation problem persist on every fold of data, rest of which can be seen in appendix Appendix E.

4.4.3 Summary of Model Training

The results of training of the U-net model supported the conclusion of the fall semester 2023, where the U-net was deemed as suitable model for the detection and segmentation of flanges in production. Contrary, YOLOv8 was tested in a similar manner, which raised a question on the model's applicability for given scenario. Due to Mask-RCNN's similar theoretical functionality, it was chosen not to implement Mask-RCNN, as it could lead to similar unsatisfactory results as with YOLOv8.

4.5 Testing and Validation

Following the machine learning pipeline, the models are now trained and can be tested using the requirements established in Table 3.1. This section will be divided accordingly to the requirement table, with must-have and should-have requirements and respective tests.

Must-Have

- MR1: Capture images.
- MR2: Detect flange in a given image.
- MR3: Produce segmentation mask of the detected flange.
- MR4: Measure the flange' width.
- MR5: Plot the width measurement.
- MR6: Store the width measurements.

Test of MR1:
Due to images of the production were extracted from a pre-recorded video, and then loaded into the model, technically the product does not perform the image capturing. Still, the implementation of this will not be very different, as in current state, the product reads from a directory containing images. While in proper scenario, the product would read from an active camera video stream, or take a picture of the production upon external signal from the press-machine PLC. Hence,

the product's capability to load images from a directory is considered the same as reading from a camera feed. This is applicable for both implemented models.

Test of MR2 and MR3:

This functionality was tested by providing the models with images from the given test set with flanges. In Figure 4.21 can be seen, that the model's direct output is the segmentation mask, where the flange has been detected. Contrary, YOLOv8 outputs the mask overplayed on the same test image, where can be seen that the model oversegments the detected flange. As it was shown the poor training performance, but also poor testing performance, the YOLOv8 will not be tested further. The rest of the predictions can be seen in Appendix E.

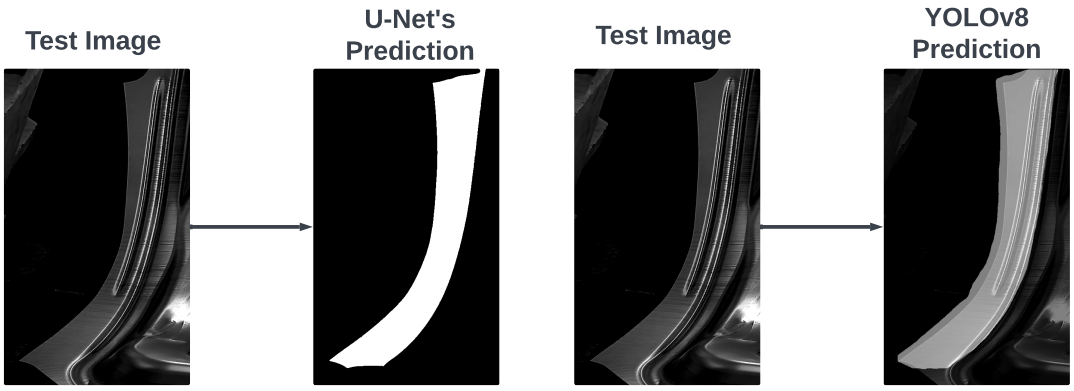


Figure 4.21: Prediction test of U-Net on the left, and YOLOv8 on the right.

Test of MR4:

This functionality was tested by providing the model with images from the given test set with flanges. While the script is running, the width measurements are outputted to the terminal, snippet of which can be seen in Figure 4.22.

```
Widths for id5.png: [109, 92, 84, 131]
Widths for id12.png: [107, 90, 83, 129]
Widths for id13.png: [108, 91, 84, 132]
Widths for id22.png: [108, 91, 83, 129]
Widths for id24.png: [107, 90, 82, 131]
Widths for id32.png: [108, 91, 83, 131]
Widths for id33.png: [107, 91, 83, 131]
```

Figure 4.22: Snippet of the terminal output displaying the width measurements.

Test of MR5:

As in the previous test, this functionality was tested by providing the model images to perform predictions on. With the help of the *matplotlib* and *PIL* open-source libraries the measurements from **MR3** are automatically plotted, which can be

seen in Figure 4.23. The y-axis represent the measured flange width specified in number of pixels, while x-axis is the id of the tested image for traceability. There are four graphs, since it was decided during pivoting meetings as in Appendix C that instead of one average width of the whole flange, the product of this semester should output measurements for four areas. Due to no further specifications, these areas were chosen to be placed as every 20th percent of the total flange width starting from the top of the flange, and named respectively.

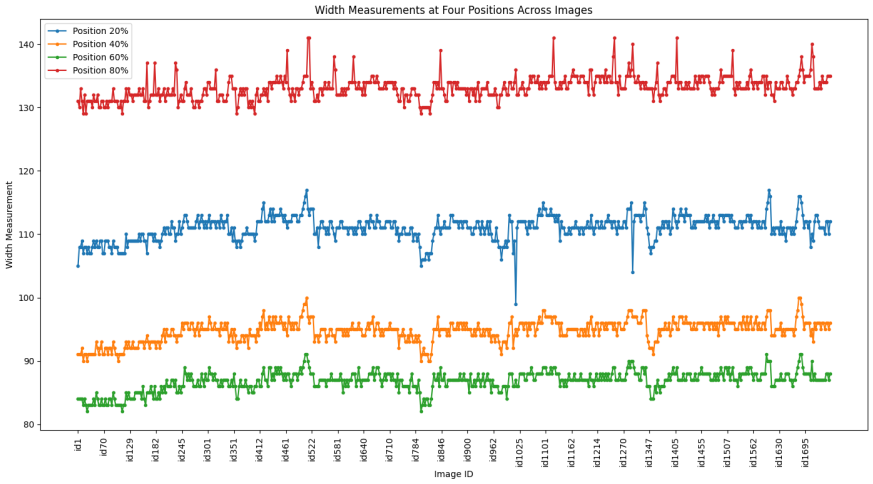


Figure 4.23: Plot of measurements.

Test of MR6:
The last must-have function is to store the measurements, which can be seen in Figure 4.24, where a snippet of the generated text file can be seen with stored width measurements as arrays containing every 20th percent of the flange length.

```
id1: [105, 91, 84, 131]
id2: [108, 91, 84, 130]
id4: [108, 91, 84, 133]
id5: [109, 92, 84, 131]
id12: [107, 90, 83, 129]
id13: [108, 91, 84, 132]
id22: [108, 91, 83, 129]
id24: [107, 90, 82, 131]
```

Figure 4.24: Snippet of the text file with stored measurements.

Should-Have

- SR1: Overlay segmentation mask on the input image.
- SR2: Measure the time taken to provide the output.
- SR3: Store the input images with overlays.

Test of SR1:
Upon starting the script as can be seen in Figure 4.25 the product overlays the prediction from the model on top of the provided input image. Additionally, as determined in pivoting meetings as in Appendix C the overlay should contain the measurements as well.

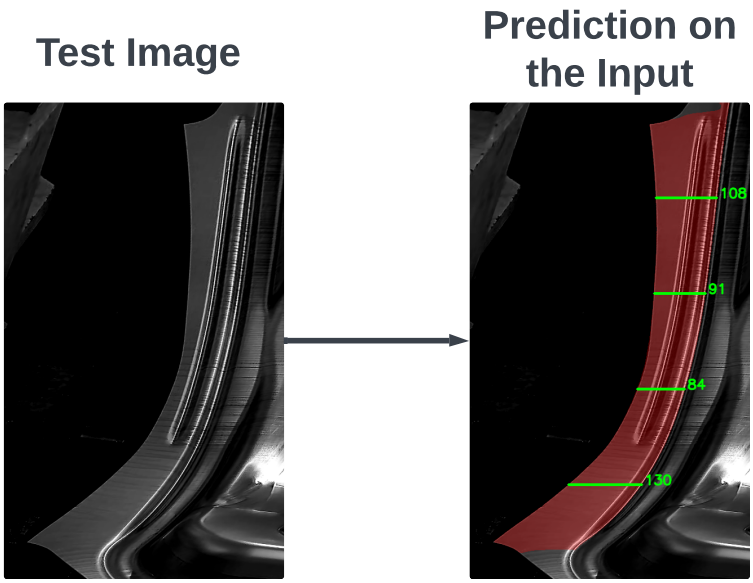


Figure 4.25: Snippet of the text file with stored measurements.

Test of SR2:
After the product processed every provided test image, it outputs to the terminal the total execution time as seen in Figure 4.26. The displayed time of 55.88 seconds is the total execution time for the model to perform predictions together with other processing for the desired output. Since the tested folder contained 580 test images, the average execution time for each test image is $0,096$ seconds on the presented computer as in Table 3.2.

```
Widths for id1738.png: [110, 95, 87, 135]
Widths for id1740.png: [112, 96, 88, 135]
Total time taken for predictions: 55.88 seconds

Process finished with exit code 0
```

Figure 4.26: Snippet of the terminal output with total execution time.

Test of SR3:
The last desired functionality for this product was tested the same as previous tests, where after the complete execution, respective code directories were checked for the output. As seen in Figure 4.27, the snippet of the directory show that the tested images were stored with applied overlays.

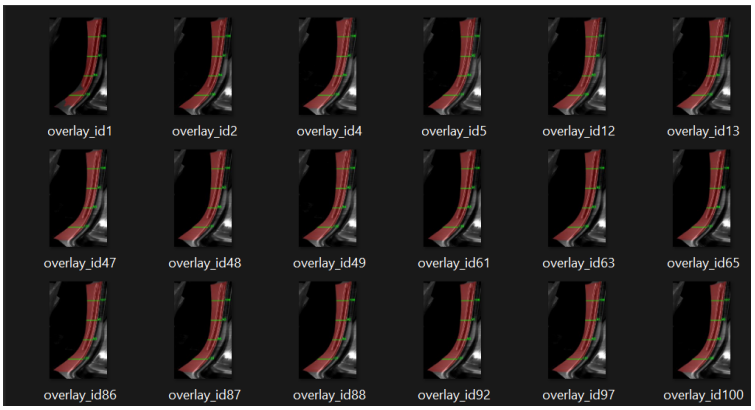


Figure 4.27: Snippet of the terminal output with total execution time.

Validation of the Results
As all vital functions of the product were tested, the results can now be validated to determine the success rate of the solution. The *Must-Have* and *Should-Have* requirements have been tested yielding 9 tests in total.

All of the 9 functions are deemed as successful in Table 4.3 since upon execution of the solution script all vital functions specified by the requirements are performed.

Capture images	+	MR1
Detect flange in a given image	+	MR2
Produce segmentation mask of the detected flange	+	MR3
Measure the flange' width	+	MR4
Plot the width measurement	+	MR5
Store the width measurements	+	MR6
Overlay segmentation mask on the input image	+	SR1
Measure the time taken to provide the output	+	SR2
Store the input images with overlays	+	SR3

Table 4.3: Validation table.

4.5.1 Summary of Testing and Validation

The core functionality of the solution is now tested and ensured, and the report can therefore move to the final step of the general machine learning pipeline, which is the deployment of the model.

