

.....

**Evaluate feasibility of using
Convolutional Neural Networks
to determine cause of damage of tempered glass**

.....

**Cleon Tay Wei De
(A0234749U)**

**Department of Mechanical Engineering
National University of Singapore
Session 2024/2025**

Summary

This study investigates the feasibility of using Convolutional Neural Networks (CNNs) to predict projectile impact parameters, specifically velocity and angle, based on the fracture patterns of tempered glass. The motivation for this research stems from the need to develop objective, automated methods for forensic analysis of glass fractures, which traditionally relies heavily on human expertise. Addressing a gap in current forensic techniques, this work explores whether CNN models can effectively extract meaningful features from fractured tempered glass images to infer critical impact parameters.

An experimental setup was designed to conduct controlled impact tests on tempered glass panels, varying the impact angle and velocity. A high-speed camera was used to accurately measure pre-impact velocities, and images of the resulting fracture patterns were captured under standardized lighting conditions. A total of 57 impact events were recorded, generating a dataset used for training and validating CNN models.

The glass fracture images were processed into two sets: non-cropped and cropped versions. The associated labels — impact angle and velocity — were normalized using min-max scaling. Various CNN models, based on the ResNeXt-101 architecture, were trained using different hyperparameter settings, resolutions, and data augmentation techniques. Model performance was evaluated primarily through Mean Absolute Error (MAE) and Mean Squared Error (MSE) loss metrics, with a 3-fold cross-validation approach employed to enhance robustness.

The findings indicate that while CNNs were able to abstract general features from the crack patterns, they were unable to predict the impact parameters with high accuracy. Velocity predictions were relatively more consistent than angle predictions, likely due

to a broader range of velocity data compared to the limited variation in angles.

Increasing image resolution beyond the default setting did not yield noticeable improvements. Data augmentation through horizontal flipping slightly improved angle prediction but also introduced mild overfitting effects. Overall, the models could only loosely identify impact characteristics rather than predict them precisely.

This study concludes that predicting tempered glass impact parameters using CNNs is currently not feasible with the available dataset and experimental conditions. However, it establishes an initial reference framework for future research efforts.

For future work, it is recommended to expand the dataset with greater variation in impact angles and to explore prediction tasks involving different types of glass, such as float or laminated glass. Alternative target parameters, such as impact energy, could also be considered. Furthermore, experimentation with different optimizers and a systematic evaluation of optimal image resolutions may help refine CNN model performance for glass fracture analysis.

Acknowledgement

I would like to express my sincere gratitude to all those who made this thesis possible.

I am deeply thankful to my family and close friends for their unwavering support throughout my research journey.

First and foremost, I would like to express my heartfelt appreciation to my supervisor, Associate Professor Dr Vincent Tan, for his invaluable guidance and support.

His insights and advice provided clear direction and motivation throughout the project.

I would also like to extend my gratitude to my industrial supervisor, Dr Tan Long Bin from A*STAR, for his valuable advice and knowledge-sharing, particularly in the field of machine learning, which was instrumental in improving various aspects of this research.

In addition, I am extremely thankful to Lab Technician Joe Low for his assistance with this project.

He provided important ideas for the design of the experimental setup and offered hands-on support during the setup process.

Finally, I would like to thank Research Engineer Lim Yun Mei for her advice, input, and assistance throughout the entire project.

Table of Contents

1	Introduction	7
1.1	Motivation	7
1.2	Conventional type of glass forensic analysis present	7
2	Objective and Scope.....	8
3	Literature Review	10
3.1	Types of Glasses	10
3.2	Tempered glass fracture mechanics	10
3.3	Impact resistance of tempered glasses.....	11
3.4	CNN and its applications.....	12
4	Experiment methodology	14
4.1	Test Overview.....	14
4.2	Apparatus Description	15
4.2.1	Tempered glass panels.....	15
4.2.2	Impactor	15
4.2.3	Fixture	16
4.3	Experimental set up	16
4.3.1	Set up of high-speed camera:	17
4.4	Procedure	18
5	Machine learning Methodology	20
5.1	Preparation of dataset for input into CNN model.....	20
5.1.1	Data format required to be loaded into CNN model	20
5.1.2	Min-max normalisation of labels	20
5.1.3	Image preparation:	21
5.2	Human learning of glass crack patterns.....	23
5.3	CNN model architecture.....	26
5.3.1	ResNext-101:	27
5.3.2	CNN model functions and hyper parameters	28
	<i>CNN models' parameter specifications</i>	30
6	Results and Discussion.....	32
6.1	Results and analysis of R-060-1x-mid and R-060-1x-mid-c.....	33
6.2	Effect of increasing resolution on MAE validation loss	36
6.3	Effect of increasing epochs number on MAE validation loss	38
6.4	Reduction of learning rate on model's stability.....	39
6.5	Data augmentation to increase range of angle.....	41

6.6 Investigation of prediction accuracy on individual data points.....	45
7 Conclusions	46
8 Recommendations for further work	47
9 References	48
10 Appendix	49

1 Introduction

1.1 Motivation

The use of unconventional projectiles, such as ball bearings launched by slingshots, presents significant challenges for law enforcement. These projectiles, often aimed at glass windows, leave fracture patterns that can provide valuable forensic evidence. To address this, the Ministry of Home Affairs (MHA) has requested research into analysing these patterns to infer critical projectile parameters, such as velocity and angle of impact, aiding in approximating the shooter's location and reconstructing incidents.

Traditional forensic methods for glass fracture analysis rely heavily on human expertise and pattern recognition, which can be labour-intensive and subjective. Recent advancements in machine learning, particularly Convolutional Neural Networks (CNNs), offer a promising solution. CNNs excel at identifying spatial patterns in images, making them suitable for analysing glass fracture images to extract projectile-related information.

Hence, this study serves to explore the feasibility of using CNNs for forensic glass analysis.

1.2 Conventional type of glass forensic analysis present

Most established glass forensic analysis methods focus on determining whether two glass samples share the same origin (Trejos, 2022). This is achieved through the characterization of distinguishing class characteristics, such as refractive index and elemental composition. Notable techniques include inductively coupled plasma mass spectrometry (ICP-MS) and micro x-ray fluorescence. While these methods are robust

for material identification, they are not designed to analyse the mechanical behaviour of glass during fracturing. In contrast, forensic analysis involving glass fracture fits and patterns remains experimental and is an area of ongoing research. A study by Hashley, titled “Evidential Significance of Multiple Fracture Patterns on Glass in Forensic Ballistics,” investigates fracture patterns on soda-lime glass of varying thicknesses impacted by projectiles fired at high speeds. The study recorded fracture characteristics such as radial fracture count, concentric fracture count, bullet hole diameter, and the thickness and diameter of the ‘mist zone.’

The results demonstrate that fracture patterns produced under identical parameters are consistent, offering potential forensic value. This consistency suggests that fracture patterns can serve as reliable indicators for distinguishing projectile types in forensic investigations.