4.6 Model Deployment

As outlined in chapter 3, the ML model needs to be deployed on an edge-device and mounted at the machine-press. During fall semester 2023 the lower-end edge-device Nvidia Jetson-Orin Nano was ordered for future deployment and testing. This section will describe necessary steps passed this semester for model deployment.

Prerequisites
In order to prepare the edge-device for work, the user has to load the necessary software onto the edge-device. To do so, the user has to have a Linux Ubuntu 22.04 or 20.04 [23]. For this, a virtual machine running Ubuntu 22.04 was created through **VMware Workstation Pro**. Subsequent installation of the required software and flashing of the edge-device followed the Nvidia's documentation [23] and a video tutorial by JetsonHacks [20].

Model Testing
After the edge-device has been flashed and necessary software installed, the solution directory was transferred through GitHub. The inference script was executed on the edge-device as seen in Figure 4.28, where the total time for prediction on 580 images has taken 376.13 seconds in total, or 1.53 seconds on average per image, which is within the production cycle-time.

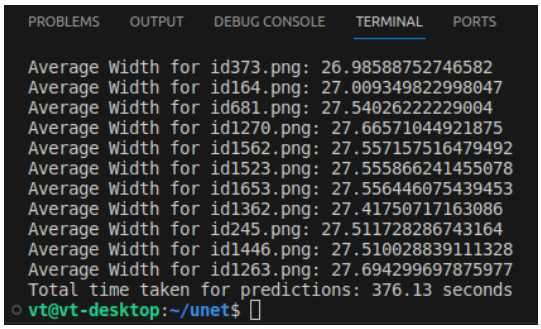


Figure 4.28: Execution time on the edge-device.

4.7 Implementation Summary

Throughout pivoting meetings in Appendix C, a set of desired functions was discovered through validated learning method in section 1.1. Upon this the solution to meet these requirements was designed by following product architecture method by S. Eppinger in subsection 3.1.1. Thus, this chapter has described the implementation process on achieving proof of concept for the proposed solution design.

Chapter 5

Discussion

This chapter will include the discussion on the solution implementation, but also the test results of the solution performance. The chapter will be divided respectively to the established sequence of this report.

5.1 Solution Design

Core Functionality The main functionality of the solution, is to detect the flange within a given frame and provide a segmentation mask of the flange for further dimensioning. The change in flange width is believed to reflect the change draw-in during the production, which is assumed to be a fair metric for the part general quality prediction. Although even if there is no correlation between flange' width and draw-in, the developed solution was still designed to automatize and optimize existing manual procedure on one of the steps in quality assessment during test stamps. In subsection 2.2.2 it was defined that such flange measurement procedure can take up to 10 minutes per part, which yield 2-3 hours daily of halted production as mentioned in Appendix B. Thus, the designed solution can still bring value through automation of existing task.

5.2 Implementation

Data Acquisition The image data during this semester was acquired through a standard video recording of the whole test production sequence, resulting in several separate videos. Some of the videos contained production idling, which added the complexity for the key frame extraction. Due to imperfection of the developed script, some of the extracted frames were still very blurred, while even more frames were empty for parts, or one part that were idling for an extended period. This added to the data preparation time. Ideally, the data should be acquired through

use of the press-machine' PLC, where the system would only take images upon specific tool setting. This way, less data would be stored, and remove the necessity of sophisticated script development for key frame extraction from a video.

U-Net

In subsection 4.4.1 the U-Net model has provided satisfactory results by properly segmenting detected flanges. The model was trained for 30 epochs for each of the 3 folds without image auto-augmentation, and for comparison the corresponding training, but with image auto-augmentation. According to the training metrics, both models converged to approximately same level, which raised a question of augmentation necessity for the given data. As can be seen in Figure 5.1 the same testing image with slight blur is undersegmented without training on augmented data. Contrary the model trained on augmented data perform as intended on the same test image with blur. This may indicate that the image augmentation should always be present during training of a CNN-based model.

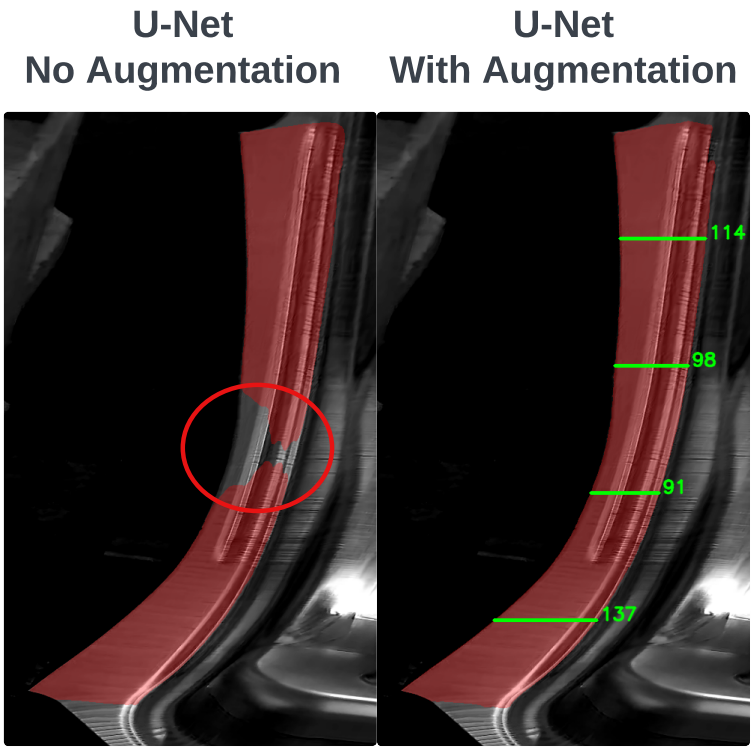


Figure 5.1: Comparison of image augmentation effect on U-nets performance.

YOLOv8

In subsection 4.3.1 the YOLOv8 was determined as one of the most popular models for image analysis. It is widespread and has frequent major updates as the suffix "v8" signifies. Originally in 2015 the model was designed for real-time object

detection, but since has grown with multiple extensions among other things segmentation functionality used in this project. [22]

The purpose of training YOLOv8 for this project was to test a different architecture that was designed for fast inference, in case the solution is desired to be used on a real-time video feed. The model is known to be less precise than the models that are designed for high resolution segmentation like U-Net, but the goal was to test this trade-off between inference speed and mask precision. Despite the unsatisfactory segmentation results after 30 epochs training for each fold in subsection 4.4.2, it was decided to train the model on 70 epochs, to see if the model would provide better results. As can be seen in Figure 5.2, prolonged training did remove false positives along top of the image, but in general it still provides oversegmented flange that can not be used for further dimension analysis. Therefore it was considered that even with a larger dataset and far more prolonged training, it is questionable, if models of this type are suitable for the given task.

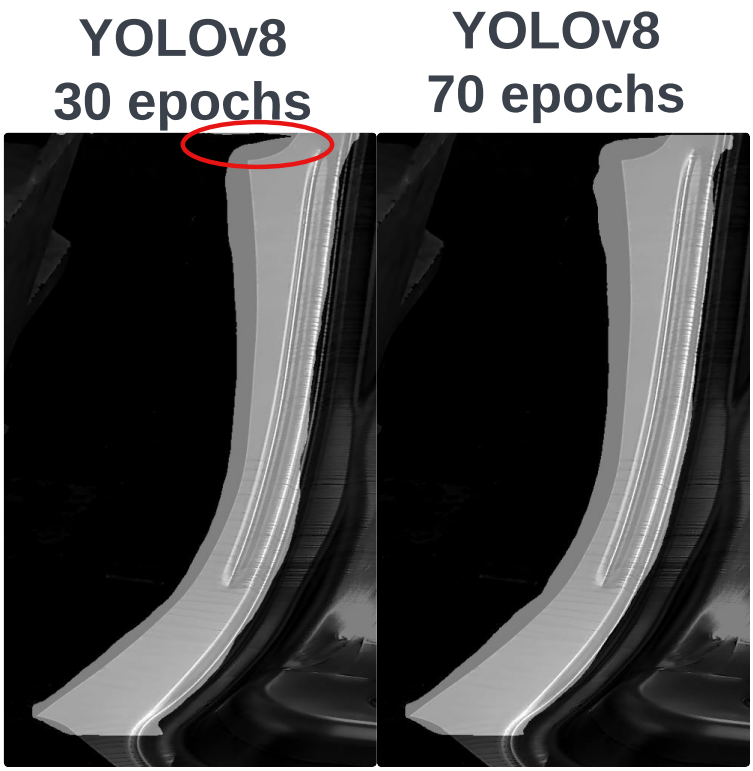


Figure 5.2: Comparison of YOLOv8 predictions after prolonged training.

5.3 Test Results and Validation

Team of scientists in chapter has provided laser measurements of the draw-in that can be seen in Figure 5.3. In order to test, if the draw-in can be estimated through flange width measurement, the correlation between change of retrieved draw-in measurements and change of the flange width measurements were calculated.

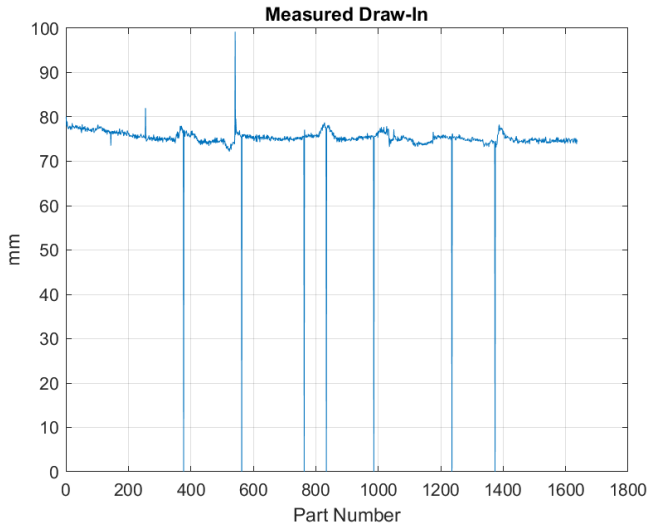


Figure 5.3: Laser data by the scientists in chapter .

For this a Matlab script was developed which can be seen in Appendix F. The script would calculate correlation between the draw-in measurements and corresponding part number flange measurement for each of the four measurement points. All calculations can be found in appendix, while in this section the average of the three folds is presented below;

$$
\begin{aligned}
20\% &\approx -0.0789 \\
40\% &\approx -0.1131 \\
60\% &\approx -0.1124 \\
80\% &\approx -0.0916
\end{aligned}
$$

Since an increase in a flange's width is supposed to decrease the part's draw-in, the correlation should be negative. As seen from the results above, the correlation is negative but very close to zero, meaning that there is weak to no correlation between a flange's width and the actual draw-in. There could be several reasons for this weak correlation, such as the position of the laser not being aligned with any of the four arbitrarily chosen areas of the flange, or the free body movement of the whole part strongly impacting the laser measurement on one side. Alternatively, a weak correlation could also indicate no direct relationship between a flange's

width and the draw-in.