However, these studies did not establish a relationship between the projectile’s impact parameters (such as speed, angle, or energy) and the resulting glass crack patterns. This highlights a significant gap in existing research, emphasizing the need to explore how impact dynamics influence fracture behaviour to provide predictive insights for forensic applications.

2 Objective and Scope

Based on the author’s knowledge, there is currently no research published on the usage of machine learning models to predict any glass impact parameters. Hence, the objective of this research is to investigate the feasibility of using Convolutional Neural Networks (CNNs) to determine projectile velocity and angle of impact on tempered glass based on its fracture patterns. This involves conducting controlled impact tests to create a dataset of crack patterns under varying parameters. The dataset will then be used to train

a CNN model to predict impact parameters from new fracture pattern inputs. This research aims to serve as a reference for subsequent researches on this topic.

The scope of this research includes:

- 1) Designing experimental setup for the glass impact test
- 2) Conducting controlled drop tests of a metal ball onto tempered glass panels at varying velocity and angles and collect images of the corresponding fracture patterns.
- 3) Processing of crack patterns images and preparation of dataset
- 4) Training, testing and validation of CNN models using the dataset and analyse the prediction accuracies.
- 5) Comment on the feasibility of using CNN to predict tempered glass impact parameters.

3 Literature Review

This chapter reviews relevant theories and findings on glass fracture mechanics and Convolutional Neural Networks (CNNs). These insights inform the planning and execution of this research.

3.1 Types of Glasses

Glass is defined as ‘an inorganic product of fusion which has cooled to a rigid condition without crystallizing’ (ASTMC162-05, 2015). Among the various types of glass, soda-lime glass is the most used due to its affordability and versatility. However, it is inherently brittle with low tensile strength (Ballarini et al., 2016), making it prone to breakage under stress or impact.

To improve its strength, methods such as annealing and thermal tempering are employed (Ernsberger, 1980). Thermal tempering involves heating the glass to approximately 600°C, followed by rapid cooling with air jets. This process induces a residual stress profile: the surface is compressed while the core is in tension. The surface compression enhances crack resistance, while the core tension stores energy that drives fragmentation upon fracture. Tempered glass is up to five times stronger than non-tempered glass, making it suitable for applications such as vehicle windows, building facades, and safety glass.

3.2 Tempered glass fracture mechanics

Tempered glass can still fracture if the impact energy breaches the compressive surface layer. Once this occurs, the tensile forces in the core drive rapid crack propagation, leading to extensive branching. The residual stress profile governs crack initiation and

fragmentation patterns, with the compressive surface delaying propagation and the tensile core driving it, resulting in consistent radial and concentric crack patterns.

A study by Stewart and Prakash (2013) titled “Modeling Dynamic Fragmentation of Tempered Glass” explores this process in depth. The concept of “frangibility”—the minimum stored energy required for fragmentation—is introduced. The study isolates the effects of stored tensile energy using methods like delayed fracture, revealing that frangibility is governed by the residual stress profile, characterized by:

Maximum central tension (σ_c): the peak stress in the core that dictates crack propagation.

Tensile stored energy (W^t): quantifies the energy available for fragmentation

Next, Gulati found that the glass fragment density is dependent on the maximum central tension magnitude for tempered glasses, expressed in the equation below:

$$N_c = \frac{f^2}{4L^2} \left(\frac{W^t}{G_{IC}} \right)^2 = f^2 \left(\frac{0.308(1-\nu)\sigma_c^2}{2EG_{IC}} \right)^2.$$

where N_c = Counted fragment density, L = thickness of glass plate, G_{IC} = critical fracture energy release rate, ν = poison's ratio.

This equation was validated through experimental data, finite element modelling, and simulations. While intrinsic fragmentation is primarily stress-driven, the study also notes that variations in external impact energy influence crack propagation, fragment density, and overall fracture dynamics. These findings provide a foundation for investigating how external energy affects tempered glass fracture patterns.

3.3 Impact resistance of tempered glasses

Gürsel and Özgül's study examines factors influencing tempered glass impact resistance. The study employed a low-speed, high-mass impact test using a drop-weight mechanism and a hardened steel ball. Glass samples, both tempered and un-tempered, with thicknesses of 4 mm and 6 mm were tested under varying ambient temperatures (-20°C, 20°C, 40°C). Key findings include:

- Higher ambient temperatures reduced the energy required to fracture the glass, while sub-zero temperatures increased it.
- Tempered glass required more energy to fracture than un-tempered glass, with thicker glass exhibiting greater resistance.

When comparing tempered and un-tempered glass, the impact energy required to fracture 4 mm and 6 mm un-tempered glass ranged from 1.8 to 3.07 J and 2.26 to 9.98 J, respectively. In contrast, the impact energy required to fracture 4 mm and 6 mm tempered glass at 20°C ranged from 2.84 to 7.51 J and 6.71 to 11.93 J, respectively. These findings underscore the enhanced impact resistance of tempered glass and the influence of glass thickness on its performance.

These results provide reference ranges for impact energy and highlight the importance of controlling ambient temperature in experiments to ensure consistent glass properties.

3.4 CNN and its applications

Neural networks (NNs) are supervised learning algorithms used to identify patterns in labeled data. CNNs, a specialized type of NN, are particularly effective for image-based tasks like classification and regression due to their ability to extract spatial features. CNNs use convolutional layers with learned kernels to perform operations over regions of input images, producing feature maps. Pooling layers reduce dimensionality while

retaining critical features, and fully connected layers aggregate these features for predictions.

CNNs require input-output pairs for training, where the input (X) typically represents images, and the output (Y) consists of labels. CNNs do not require explicit assumptions about spatial relationships, as they learn these relationships during training, making them adaptable to diverse datasets.

Applications

CNNs have shown promise in forensic analysis. For example, EfficientNetB4 has been used for bone age estimation using X-ray images, achieving a Mean Squared Error of 7.93 and an R-squared score of 0.88 (Remya et al., 2023). ForensicNet, a custom CNN, has demonstrated high accuracy in detecting image forgeries, identifying inconsistencies such as splicing and copy-move alterations (Tyagi & Yadav, 2023). These applications highlight the versatility of CNNs in automating and enhancing forensic processes, making them a suitable candidate for analyzing glass fracture patterns in this study.

4 Experiment methodology

This section outlines the design of the experimental setup, the variables for the glass impact test, and the rationale behind them. The glass impact test is developed based on insights from the literature review to ensure the reliability of the collected results.

4.1 Test Overview

The objective of the following experiment is to collect a (X,Y) labelled dataset to be used for CNN modelling.

X (input feature): $X = (x_n)$ glass fracture pattern images.

Y (output feature): $Y = (y_n^1, y_n^2)$: labels consisting of angle and velocity

These labelled datasets will be processed further before input into the CNN models for training and validation, which will be detailed in later sections.

To collect the above labelled dataset, a glass impact test is conducted to obtain different glass fracture patterns from being dynamically impacted at different impact velocity and angles. The impact test involves a high-mass projectile traveling at a low velocity, following a similar approach to that described in Gürsel and Özgül's study. A drop-weight mechanism is employed, using an impactor with a spherical contact point dropped from various heights onto the tempered glass panel to vary the velocity right before impact. The glass panel is mounted on an adjustable fixture to achieve different angles. The mass of the impactor remains constant, while the impact angle and velocity are varied to produce distinct crack patterns for analysis. The bottom side of the tempered glass panels are covered with a layer of plastic film to prevent catastrophic shattering, preserving the integrity of fragment patterns to allow consistent analysis.

Detailed descriptions of the apparatus, setup, and procedure are provided in subsequent sections.

4.2 Apparatus Description

4.2.1 Tempered glass panels

The specifications of the tempered glass panels used are summarized in Table 4.1:

Table 4.1: Specifications of tempered glass panels used

Specification	Details
Dimensions	15cm x 10cm x 3mm
Base glass material	Soda-Lime glass
Tempering process	Thermal tempering: heated in furnace to 700°C for 145 seconds, fast quenching for 40 seconds
Industry standards	CCC mark, GB 15763.4-2009
Manufacturer	Manufactured by Qibin glass, processed by Huji Glass Co Ltd

All tempered glass panels have consistent thickness, manufacturing, and tempering processes to ensure uniform frangibility, isolating the effects of external impact energy on fracture patterns.

4.2.2 Impactor

The impactor consists of a 304 stainless steel ball (3 cm diameter) with an M8 thread, attached to a cylindrical weight and a screw eye hook, as shown in Figure 4.1. The total weight of the impactor is 275.5 g. A string is used to raise the impactor up to the desired height for controlled release.



Figure 4.1: Image of impactor used

4.2.3 Fixture

A custom-designed, tiltable fixture is constructed to hold the tempered glass panels at varying angles. A schematic of the fixture is presented in Figure 4.2.

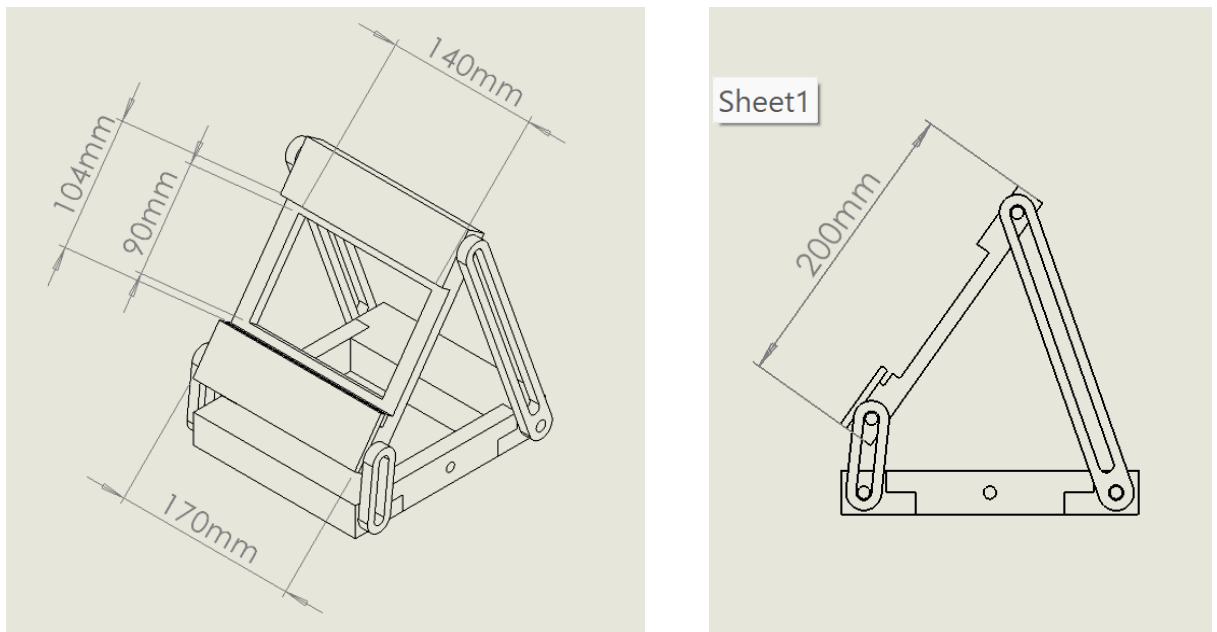


Figure 4.2: (a) Isometric view of the fixture; (b) Side view of the fixture

4.3 Experimental set up

The tempered glass is mounted on the fixture, and a 2m long pipe with an inner diameter of 3.1 cm is installed to ensure the impactor hit the centre of the glass precisely. A high-

speed camera is positioned to measure the velocity of the impactor immediately before impact. A schematic of the experimental set up is presented in Figure 4.3.

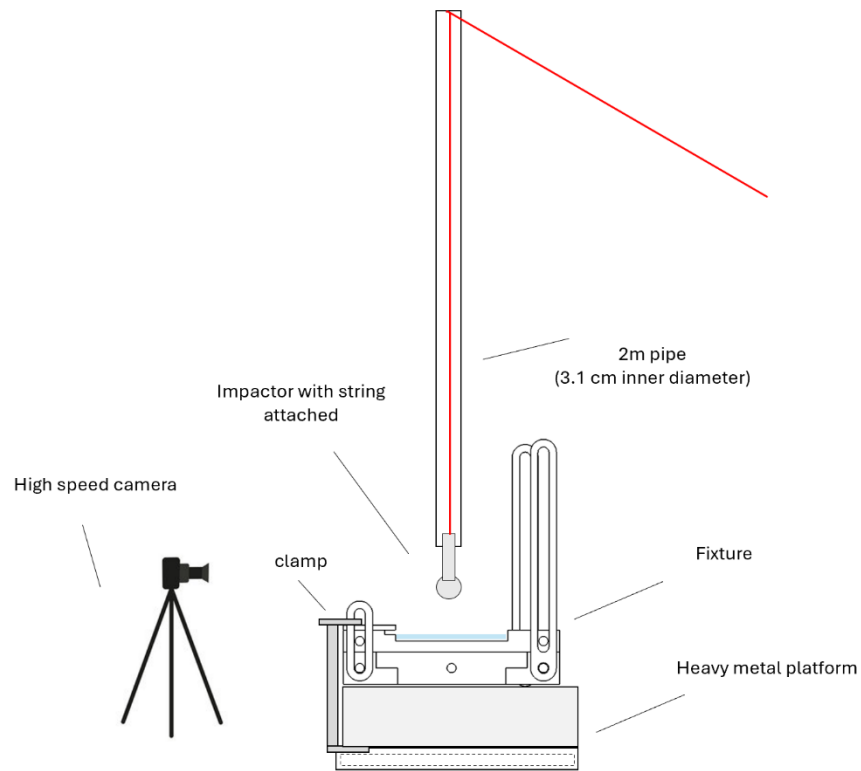


Figure 4.3: schematic of the experimental set up

4.3.1 Set up of high-speed camera:

- Velocity of impactor calculated by measuring the time interval for ball to leave the pipe and first contact with the glass panels
- Camera is tilted at an angle, requires calibration to ensure distance measured is right
- Steps to do so
 - Use a reference, between 2 markings on the pipe, mark out y axis pixels P_{r1} and P_{r2} from the camera, coinciding to the top and bottom markings of the pipe