Observing the model results of change in flange' width in Figure 4.23 and comparing that to the laser data, it can be seen that in the beginning of production, draw-in decrease with approximately same rate as the flange's width increase until around part 350 for both datasets. Consequently delimited correlation calculation was calculated for that segment for each of the three folds and averaged, which can be seen below;

$$20\% \approx -0.7461$$
$$40\% \approx -0.7728$$
$$60\% \approx -0.7511$$
$$80\% \approx -0.4016$$

This yield stronger correlation much closer to 1 than correlation for the whole dataset. This could due to imperfection method in numbering of the parts during the frame extraction from the video, where some frames needed to be removed manually due to empty idling station, or strong blur. This could explain worsening of the correlation between measurements over time. The other reason could be the aforementioned free body movement of the whole flange, which together with lack of camera calibration could result in different measurements in model prediction. Camera could also drift during the production and impact the flange' appearance in image, but this was not proven.

Camera Calibration
Separate camera calibration for the exact purpose of the flange measurement could provide more metrics on the system performance. With custom calibration the flange' width measurements in pixels could be converted into millimeters, which would asses the system precision performance, but also provide more intuitive measurements data for the shop-floor workers. Converted measurements could also be more beneficial for later research in mapping of the press-machine parameters to the output part geometry.

Camera Resolution and Optical Zoom
From the model training perspective it could be beneficial to test train the model on different lower resolutions, which would decrease the computational load, hence, reduce the training time. Also it could be interesting to test different optical zoom settings to find optimal zoom to resolution ratio for best model precision and lowest resolution possible.

Chapter 6

Conclusion

Concluding this master's thesis on manufacturing technology, a long-term collaboration with Volvo in the field of metal stamping is completed. The primary objective was to develop a product capable of automating and optimizing the manual task of quality assessment during production, thereby contributing to the overall advancement of automatic quality control systems. This product, built on machine learning-based models, was evaluated for its feasibility.

During this project, the Volvo car body components production plant in Olofström was visited. The student of this project engaded in close dialogue with the palnt staff, using the validated learning method to discuss and develop a solution concept with the highest possible viability. The designed solution was implemented and tested on two models such as U-Net and YOLOv8. While the deployed U-Net met the required production cycle time, it was proven to be slower than the YOLOv8. However, U-Net outperformed YOLOv8 with a better segmentation mask, which could be used for dimensioning of the flanges.

Regarding the final problem statement:

Can the change in flange width of produced parts be used for draw-in estimation?

Based on test results, there is no strong evidence to suggest that the change in draw-in can be directly estimated using the change in flange width during production. Nevertheless, the designed solution could be considered as an additional measure to track the production process at the press-line.

Chapter 7

Future Work

As this master's thesis work ends, this chapter will briefly describe potential future work that can be built upon this work and expand the capabilities of the proposed solution.

7.1 Implementation

In chapter 3 a set of required functions were established based on the iterative dialogue with the factory staff and were divided with respect to necessity and implementation feasibility with the given time frame.

While detection of flanges within a video stream is not a proven necessity, some other requirements from the Table 3.1 categorized as *Won't Have* would improve the solution's value for the company according to the staff. These additional function are as follows:

- **Measure the flange in user defined area**
  During the dialogues with the staff it was revealed that the press-machine operators measure the flange width different places depending on the given part. Therefore instead of student defined four static areas as demonstration, it would be highly beneficial to improve the solution to be capable of user input in order to define the desired area for measurement.

- **Detect surface defects**
  Due to the given edge-device's computational time, there is a room for implementation of an assistive model, that could be trained to detect surface defects such as cracks, which will also be an additional quality control measure.

- **Store press-machine parameters with respective output**
  If the solution records the press-machine parameters with the measured output, it can than be used for further research in mapping the input to output, or for quality prediction in production. This was noted as the highest desired feature by the quality education responsible in Appendix B.
- **Compensate for increase or decrease of the flange width**
  Ideally, after the research on mapping input to output, another model can be trained to control the press-line output in a closed-loop system.
- **Display Relevant Production Data**
  Currently the system displays only the measurement plot, and stores measurements and predictions with overlays in respective folders. Ideally a graphical user-interface should be designed and developed for the most efficient use of the solution.

7.2 Core Model

During this semester two different CNN-based models have been implemented and tested, where U-Net was chosen as the most-suitable model for the given task. Despite the near perfect segmentation of the flanges, on relatively small dataset, the future work could focus instead of testing other models, but optimize this model for the exact task. As a suggestion, the model could potentially work faster if the model's architecture is redesigned to take gray-scale images as input instead of RGB images, presumably reducing the computational load, hence, improving the model's prediction speed.

Bibliography

[1] William J Abernathy, James M Utterback, et al. "Patterns of industrial innovation". In: *Technology review* 80.7 (1978), pp. 40–47.

[2] SICK AG. *Machine Vision*. https://www.sick.com/us/en/products-and-solutions/products/machine-vision-and-identification/machine-vision/c/g114858. Online; accessed 29 April 2024.

[3] Baeldung. *What Is a Learning Curve in Machine Learning?* Baeldung. 2024. URL: https://www.baeldung.com/cs/learning-curve-ml.

[4] Volvo Cars. *Our 2023 Annual and Sustainability Report is here!* Volvo Cars. 2024. URL: https://www.volvocars.com/intl/news/corporate/our2023-annual-and-sustainability-report-is-here/.

[5] Volvo Cars. *Our heritage*. Volvo Cars. 2024. URL: https://www.volvocars.com/intl/v/our-heritage.

[6] Volvo Cars. *Our Story*. Volvo Cars. 2024. URL: https://www.volvocars.com/intl/v/our-story.

[7] Yanjiao Chen et al. "Deep Learning on Mobile and Embedded Devices: State-of-the-art, Challenges, and Future Directions". In: 53.4 (2020). ISSN: 0360-0300. DOI: 10.1145/3398209. URL: https://doi-org.zorac.aub.aau.dk/10.1145/3398209.

[8] Cognex. *Cognex Product Guide*. https://www.cognex.com/downloads/edge-learning-applications-guide. Online; accessed 29 April 2024.

[9] European comission. *End-of-life vehicles Regulation*. URL: https://environment.ec.europa.eu/topics/waste-and-recycling/end-life-vehicles/end-life-vehicles-regulation_en. (accessed: 16.03.2024).

[10] Atlassian Community. *Understanding the MoSCoW prioritization How to implement it into your project*. URL: https://community.atlassian.com/t5/App-Central/Understanding-the-MoSCoW-prioritization-How-to-implement-it-into/ba-p/2463999. (accessed: 22.04.2024).

[11] The SciPy community. *scipy.signal.find_peaks*. https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html. Accessed: 07 May 2024. 2024.

[12] Computervisioneng. *image-segmentation-yolov8*. Github. 2023. URL: https://github.com/computervisioneng/image-segmentation-yolov8?tab=readme-ov-file.

[13] Datagen. *Convolutional Neural Network: Benefits, Types, and Applications*. https://datagen.tech/guides/computer-vision/cnn-convolutional-neural-network/. 2024.

[14] Nvidia Developer. *Buy the Latest Jetson Products*. https://developer.nvidia.com/buy-jetson?product=all&location=DK. Accessed: 02 May 2024. 2024.

[15] Google For Developers. *Problem Framing*. https://developers.google.com/machine-learning/problem-framing/ml-framing. Accessed: 07 May 2024. 2023.

[16] Scikit learn developers. *Precision-Recall*. Scikit-learn. 2024. URL: https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html.

[17] S. Eppinger and K. Ulrich. *Product Design and Development*. McGraw-Hill Education, 2015. ISBN: 9781259297137. URL: https://books.google.dk/books?id=S7FZCgAAQBAJ.

[18] HowMuch.one. *Buy the Latest Jetson Products*. https://howmuch.one/product/average-ssd-1tb-nvme/price-history. Accessed: 02 May 2024. 2024.

[19] Harsh Jain. *How to deal Overfitting In Convolutional Neural Network*. https://www.kaggle.com/discussions/getting-started/236883. Accessed: 07 May 2024. 2021.

[20] kangalow. *Jetson Orin Nano Tutorial: SSD Install, Boot, and JetPack Setup*. JetsonHacks. 2023. URL: https://jetsonhacks.com/2023/05/30/jetson-orin-nano-tutorial-ssd-install-boot-and-jetpack-setup/.

[21] Lucy Ellen Lwakatare et al. "Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions". In: *Information and Software Technology* 127 (2020), p. 106368. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2020.106368. URL: https://www.sciencedirect.com/science/article/pii/S0950584920301373.

[22] Joseph Nelson. *What is YOLO? The Ultimate Guide*. Roboflow. 2024. URL: https://blog.roboflow.com/guide-to-yolo-models/.

[23] Nvidia. *Preparing a Jetson Developer Kit for Use*. Nvidia. 2024. URL: https://docs.nvidia.com/jetson/archives/r36.3/DeveloperGuide/IN/QuickStart.html#preparing-a-jetson-developer-kit-for-use.

[24] Aladdin Persson. *PyTorch Image Segmentation Tutorial with U-NET: everything from scratch baby*. YouTube. 2021. URL: https://www.youtube.com/watch?v=IHq1t7NxS8k&t=1s.

[25] Jason Brownlee PhD. *A Gentle Introduction to k-fold Cross-Validation*. Machine Learning Mastery. 2023. URL: https://machinelearningmastery.com/k-fold-cross-validation/.

[26] Eric Ries. *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Currency, 2011.

[27] Roboflow. *YOLOv8 Instance Segmentation*. Roboflow. 2024. URL: https://roboflow.com/model/yolov8-instance-segmentation.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597.

[29] Ranjan Sapkota, Dawood Ahmed, and Manoj Karkee. *Comparing YOLOv8 and Mask RCNN for object segmentation in complex orchard environments*. 2023. arXiv: 2312.07935 [cs.CV].

[30] Yaroslav Semenyuk. *FrameEX*. https://github.com/Yaro-slav/FrameEx. 2024.

[31] Yaroslav Semenyuk. *P10*. 2024. URL: https://github.com/Yaro-slav/P10.

[32] Daniel Strigl, Klaus Kofler, and Stefan Podlipnig. "Performance and Scalability of GPU-Based Convolutional Neural Networks". In: *2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing*. 2010, pp. 317–324. DOI: 10.1109/PDP.2010.43.

[33] Agmanic Vision. *VISION TOOLS: COGNEX IN-SIGHT EXPLORER*. https://agmanic.com/vision-tools-cognex-in-sight-explorer/. Online; accessed 29 April 2024.

[34] Wikipedia contributors. *Jaccard index — Wikipedia, The Free Encyclopedia*. [Online; accessed 18-May-2024]. 2024. URL: https://en.wikipedia.org/w/index.php?title=Jaccard_index&oldid=1220812875.

Appendix A

Project Proposal

Master Thesis Proposal – 2023/2024

Title:	Optical Draw-In Measurement and Compensation in an Automotive Press Line
Collaborators:	Volvo Cars, Sweden Blekinge Institute of Technology, Sweden TATA Steel, The Netherlands
Location:	Remotely, but some time spent in Olofström, Sweden required. Opportunity to go to IJmuiden, The Netherlands possible.
Industrial Supervisor:	Johan Pilthammar, PhD Technical Expert, Simulation Driven Stamping Development Volvo Cars
Duration:	6-12 months

Abstract

With the increased requirement for sustainable solutions in the manufacturing industry, smart technologies such as vision systems combined with Machine Learning and Artificial Intelligence becomes increasingly interesting.