- Also measure the absolute distance = d_r between the 2 markings using a vernier calliper
- Mark out y axis pixels P_{s1} , P_{s2} as recorded by camera, coinciding with the opening of the pipe, and the glass contact point respectively
- Actual distance between end of pipe and contact point of glass is calculated using:

$$d_{\text{pipe to contact point}} = d_r \cdot \frac{P_{r1} - P_{r2}}{P_{s1} - P_{s2}}$$

4.4 Procedure

The experiment is conducted at ambient temperature, with the tempered glass is angled at 0° , 7.5° , 17.5° , and 27.5° , according to the test parameters. The impactor is raised to heights ranging from 150 cm to 195 cm using the string attached. Upon fracture, the string connected to the impactor is pulled naught to prevent secondary impacts caused by the impactor rebound. To ensure that air resistance and friction between the impactor and pipe is accounted for, the velocity before impact is measured by the high-speed camera and recorded. This recorded velocity is used instead of measuring the theoretical free fall velocity at different free fall heights. A total of 57 glass impact tests are conducted. The parameters: impact angle and velocity from the glass impact tests detailed are used for analysis, as detailed in appendix.

Next, to capture images of the fracture patterns, the fractured glass is then placed on a white sheet (opaque plastic sheet or white paper) in an image-capturing setup, as shown in Figure 4.4. There is a light source beneath the sheet to eliminate shadows and obtain high-quality images. Each fractured glass sample is photographed twice—once on the opaque plastic sheet and once on the white paper.



Figure 4.4: Setup to capture image of glass fracture pattern

Figure 4.5 shows 2 images of the same glass crack pattern (dataset number 1), taken from the opaque plastic sheet and white paper respectively. The images taken with the plastic sheet can be seen to have a cleaner background as compared to the white paper. Hence, subsequent analysis will first be done using the images with plastic sheet backing.

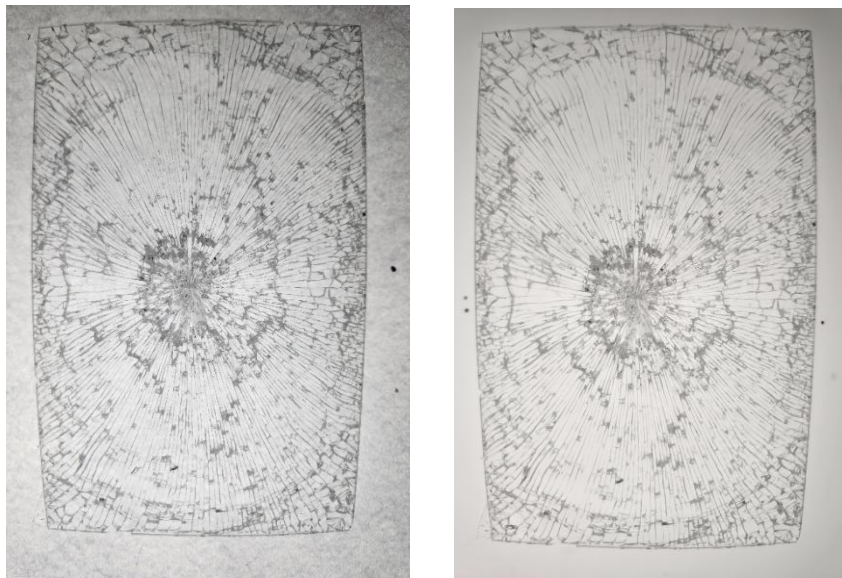


Figure4. 5 :(a) Image taken with paper backing; (b) Image taken with plastic sheet backing

5 Machine learning Methodology

This section outlines the methodology for preparing the dataset and designing the Convolutional Neural Network (CNN) model used for predictive analysis. The CNN model is tasked with performing a two-variable regression, aiming to predict the impact angle and velocity of a projectile based on glass fracture pattern images.

5.1 Preparation of dataset for input into CNN model

The raw data collected from the experimental study was first reformatted into a structure suitable for input into the CNN model.

5.1.1 Data format required to be loaded into CNN model

Each dataset entry comprises an image and its corresponding set of labels.

Image: input feature for the CNN model.

Each image represents the crack pattern of a fractured tempered glass panel, formatted as a 3D tensor of dimensions (C,H,W), where C represents the number of color channels (3 for RGB images), while H and W represents height and width of the image in pixels respectively.

The associated labels $Y=(y_1,y_2)$ correspond to the ground truth values for each image, where y_1 represents the impact angle and y_2 represents the impact velocity.

5.1.2 Min-max normalisation of labels

As there are different numerical ranges of the two target variables: angle values of (0°, 27.5°) and velocity of (4.51, 5.77) m/s, it is necessary to normalize the labels to a

common scale before model training. Hence, min-max normalization is employed to transform both labels into a standardized range between 0 and 1. This normalization ensures that both target variables are treated equally during training, preventing the larger-magnitude label from disproportionately influencing the loss function, and improves the model's convergence rate and overall training stability.

Min max conversion is given as follows:

$$\text{Normalized Angle } y_1 = \frac{y_{1,\text{raw}} - y_{1,\text{min}}}{y_{1,\text{max}} - y_{1,\text{min}}}$$

$$\text{Normalized Velocity } y_2 = \frac{y_{2,\text{raw}} - y_{2,\text{min}}}{y_{2,\text{max}} - y_{2,\text{min}}}$$

Where $y_{1,\text{raw}}$ and $y_{2,\text{raw}}$ are the original raw values and $y_{1,\text{min}}$, $y_{1,\text{max}}$, $y_{2,\text{min}}$, $y_{2,\text{max}}$ are the minimum and maximum values of the respective datasets.

5.1.3 Image preparation:

Each glass fracture image collected during the experiment was named and saved using the format PXX.jpg, where XX corresponds to the dataset index number.

Before being fed into the CNN model, all images underwent resizing and downscaling to a fixed resolution (p_x , p_y). The resizing helps to standardize the input aspect ratio across all samples, and to reduce the computational load during CNN training, as the original images were of high resolution and would require extensive computational resources.

Two distinct sets of images were prepared for model analysis:

1) Image Set 1 (Non-Cropped):

This set consists of non-cropped images retaining the original capture, resized

to a fixed aspect ratio of 16:9.

Image resolution: 336×448 pixels.

2) Image Set 2 (Cropped):

This set consists of images that were manually cropped to remove most of the background surrounding the fractured glass panel. The images were resized to match the physical aspect ratio of the tempered glass panels (3:2)

Image resolution: 336×504 pixels.