In the automotive industry, an area for application is when it comes to the manufacturing of body components through the sheet metal forming process. Flat sheets of metal are transformed into complex geometries, where during the process, the edge of the sheet is draw-in from its original position. The measurement of the draw-in is of interest as it is used as a quality control point for the components, comparing it to the designed component design.
During the forming of the component, the edge of the sheet is hidden by the tools used for the forming process, why only a comparison of a before and after image is possible.
The information about the measured draw-in can be directly linked to various process parameters such as the blankholder force, lubrication amount etc. Therefore, with an accurate vision system measuring the draw-in, this makes the process the perfect candidate for an implementation of Machine Learning and Artificial Neural Networks for the compensation of drifting process parameters.

Figure A.1: This is the proposal by Volvo Cars

Appendix B

Interview Transcript

Below is the transcript of the two larger meeting with shop floor mangers, and with one responsible for quality control education at the plant. Black text represent meeting notes during the meeting with shop floor managers, while the red text is the updates on the same questions gained during the meeting with quality control expert.

1. How is quality control performed on the produced items?

a. Which types of information are being collected during the production?
- Temporarily stored information on the machine parameters.
- Temporarily stored size measurements of the random samples (Usually beginning, mid and end of the batch.
- (Not all machines store the same amount of information, on some more on some less, depending on the history of problems with the current type of product and machine)

b. How are quality issues detected in the production?
- Manual inspections on random samples, or during the test stamps prior to production.
- Highly based on the experience of the operators and with severe time limitations.
- During the process of determining the test stamps or "press windows" they stop machine and manually measure the flanges as well.

c. Is there any other inspection afterwards?
- No, if the part is accepted on the conveyor after the press machine, then it is packed and sent further. But often if something comes up, then it is mostly the supplier that detects the defect. In that case the part or whole car body is sent back, investigation is made, and the part/whole body is scraped. In some cases, the whole batch can be revoked. It is always based on the severity of the defect.

d. How is every part tracked through the production line?
- There is a press counter, that counts the number of strokes for a given batch. But it depends on the machine operator, where sometimes it is not recorded, and there is no more information besides the batch number.

e. Which issues/defects are known, but cannot be detected by the machine operators?
- The material properties problem is both known and unknown, it can not be detected directly in the production, but if problem occurs, operators can deduct that there is something wrong with the material. (We have been warned by the quality instructor, that the machine operators are fast to blame the material, but 99,9% of defects come from the production process and not the material) Then further testing of the material can be done, where if it does not meet the standards, it can be sent back to the supplier.
- Amount of lubrication is also to some extent unknown due to the migration of the lubrication to the edges.
- (Quality instructor) Material thinning, and material tension are known defects, but almost impossible to detect, where a lot of these slip through. (I have images of examples)

f. How are the detected quality issues dealt with? Are there instructions, or is it done by experience?
- There is some education for new workers that introduces the defects, and how they can be compensated, but mostly the defects are detected solely on the experience as well as their compensation. This applies to other decisions in the production – based on experience and dedication of the individual worker.
- (Quality instructor) They have instructions, but it is mostly for the new workers, where older workers rarely get the opportunity to get updated instructions. Also

there is a high degree of uncertainty regarding the press machine parameters and the output, which makes it difficult to make a clear guideline on how to compensate for the errors in production.

2. Detailed description of the T9 production process?

a. **Description of production process**
   - Information is on the way exactly for T9, but we have received a production description of a similar process. I have now the production layout for T9.

b. **Where is it possible to perform visual inspections in that production process?**
   - According to workers, but also visible from the production process, it is impossible to perform flange measurement other stations besides the main forming station, because on the next station they cut off the flanges.

c. **Should the visual inspection be performed several times during the production?**
   - No need according to operators. Only if it is a surface defects detection at the end, to help spot cracks or wrinkles. Or thinning + tension.

d. **If there is an automatic measuring system, how could it impact the production process?**
   - It would provide the workers with an additional tool, to help them do the flange measurement faster, so they have more resources to decide on the production plan.

e. **Is there an interest to store the inspections performed by an automatic measuring system?**
   - Without an adequate packing, transportation and storing system, logging the data would be of no use in general.
   - After a meeting with "quality inspection" instructor: Currently they have no measures to track and store the measurements of produced parts and the machine parameters, which are crucial to determine the relation and effect of the machine parameters and output. Therefore, an automatic measuring system for quality responsible, but also for R&D department would be of high value.

f. **How often is production process paused, is it expensive to pause the production process?**

   Several times during the production day. Every time they need to change a batch of the material, where they need to do test stamps to determine the approx. machine parameters / tolerances for the given material. Where to measure flanges of each test stamp takes up to 10 minutes. According to the Volvo representative, total time spent each day on these test stamps is around 2-3 hours. Also, according to the same representative, if the solution could automatize this flange measurement process, the whole plant could save 20-30 mio. SEK solely on the salaries of these 5 workers. On addition to that there are also expanses due to production stop, logistics and electricity.

Appendix C

Pivotal Meetings

This chapter will include important meetings held during the spring semester 2024, which were used as a part of *Lean Startup* methodology. As mentioned in section 1.1 central element of this methodology is the validated learning, which implies engaging closely with the customers to validate and refine the project's direction on product idea, or to pivot from the solution to something completely different. Through iterative cycles of feedback and learning between meetings, the group of this project were able to gain a higher product success probability.

In the course of this semester, two pivotal meetings were conducted with different departments within the company, that were used as learning about the objective, but also for feedback on the product concept. Transcript of these two meetings can be found in Appendix B.Additionally, regular consultations with the designated contact person of this project from the company and the supervisors from the university, were of crucial role in grouping and processing learning gathered during these pivotal meetings.

C.1 Preliminary Meeting

Date - 23/02/2024

Participants - Student of this project together with student from the parallel project from Aalborg University, Volvo representative of this project.

Objectives - To prepare for the upcomming meeting with shop floor managers, and get feedback on the current product concept.

Summary
This brief meeting was organized before the session with shop floor managers to

devise the interview questions and to introduce the Volvo project representative to the current product concepts. The results achieved during the fall semester of 2023 were presented, as described in subsection 2.3.1. During this presentation, the contact person suggested pivoting from measuring the average width of the flanges to providing several static measuring points. Additionally, it was proposed to measure from the static areas of the press machine to the edges of the metal sheet instead.

Outcome

Based on this feedback, two conceptual figures were developed in preparation for the upcoming meeting with the shop floor managers as a part of validated learning activity. The first concept, as seen in Figure C.1, pivoted from measuring an average width across the entire flange to using 4 static points, thereby offering four distinct width measurements of each flange for the machine operators. The second concept, as seen in Figure C.2, involves storing the historical data of measured width changes at each flange point on a graph for each flange.

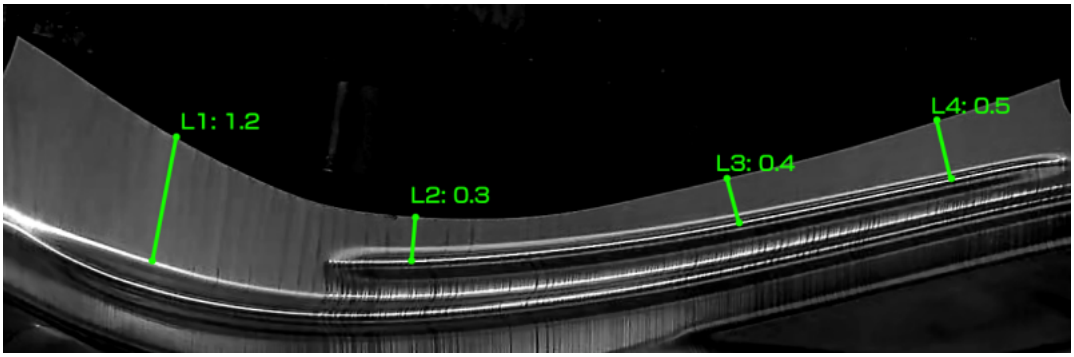


Figure C.1: Part of the interface concept based on the feedback.

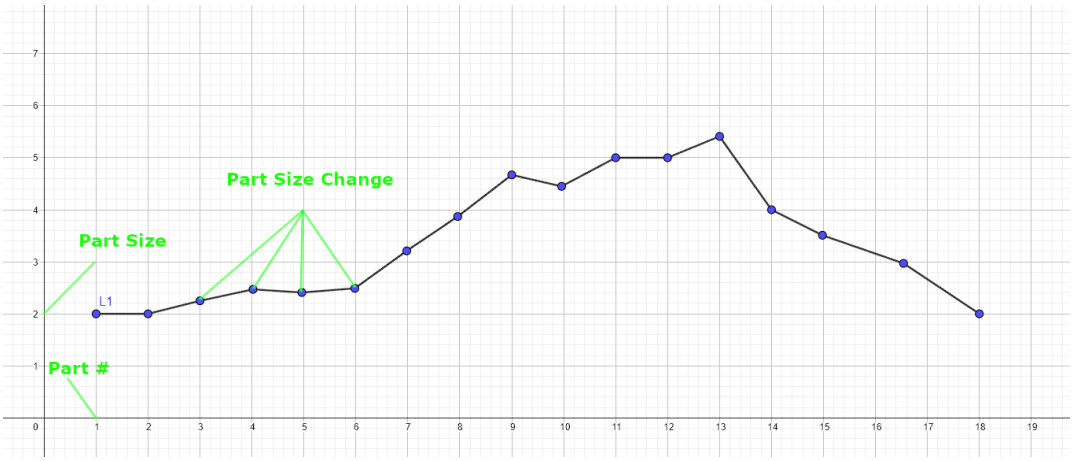


Figure C.2: Other part of the interface concept based on the feedback.

C.2 Meeting with Shop Floor Managers

Date - 28/02/2024

Participants - Student of this project together with student from the parallel project from Aalborg University, shop floor managers from Volvo plant in Olofström.

Objectives - Learn about the production process, learn about quality control in production, and get feedback on the current product concept.

Summary
This meeting was fruitful regarding the production process at the press line, where the group of the project received detailed production plan and description on each production step, which was described in subsection 2.2.1. During production, machine parameters and the sizes of random samples are temporarily recorded, with the extent of data varying across press lines based on historical performance and issues.

The group also learned about the quality control at the press line. Quality defects are mainly identified through manual inspections by experienced operators, either on random samples or during pre-production test stamps on the beginning of a new material batch, which can take up to 10 minutes per produced part. Once parts pass the conveyor and are packed, the potential defects are rarely detected later in production, but could be identified by the costumer. This can lead to costly investigations through the batch number and potential withdrawal of produced products. Some defects as material thinning are impossible to detect without cross

section analysis, while general defects can only be detected during 4 second time frame.
Lastly, during this meeting the previously achieved concept was demonstrated and described as in subsection 2.3.1 alongside new figures as in Figure C.1 and in Figure C.2. These concepts received positive feedback with minor proposals for improvement, while the proposal to measure from the press machine to the metal sheet was deemed as of little to no use.