For simplicity and consistency in data management, the naming convention PXX.jpg was retained for both Image Set 1 and Image Set 2. However, the two sets were saved in separate directories to avoid confusion.

Each CNN model will be trained and analysed twice: once using the non-cropped images and once using the cropped images. This allows the investigation of whether removing the background improves the CNN model's predictive performance. Next, the pixel values of the images were normalized to the range (-1,1) by applying a channel-wise normalization using a mean and standard deviation of 0.5 across all RGB channels. This normalization improves the convergence speed and stability during model training.

For internal dataset management, two datasets were defined:

- 1) **S_{nc}**: dataset consisting of non-cropped images with normalized labels.
- 2) **S_c**: dataset consisting of cropped images with normalized labels.

A table containing the filenames and corresponding normalized label values for both datasets S_{nc} and S_c can be found in the Appendix.

5.2 Human learning of glass crack patterns

Before training and evaluating the CNN model, a preliminary human analysis of selected glass crack patterns was performed to better understand the physical characteristics associated with different impact parameters. This step provides intuition about what features the CNN model may learn during training. Three fracture patterns, corresponding to dataset indices P04, P09, and P26, were selected for visual inspection. The corresponding impact parameters are shown in table 5.1

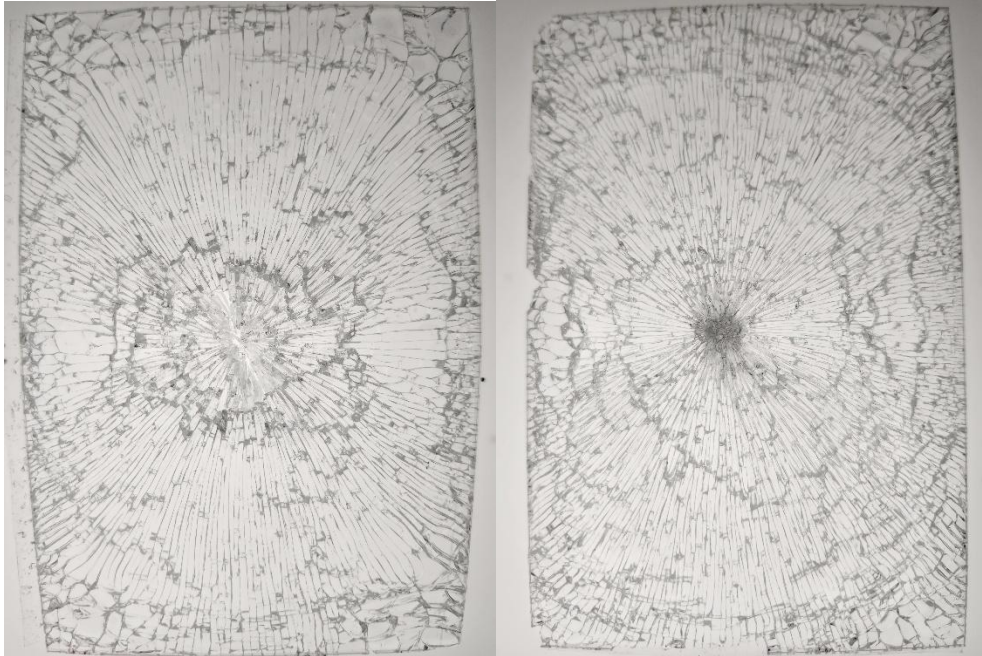
Table 5.1: impact parameters of P04, P09, P26

Image	P04	P09	P26
Angle	0	0	27.5
Velocity	5.21	5.77	5.15

Several key fracture characteristics were examined, including the fracture zone area and size, crack density, and the distribution of radial and concentric cracks

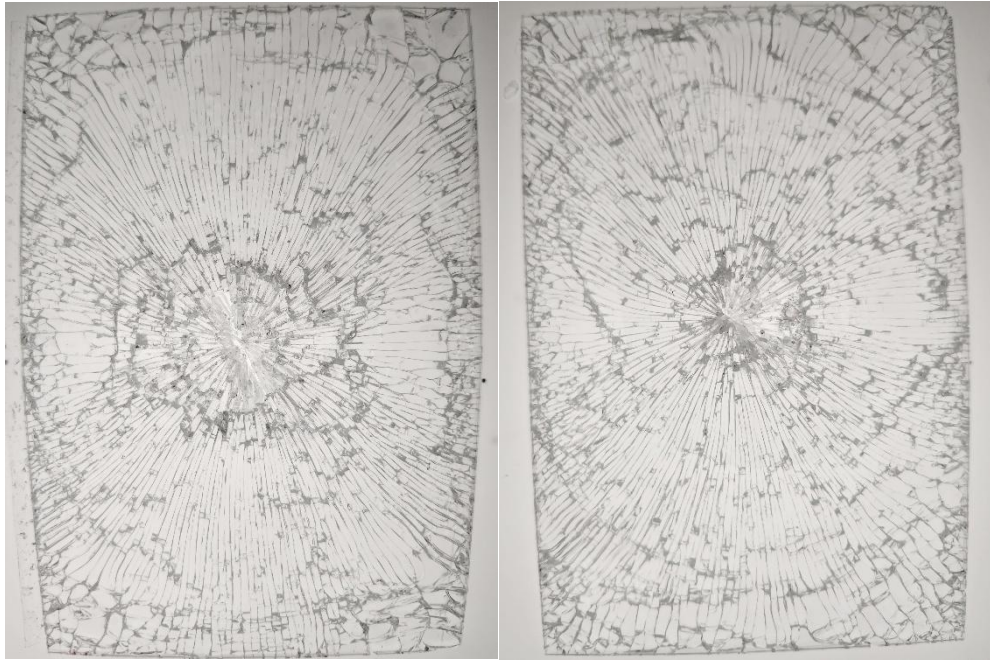
Comparison between P04 (0, 5.21) vs P09 (0,5.77)

Both P04 and P09 were impacted at an angle of 0° , but with different velocities (5.21 m/s for P04 and 5.77 m/s for P09).



Both patterns displayed relatively uniform radial and concentric cracks; however, P09 exhibited a higher crack density, with more frequent and condensed concentric rings and a denser mist zone. In contrast, P04 had fewer concentric cracks, longer uninterrupted radial cracks, and a less prominent mist zone. These observations suggest that higher velocity increases crack density and fragmentation near the impact point.

Comparison between P04 (0, 5.21) vs P26 (27.5, 5.15), inclined on the right side



A comparison between P04 and P26 isolates the effect of impact angle. P26, impacted at 27.5° , shows asymmetry in its crack pattern, with radial cracks longer and denser on the right side of the image, corresponding to the tilt direction. The concentric cracks in P26 are also stretched asymmetrically, in contrast to the symmetrical rings in P04.

This suggests that inclination during impact leads to directional bias in crack propagation.

Overall, preliminary observations indicate that both impact velocity and angle produce distinguishable differences in fracture patterns, supporting the feasibility of using CNNs to predict these parameters based on image analysis.

5.3 CNN model architecture

This subsection provides a background context and relevant information on the CNN models used, such as model architecture, functions and hyper parameters. It also details the parameters for the CNN models trained for this research.

Supervised machine learning models aim to learn a mapping from input features to output labels based on a provided dataset. In this study, a regression task is performed: given a glass fracture pattern image as input, the CNN model is trained to predict two output parameters — the impact angle and velocity.

CNN modeling follows a typical supervised learning framework. The model contains learnable parameters, referred to as weights w , which are optimized during training. For each epoch, the training dataset is passed through the model to compute a training loss, typically using a loss function such as Mean Squared Error (MSE).

Backpropagation is then used to compute gradients of the loss with respect to the model weights, and these gradients are then used to update the weights. During training, the validation dataset is also evaluated after each epoch to monitor validation loss. If the validation loss reaches a minimum, the current state of the model is saved.

This training flow is illustrated in the schematic diagram provided in Figure 5.1.

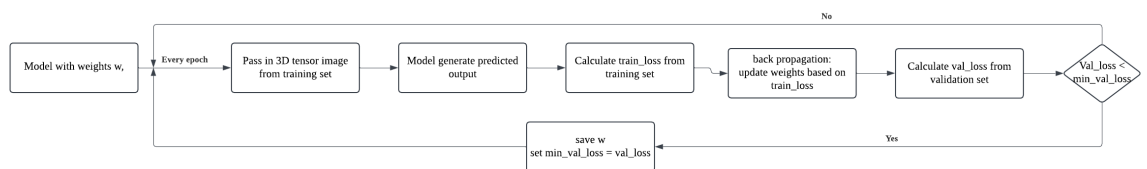


Figure 5.1: Schematics of CNN training process

5.3.1 ResNext-101:

For this research, the ResNext-101 CNN model was selected as the primary architecture. It is a high-performing CNN model known for its strong feature extraction capabilities. Strong CNN architectures refers to models that have demonstrated consistently high performance across a wide range of image-based tasks, typically in large-scale benchmarks such as ImageNet. Implementations of many popular CNN models, including ResNeXt-101, are available through the PyTorch Vision GitHub repository, with their performances well-documented based on standardized tasks such as object classification.

ResNext-101 is a variant of the ResNet architecture, incorporating a grouped convolutional design known as "cardinality," which refers to the number of independent paths within each block. By increasing cardinality rather than just depth or width, ResNext achieves better accuracy without significantly increasing computational cost. This study uses ResNeXt-101-32x8d, which has cardinality = 32, a depth= 101 layers, and a width multiplier = 8

Figure 5.2 illustrates the architecture of a ResNeXt block as implemented by NVIDIA (NVIDIA, n.d.)

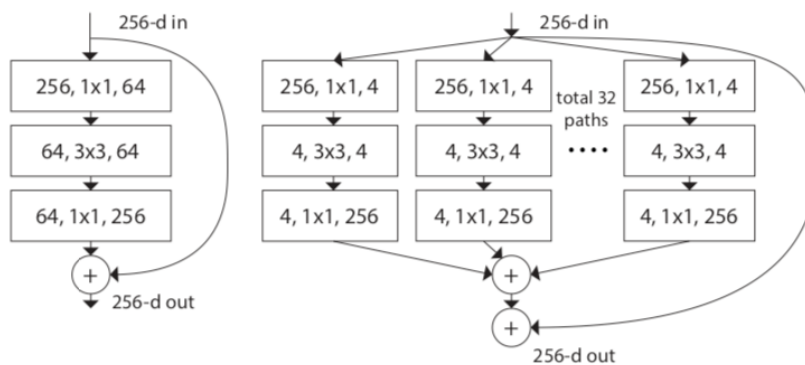


Figure 5.2: Resnext block architecture

The choice of ResNeXt-101 for this study is motivated by its proven feasibility in related applications, such as the analysis of composite thermoforming images to predict multi-variable parameters, which involves similar task requirements of feature extraction and regression (Long Bin Tan, 2022).

5.3.2 CNN model functions and hyper parameters

The CNN model training process relies on optimization functions, loss functions, and hyperparameter tuning to achieve effective learning.

Optimization function responsible for adjusting the model's internal weights based on the computed loss. In this study, the Adam optimizer is used due to its ability to adapt learning rates for each parameter, combining the advantages of momentum and RMSprop methods. Adam typically requires less hyperparameter tuning and converges faster compared to traditional stochastic gradient descent, making it suitable for feasibility studies such as this research where rapid initial exploration is prioritized.

The loss function measures the accuracy of the model's predictions by quantifying the difference between the predicted outputs and the ground truth labels.

Two types of loss functions are employed: Mean Absolute Error (MAE) and Mean Squared Error (MSE).

The Mean Absolute Error (MAE) is defined as:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$$

The Mean Squared Error (MSE) is defined as:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

MSE emphasizes larger errors by squaring the differences, thus penalizing outliers more heavily. In contrast, MAE treats all errors equally regardless of magnitude. During training, MSE is used to monitor and compare the training and validation losses across epochs. For final model evaluation, MAE is used as it provides a more straightforward comparison between predicted and actual impact parameters.

***Hyperparameters** govern the training dynamics of the CNN model. The key hyperparameters considered in this study include learning rate, batch size, number of epochs, and parameters related to learning rate scheduling.*

Table 5.2 summarizes the hyperparameters and their selected values for the first model iteration.

Table 5.2: CNN learning rate hyperparameters description

Hyperparameter	Description	Value
Learning Rate	Step size for weight updates	1×10^{-5}
Batch Size	Number of samples per training batch	16
Epochs	Number of complete passes through the training dataset	60
Factor	Learning rate decay factor upon plateau	0.5
Patience	Number of epochs to wait before reducing learning rate	5
Threshold	Minimum change in validation loss to qualify as improvement	1×10^{-4}
Minimum Learning Rate	Lower bound for the learning rate during training	1×10^{-6}

K-Fold

Given the small dataset size, standard fixed train-validation splits would risk overfitting or failing to represent the data distribution adequately. Therefore, a 3-fold cross-validation approach is employed. K-fold cross-validation involves splitting the dataset S into k equal subsets. In this study, the 57 data points are divided into three subsets S_{3k1} , S_{3k2} and S_{3k3} , each containing 19 samples.

The folds are organized as follows:

1st fold: Training set = $S_{3k1} \cup S_{3k2}$, Validation set = S_{3k3}

2nd fold: Training set = $S_{3k2} \cup S_{3k3}$, Validation set = S_{3k1}

3rd fold: Training set = $S_{3k1} \cup S_{3k3}$, Validation set = S_{3k2}

Each CNN model configuration is trained three times, once for each fold, resulting in three models representing different coverage of the data space. The final model performance is evaluated by taking the average MAE validation loss across the three folds. This approach ensures that each sample is used for both training and validation, improving robustness in the evaluation of the CNN model's predictive capabilities.