Outcome
As a result of this meeting, the concept was refined to focus solely on measuring the flange width, which is within the existing routines of the press teams. The shop floor managers clarified that the measurement locations for the flange vary depending on the part type. Therefore, they suggested pivoting away from multiple fixed measurement points to a dynamic system. Such dynamic system would allow the press team to select the measurement location themselves, which would be most suitable for the production quality control.

C.3 Meeting with Quality Supervisor

Date - 29/02/2024

Participants - Student of this project together with student from the parallel project from Aalborg University, quality supervisor and a simulation engineer from Volvo plant in Olofström.

Objectives - Learn about quality control and defects in production, but also to get feedback on the current product concept.

Summary
This meeting was arranged with the assist from shop floor managers and a simulation engineer acquainted during a coffee break. The meeting aimed to gain further insights into shop floor quality control and, more importantly, to deepen understanding of the characteristics and appearance of production defects.

The group of this project received a presentation from the quality supervisor on the defect types and their frequency, where a major part of the information was confidential and could not make it into this report. Still the group have received more details on questions from the previous interview, but also confirmed uncertainties from the shop floor managers on account of quality education standards, former and ladder can be found in Appendix B, where red colored text represent the information gained from this meeting. Lastly the group was presented for physical

examples of the defects that were described in section 2.1.

Lastly the presentation of the concept as illustrated in subsection 2.3.1 with the updated direction shown in Figure C.1 and in Figure C.2, received significant interest from the quality supervisor. He noted that there were limited means of collecting consistent production data for researching the effect of machine parameters on the draw-in of metal sheets in the press. And thus such tool could assist in consistent data collection.

Outcome
This meeting provided additional detailed information on the topic of production defects, while also supported the interest in the product concept. There were no suggestions for additional features, but the high interest of the quality supervisor indicated the fact that the R&D department could be a more interested stakeholder of the suggested concept.

C.4 Summary of Meetings

These meetings, supported by the *Lean Startup* methodology, were essential for gathering feedback and validating the project's direction. Through discussions with company representatives and shop floor managers, the group of this project engaged in a process of iterative learning, leading to the refinement of their product concept based on stakeholder input. This approach allowed for a flexible development path, by the use of a pivot-or-persevere decision making process, that guided the project towards a solution more closely aligned with the customer needs and practical applications. The chapter demonstrates the application of Lean Startup principles in an metal press forming context, emphasizing the importance of close potential customer engagement, feedback driven iteration, and adaptability in concept development.

Appendix D

Appendix D - Cost and Components

This section contain the price origin of of components mentioned in section 3.2, which were discovered throughout the fall semester 2023 project.

D.1 Prototype Solution Components

Camera
The link to the product store: Amazon.de

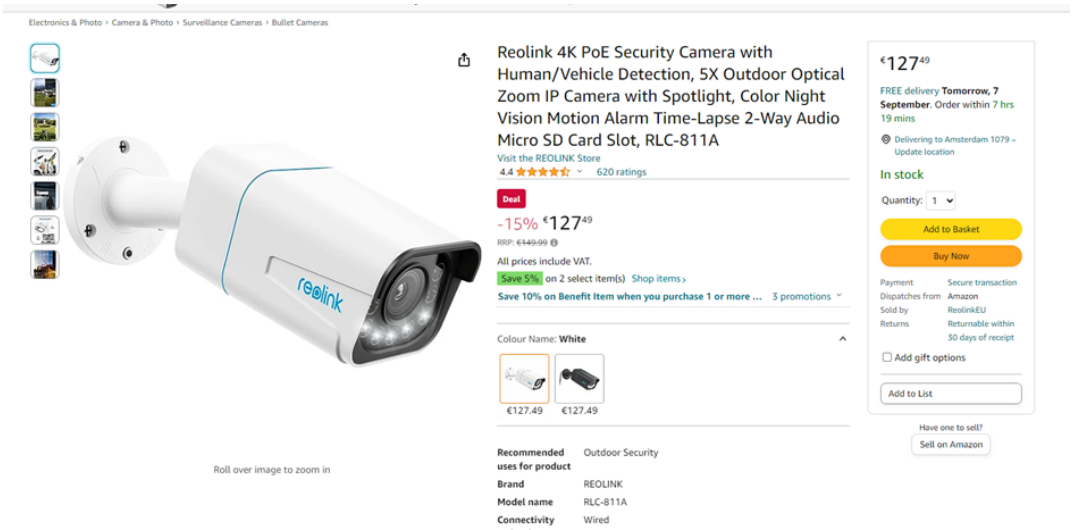


Figure D.1: Price of the ordered camera.

Edge Device
The link to the product store: Amazon.se

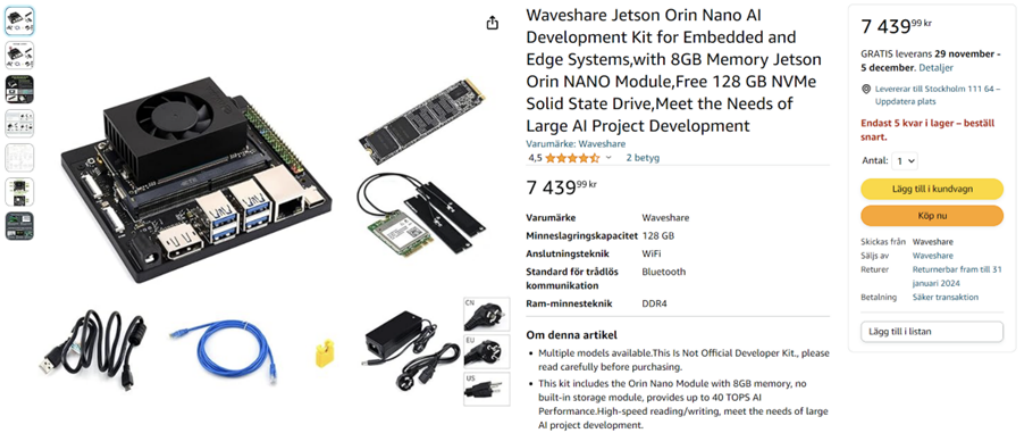


Figure D.2: Price of the ordered edge device.

Memory Storage Hub

The link to the product store: Amazon.de

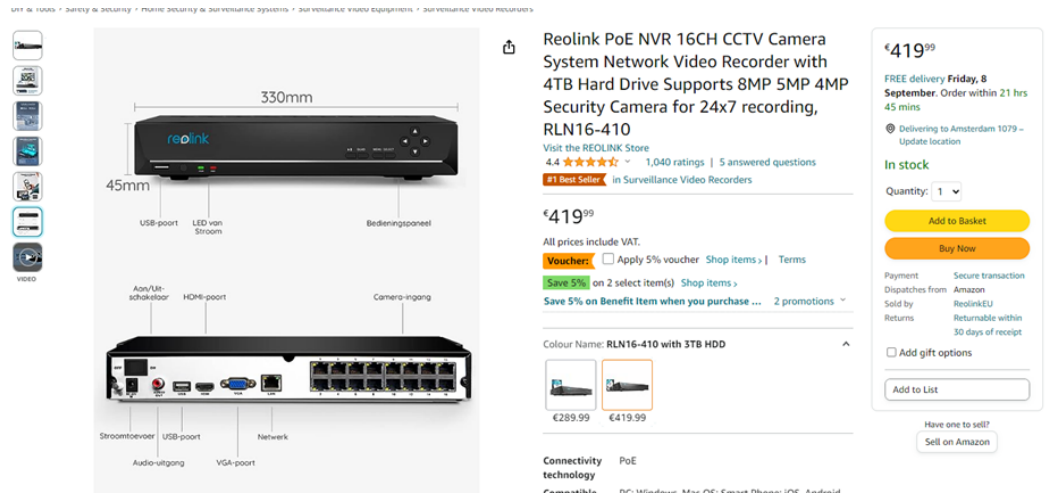


Figure D.3: Price of the ordered memory storage hub.

Computer

The link to the product store: Elgiganten.dk

Figure D.4: Price of the used computer for model training.

D.2 Further Implementation Components

The list of components for the computer originates from Aalborg University computer that was assembled at the university for the purpose of machine learning researches. While the edge device, camera and memory storage hub, should also be bought for this solution. The same list of components was also suggested during fall semester 2023 project.

Name of the product: **Lenovo Workstation P620**, but with changed components by the Aalborg University that are as follows in Figure D.5, while the price of this workstation is according to supervisor of this thesis estimated to be around DKK 50,000 or EUR 6,704.75.

Type	Model and Characteristics	Quantity
Processor	AMD Threadripper PRO 5955WX Processor (4.00 GHz up to 4.50 GHz)	QTY: 1
CPU Memory	32GB DDR4 3200MHz ECC RDIMM	QTY: 4
Graphics	NVIDIA RTX A6000 48GB (4xDP), LE	QTY: 1
M.2 SSD Memory	2 TB SSD M.2 2280 PCIe Gen4 TLC Opal	QTY: 1
OS	Windows 11 Pro 64	
Power Supply	Tower WRX80 92% Power 1000W	QTY: 1
Thunderbolt I/O	Rear Thunderbolt Card	QTY: 1
Keyboard	Nordic (included)	QTY: 1
Mouse	USB Calliope Mouse Black	QTY: 1
Motherboard	MB AMD V 1.2	QTY: 1
Base Warranty	3 Years On-site	

Figure D.5: List of components that are included in Aau's researching computer.

Appendix E

Appendix E - Model Training and Test Results

This chapter will include all possible screenshot proof of the conducted model training and testing.

E.1 U-Net Training

Fold 2 training results on non augmented data:

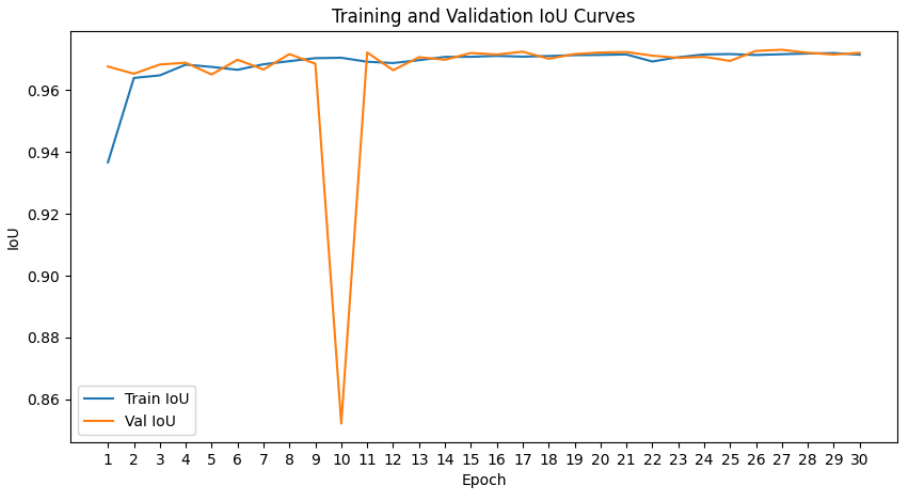


Figure E.1: Fold 2 IoU curves.

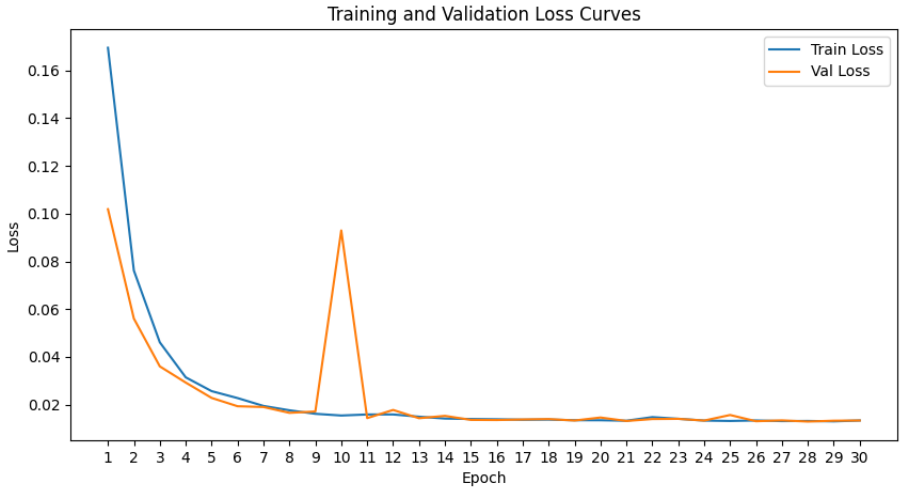


Figure E.2: Fold 2 loss curves.

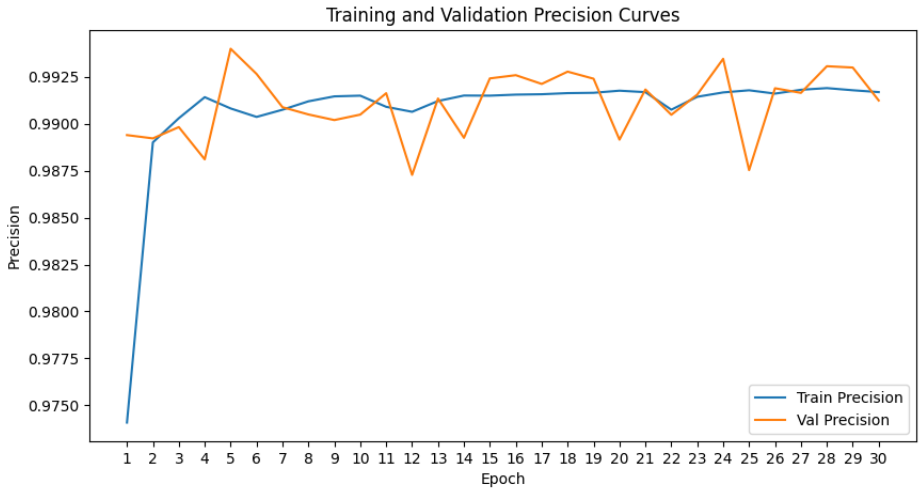


Figure E.3: Fold 2 precision curves.

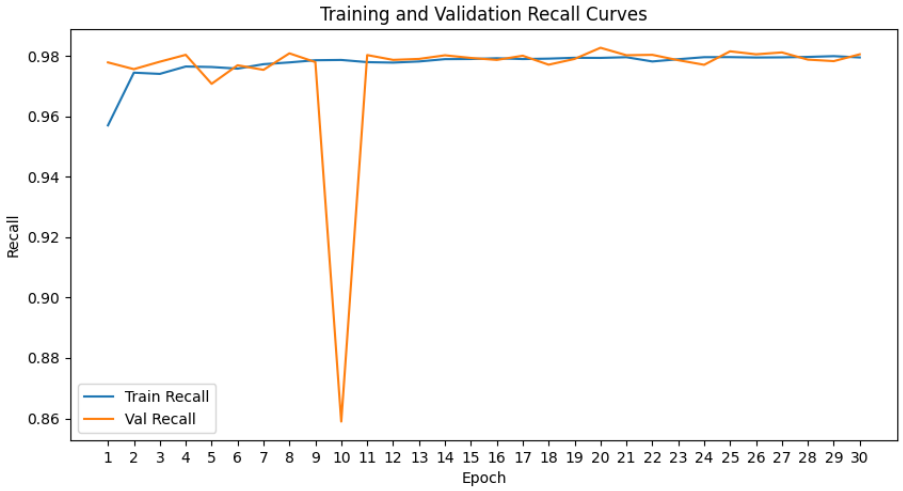


Figure E.4: Fold 2 loss recall.

Fold 2 training results on augmented data:

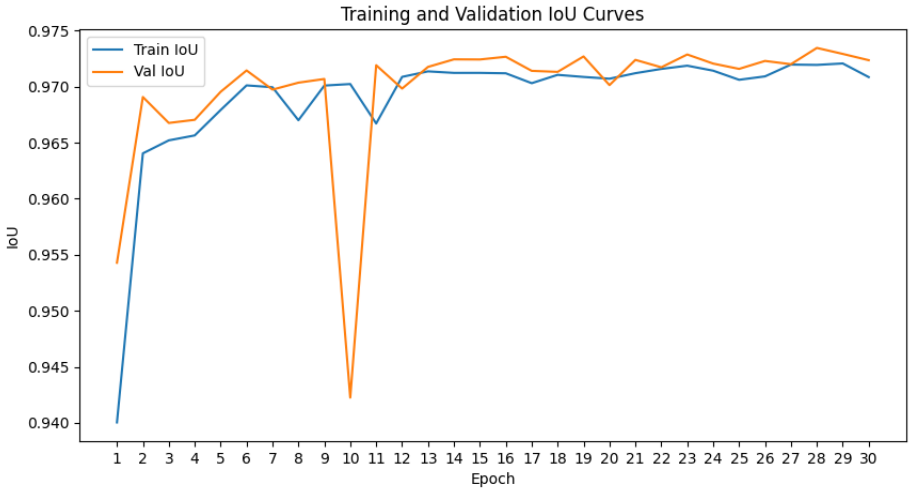


Figure E.5: Fold 2 IoU on augmented data.

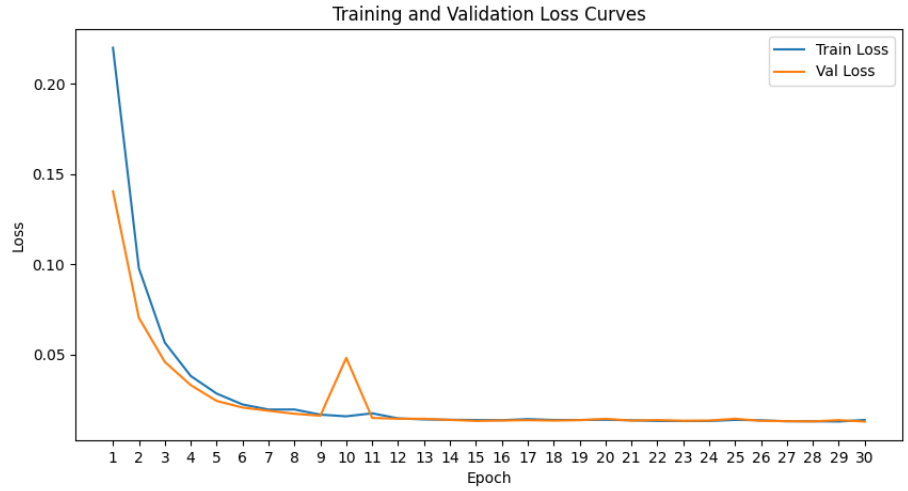


Figure E.6: Fold 2 loss on augmented data.

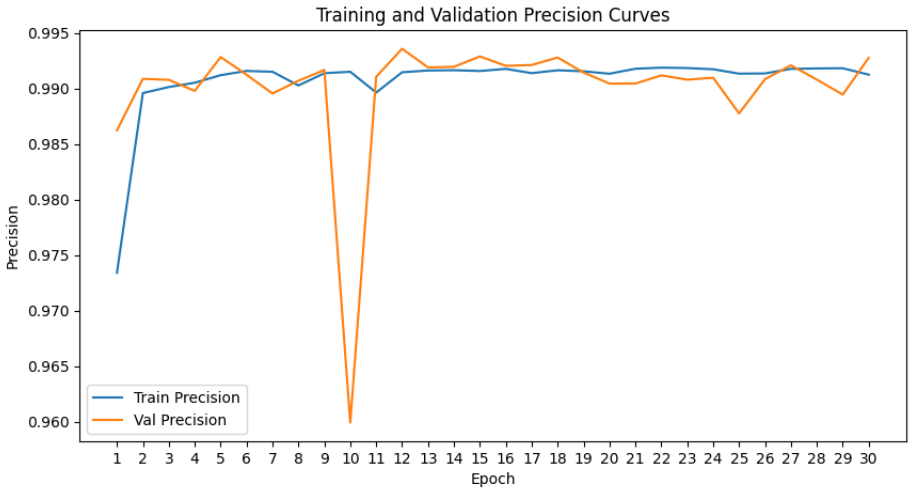


Figure E.7: Fold 2 precision on augmented data.

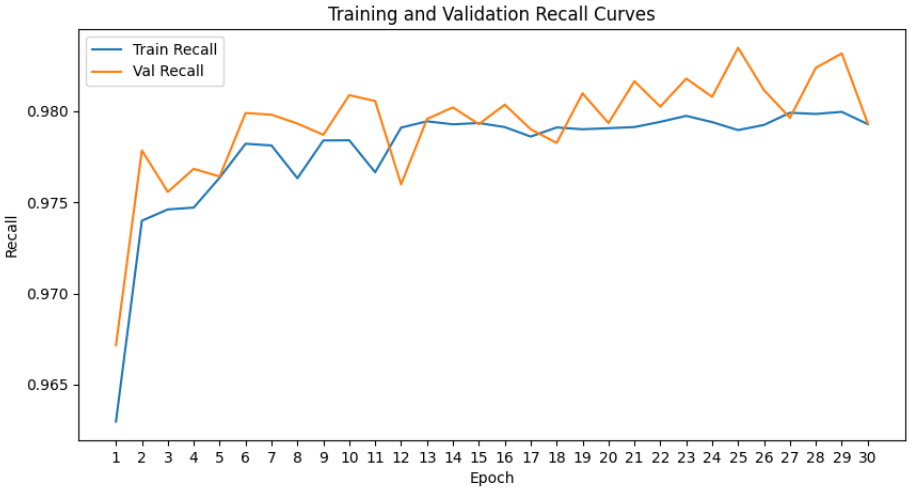


Figure E.8: Fold 2 recall on augmented data.