Training and Validation loss

During training, two types of loss values are monitored to evaluate the model's learning progress. The training loss is computed using the Mean Squared Error (MSE) between the model predictions $f_{\theta_n}(x)$ and the true labels within the training set S_{training} . Next, the validation loss is computed using the Mean Absolute Error (MAE) between the model predictions and the ground truth within the validation set $S_{\text{validation}}$. All labels have been min-max normalized to the range (0,1).

Next, the validation loss can also be computed with or without clamping the predicted values. Clamping refers to the limiting the predicted y_1 (angle) and y_2 (velocity) values within the range (0,1) to be consistent with the normalized label range. This helps to mitigate the effects of extreme outliers and exploding loss value.

CNN models' parameter specifications

To assess the feasibility of using CNN models to predict impact angle and velocity, a series of 3-fold cross-validation experiments were conducted on both the non-cropped

(S_{nc}) and cropped (S_c) datasets. The study follows an iterative model refinement approach. A base model f_0 was first trained under standard parameters. Subsequent models were progressively modified based on prior results to explore factors influencing predictive performance. An overview of all trained models f_{θ_n} and their parameter settings is presented in Table 5.3.

Table 5.3 CNN model parameters

f_{θ^n}	Model Name <i>R – epochs – pixels – learning rate – crop</i>	Epochs	Pixel dimension	Learning Parameters					Image dataset (S_c = cropped) (S_{nc} =non- cropped)
				Factor	Patience	Thresho ld	Learnin g rate (Lr)	Min Lr	
f_{θ^1}	R-060-1x-mid	60	336 x 448	0.5	5	1E-04	1E-05	1E-06	S_{nc}
f_{θ^2}	R-060-1x-mid-c	60	336 x 504	0.5	5	1E-04	1E-05	1E-06	S_c
f_{θ^3}	R-060-2x-mid	60	672 x 896	0.5	5	1E-04	1E-05	1E-06	S_{nc}
f_{θ^4}	R-060-2x-mid-c	60	672 x 1008	0.5	5	1E-04	1E-05	1E-06	S_c
f_{θ^5}	R-120-1x-mid	120	336 x 448	0.5	5	1E-04	1E-05	1E-06	S_{nc}
f_{θ^6}	R-120-1x-mid-c	120	336 x 504	0.5	5	1E-04	1E-05	1E-06	S_c
f_{θ^7}	R-120-1x-low	120	336 x 448	0.3	3	1E-03	1E-05	5E-08	S_{nc}
f_{θ^8}	R-120-1x-low-c	120	336 x 504	0.3	3	1E-03	1E-05	5E-08	S_c
f_{θ^9}	R-aug	120	336 x 448	0.3	3	1E-03	1E-05	5E-08	S_{nc} (augmented)
$f_{\theta^{10}}$	R-aug-c	120	336 x 504	0.3	3	1E-03	1E-05	5E-08	S_c (augmented)

Further discussion on model variations and dataset treatments will be provided in the Results and Discussion sections.

6 Results and Discussion

This section presents the results and analysis of the different CNN models developed through an iterative approach. The changes in training and validation MSE loss across epochs are used to evaluate each model's generalization capability and training stability. The validation MAE loss for angle, velocity, and the combined metrics is used to assess each model's predictive performance and the overall feasibility of using CNNs for impact parameter prediction. The models are analysed iteratively to explore the effects of different parameters and to identify the configuration that offers the best predictive performance, enabling a more informed conclusion on the feasibility of the approach.

6.1 Results and analysis of R-060-1x-mid and R-060-1x-mid-c

First, the initial models f_{θ}^1 : R-060-1x-mid and f_{θ}^2 : R-060-1x-mid-c are trained and analysed.

The models are trained with 3-fold, factor = 0.5, patience = 5, threshold = 1E-04, lr= 1E-05,

Min lr = 1E-06

Table 6.1 shows the MAE validation loss of angle, velocity and both combined for both models, with and without clamping.

Table 6.1: Validation MAE loss for R-060-1x-mid and R-060-1x-mid-c

R-060-1x-mid f_0^1 un-clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.537	0.139	0.338	0.275
Fold 2	0.261	0.200	0.230	
Fold 3	0.294	0.223	0.258	
clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.286	0.139	0.212	0.234
Fold 2	0.261	0.200	0.230	
Fold 3	0.294	0.223	0.258	

R-060-1x-mid-c f_0^2 un-clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.467	0.151	0.309	0.270
Fold 2	0.274	0.192	0.233	
Fold 3	0.338	0.200	0.269	
clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.286	0.151	0.218	0.233
Fold 2	0.274	0.192	0.233	
Fold 3	0.294	0.200	0.247	

Clamped vs unclamped

Comparing the clamped and unclamped results, it is observed that the MAE validation loss for angle is significantly higher in the unclamped case for the first fold of both models. This indicates the presence of outlier predictions that fall outside the normalized range for the angle label. In contrast, there is no significant difference between the clamped and unclamped MAE validation losses for velocity. For subsequent analyses, unless otherwise stated, all validation results are recorded with clamping applied.

Overall validation MAE loss

The models R-060-1x-mid and R-060-1x-mid-c exhibited similar validation MAE losses for both impact angle and velocity. Both models achieved an average overall

validation MAE loss of approximately 0.23. Since the labels are min-max normalized to the range (0,1) a MAE loss of 0.23 suggests that the models are able to detect general patterns in the glass crack images, but are not accurate enough to predict the exact impact parameters. To improve feature extraction, the number of input pixels was increased to provide higher image resolution during training. Across both models, the angle MAE loss ranged from approximately 0.26 to 0.29, while the velocity MAE loss ranged from approximately 0.13 to 0.20. This shows that prediction errors for velocity are consistently lower than for angle.

To address the higher angle prediction error, data augmentation through horizontal flipping was considered. By flipping the images, the angle range can be extended from $(0^\circ, 27.5^\circ)$ to $(-27.5^\circ, 27.5^\circ)$, potentially helping the model learn a wider range of angle variations. It is also observed that the dataset has greater inherent variation in velocity compared to angle, which may contribute to the lower prediction errors in velocity.