Fold 3 training results on non augmented data:

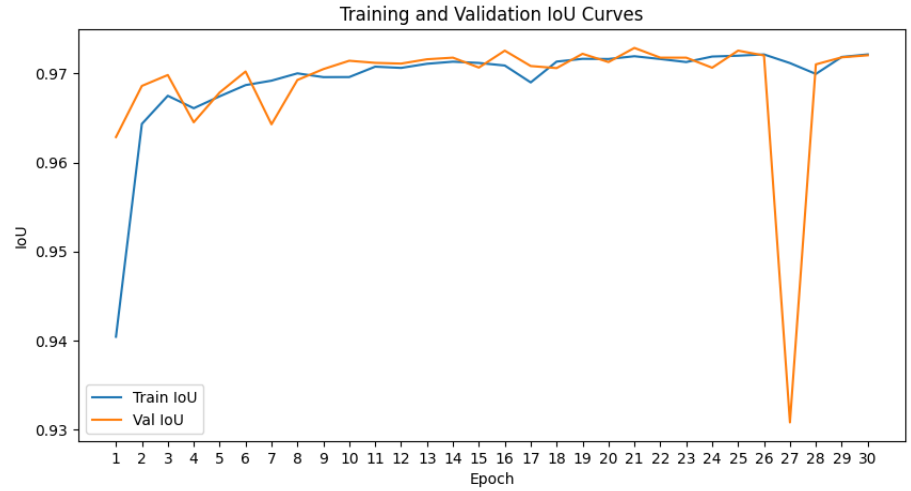


Figure E.9: Fold 3 IoU curves.

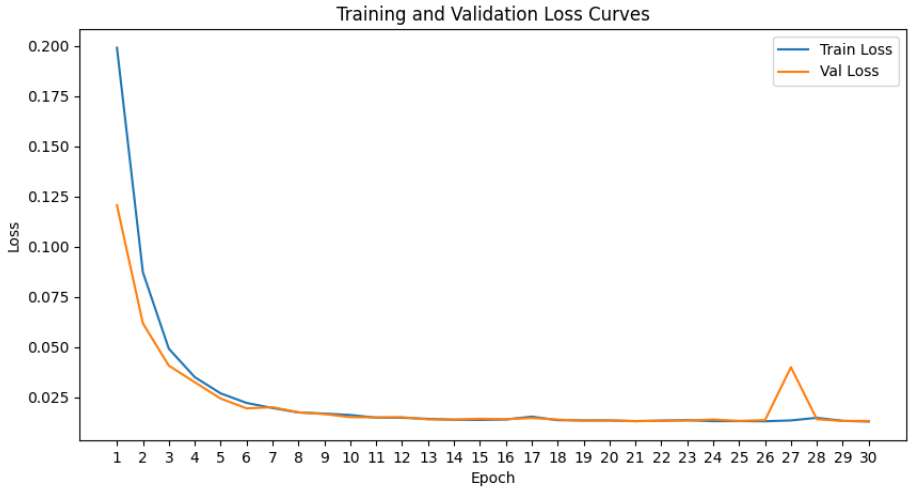


Figure E.10: Fold 3 loss curves.

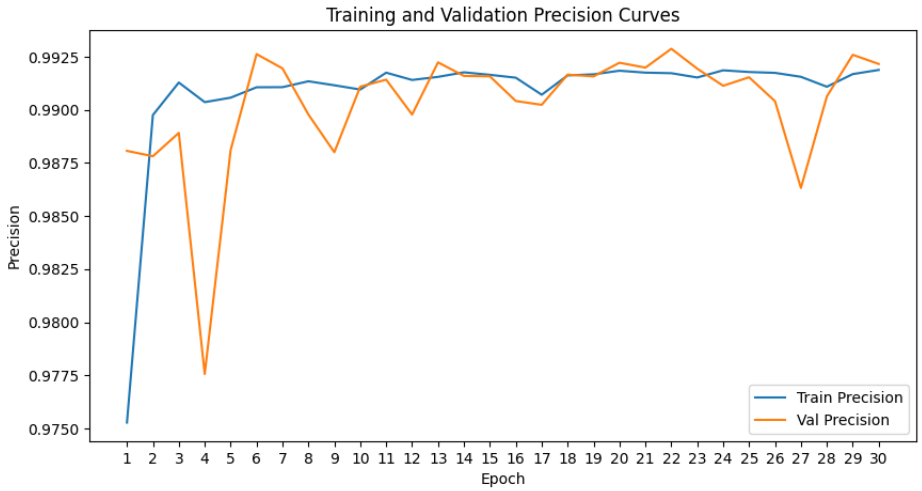


Figure E.11: Fold 3 precision curves.

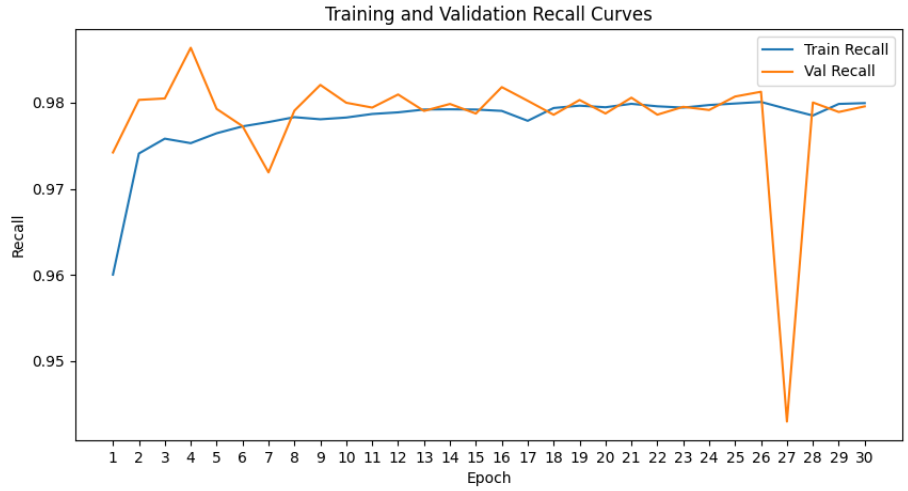


Figure E.12: Fold 3 recall curves.

Fold 3 training results on augmented data:

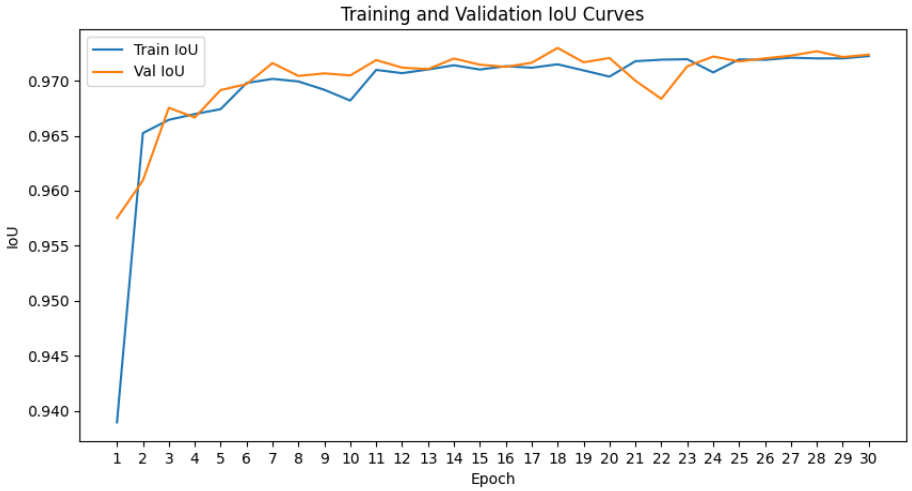


Figure E.13: Fold 3 IoU curves on augmented data.

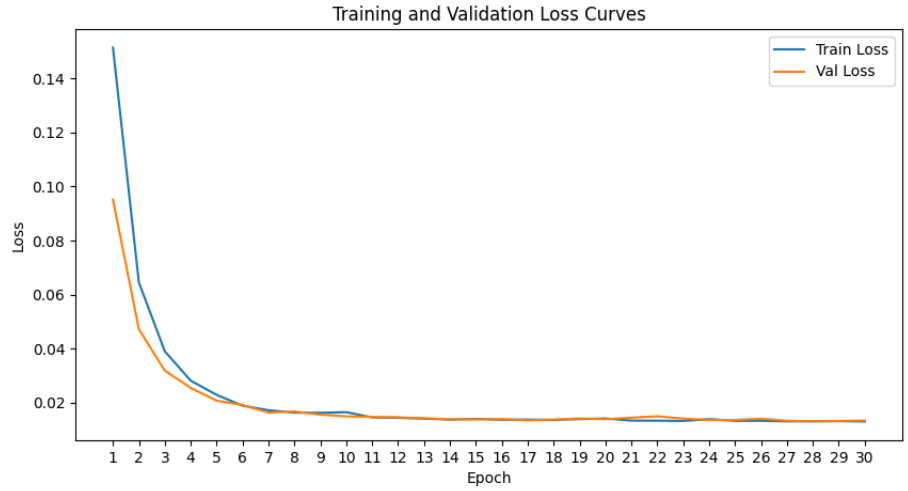


Figure E.14: Fold 3 loss curves on augmented data.

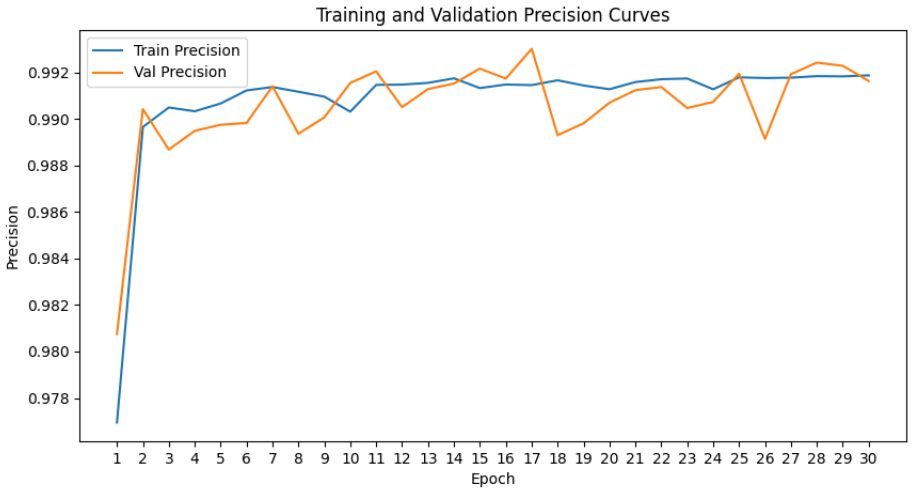


Figure E.15: Fold 3 precision curves on augmented data.

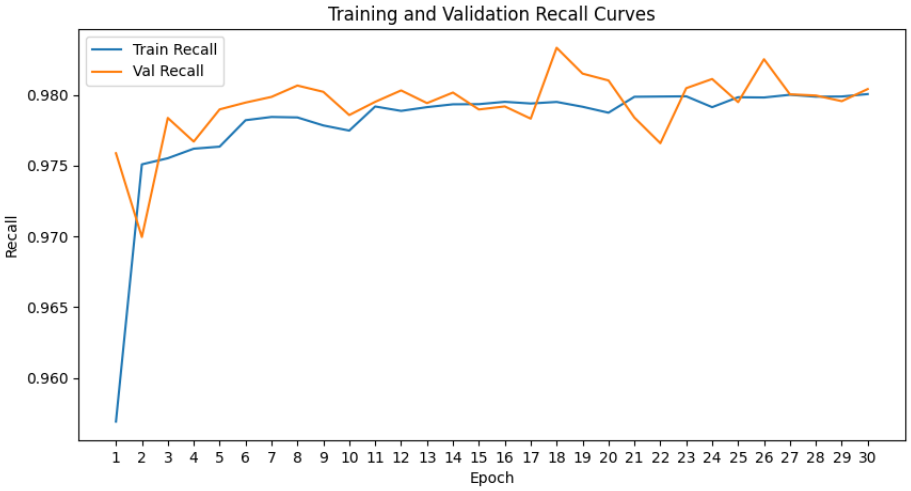


Figure E.16: Fold 3 recall curves on augmented data.

E.2 YOLOv8 Training

Fold 1 training results on augmented data on 30 epochs:

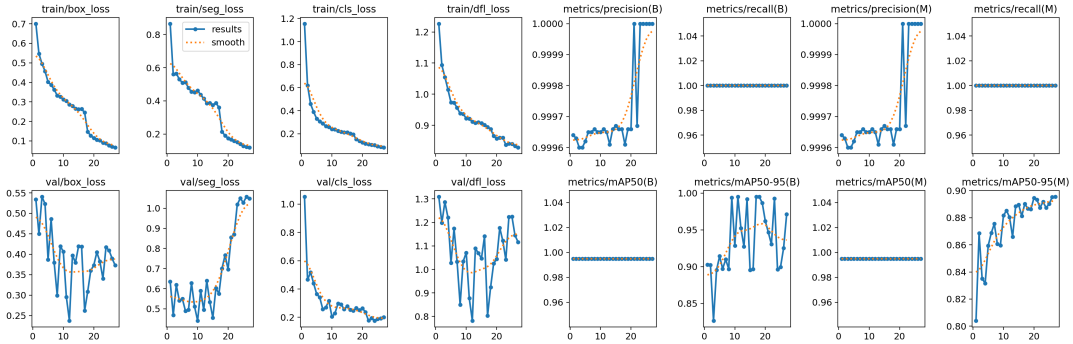


Figure E.17: Fold 1 curves on augmented data.

Fold 2 training results on augmented data:

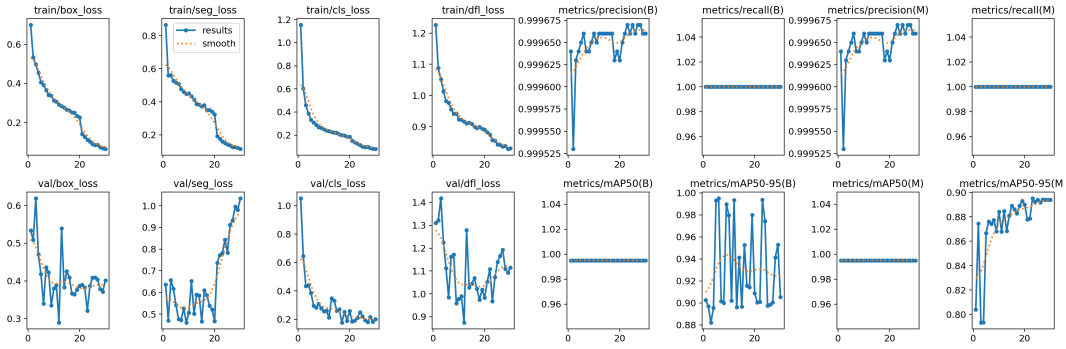


Figure E.18: Fold 2 curves on augmented data.

Fold 3 training results on augmented data on 70 epochs:

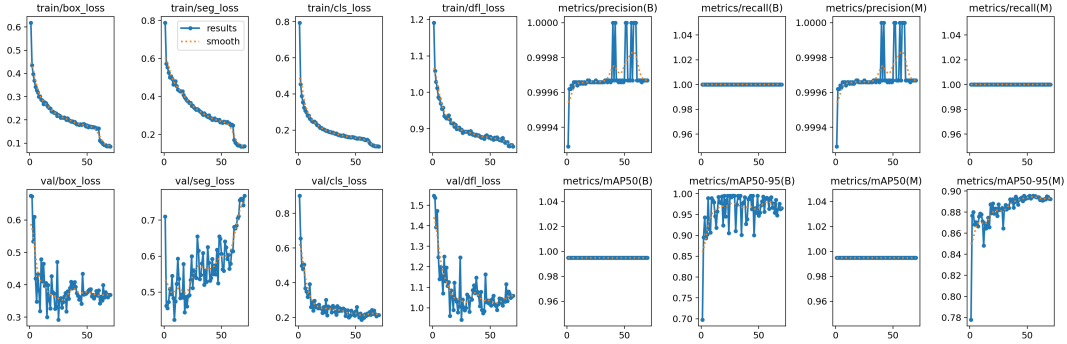


Figure E.19: Fold 3 curves on augmented data.

E.3 U-Net and Yolo Test Results

Fold 3 test results folder for U-Net with data augmentation on 30 epochs:

Figure E.20: Fold 3 testing result.

Fold 3 test results folder for YOLOv8 with data augmentation for 100 epochs:

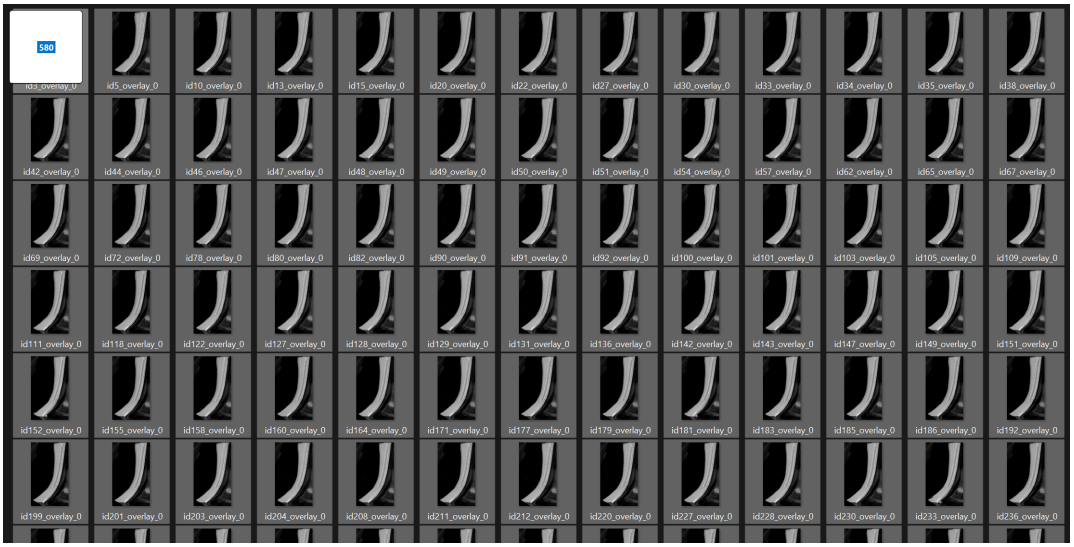


Figure E.21: Fold 3 testing result.

Appendix F

Appendix F - Matlab Correlation Script

```
% Load the first dataset
data1 = readtable('MasterData1.txt');
partNumbers1 = data1{:, 1}; % First column is PartNumber
measurements1 = data1{:, 3}; % Third column is the valuable Measurement

% Load the second dataset
fileID = fopen('width_measurements.txt', 'r');
data2 = textscan(fileID, 'id%d: [%d, %d, %d, %d]', 'Delimiter', '\n');
fclose(fileID);

partNumbers2 = data2{1};
measurements2_1 = data2{2};
measurements2_2 = data2{3};
measurements2_3 = data2{4};
measurements2_4 = data2{5};

% Initialize correlation coefficients
corrCoeff = zeros(1, 4);

% Find matching part numbers
[commonParts, idx1, idx2] = intersect(partNumbers1, partNumbers2);

% Extract corresponding measurements and convert to double
commonMeasurements1 = double(measurements1(idx1));
commonMeasurements2_1 = double(measurements2_1(idx2));
commonMeasurements2_2 = double(measurements2_2(idx2));
commonMeasurements2_3 = double(measurements2_3(idx2));
commonMeasurements2_4 = double(measurements2_4(idx2));

% Calculate correlation coefficients
corrCoeff(1) = corr(commonMeasurements1, commonMeasurements2_1);
corrCoeff(2) = corr(commonMeasurements1, commonMeasurements2_2);
corrCoeff(3) = corr(commonMeasurements1, commonMeasurements2_3);
corrCoeff(4) = corr(commonMeasurements1, commonMeasurements2_4);

% Display results
fprintf('Correlation with Measurement1: %.4f\n', corrCoeff(1));
fprintf('Correlation with Measurement2: %.4f\n', corrCoeff(2));
```

Figure F.1: MatLab script used for correlation calculation.

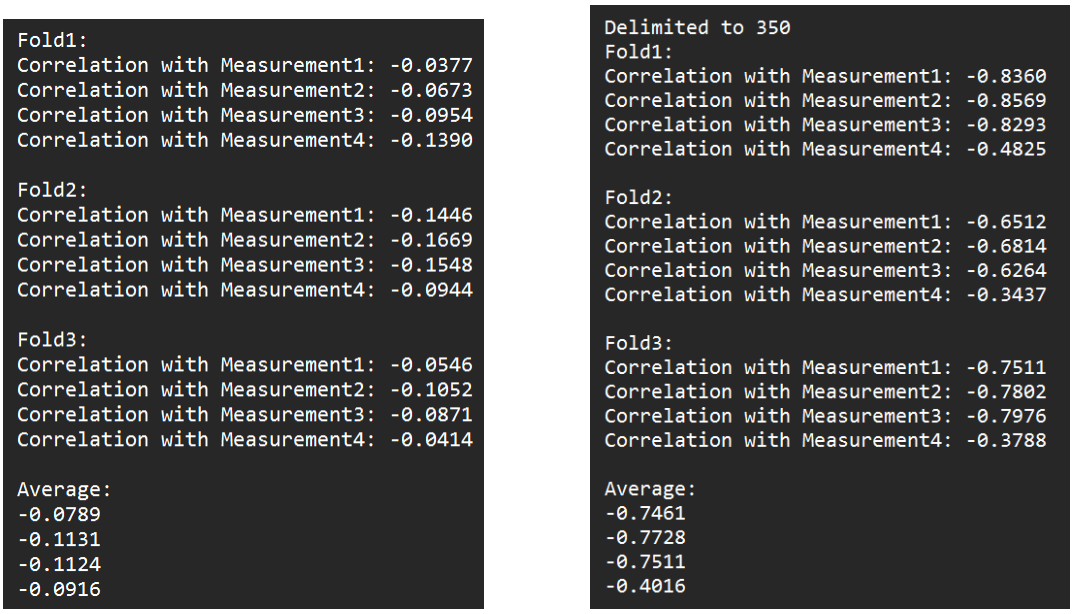


Figure F.2: Correlation calculations for U-Net results.