Training and validation MSE loss across epochs

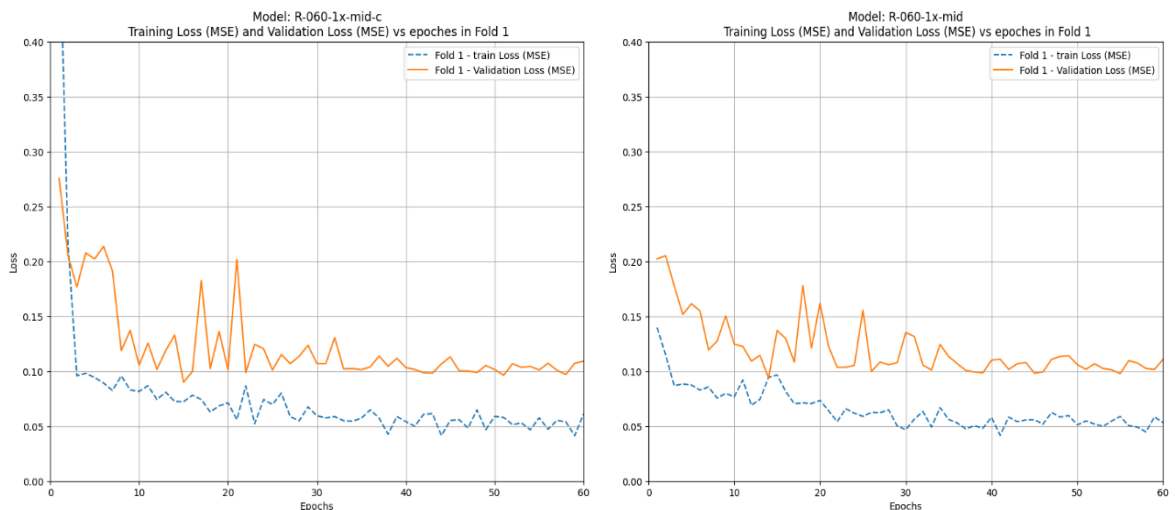


Figure 6.1: R-060-1x-mid and R-060-1x-mid-c fold 1 training and validation MSE loss across epochs

Figure 6.1 shows the training and validation MSE loss across epochs for the two models. The MSE loss fluctuates across epochs, and it is unclear whether convergence

has been achieved. To investigate further, the number of epochs can be increased to observe if convergence occurs. If not, adjustments to parameters related to the learning rate.

The subsequent models will explore the effects of the following changes to assess whether improvements can be achieved:

- 1) Increasing the input image resolution.
- 2) Increasing the number of training epochs and analysing whether a lower learning rate is necessary.
- 3) Applying data augmentation to expand the dataset.

6.2 Effect of increasing resolution on MAE validation loss

Next, the models R-060-2x-mid and R-060-2x-mid-c are trained using the same learning rate parameters but with twice the image resolution. The pixel dimensions used are 896×672 for R-060-2x-mid and 1008×672 for R-060-2x-mid-c. This higher resolution is obtained by downscaling the original high-definition (HD) images.

Table 6.2 shows the validation MAE loss of angle, velocity and both combined for both models with clamping.

Table 6.2 Validation MAE loss for R-060-2x-mid and R-060-2x-mid-c with clamping

R-060-2x-mid clamped					R-060-2x-mid-c clamped				
Fold	MAE				Fold	MAE			
	angle	velocity	overall	Avg overall		angle	velocity	overall	Avg overall
Fold 1	0.285	0.167	0.226	0.236	Fold 1	0.308	0.151	0.230	0.240
Fold 2	0.265	0.190	0.228		Fold 2	0.280	0.181	0.230	
Fold 3	0.311	0.197	0.254		Fold 3	0.323	0.200	0.261	

Overall MAE validation loss: R-060-2x-mid vs R-060-1x-mid

Comparing the normal and 2x pixel resolution models, the average overall MAE loss across all folds for R-060-2x-mid and R-060-2x-mid-c is slightly higher than for R-060-1x-mid and R-060-1x-mid-c. This suggests that increasing the resolution beyond the default dimensions ,(336 x 448) and (336 x 504) does not significantly improve feature extraction or model performance. Hence, it is sufficient to continue further analysis using the default resolution

Next, a preliminary visual comparison between the standard resolution images and a halved resolution version (168 x 224) shows that important fracture features, such as

radial cracks, become less distinguishable at lower resolutions. Future work could focus on training and evaluating models across a range of resolutions to determine the optimal image clarity required for reliable feature extraction.

6.3 Effect of increasing epochs number on MAE validation loss

The models R-120-1x-mid and R-120-1x-mid-c are trained over 120 epochs, as compared to the base models which were trained for 60 epochs, to observe whether the training and validation losses would converge with a higher number of epochs.

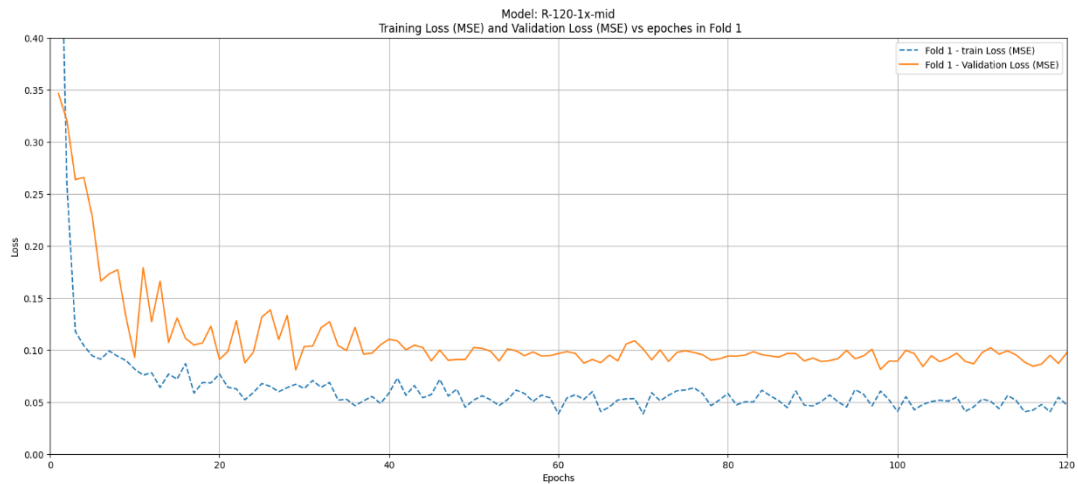


Figure 6.2: R-120-1x-mid fold 1 training and validation MSE loss across epochs

The validation MSE loss does not continue to decrease after approximately 40 epochs and instead fluctuates, indicating that the model has not fully stabilized. In contrast, the training MSE loss continues to decrease slowly and shows signs of stabilization. These observations suggest that the model has not fully converged.

Hence, reducing the learning rate slightly may help improve convergence of the model

6.4 Reduction of learning rate on model's stability

The models R-120-1x-low and R-120-1x-low-c are trained with the adjusted parameters as shown below.

Table 6.3: Changes in learning rate and rationale

Parameter	Current (R-120-1x-mid/ R-120-1x-mid-c)	New (R-120-1x-low/ R-120-1x-low-c)	Reason
Lr	1E-5	1E-5	Same starting LR to retain initial convergence
Factor	0.5	0.3	Reduce LR faster
Patience	5	3	Reduce waiting before LR decay
Threshold	1E-4	1E-3	More lenient on detecting improvements
Min Lr	1E-6	5E-8	Allow further LR reduction

Training and validation MSE loss across epochs

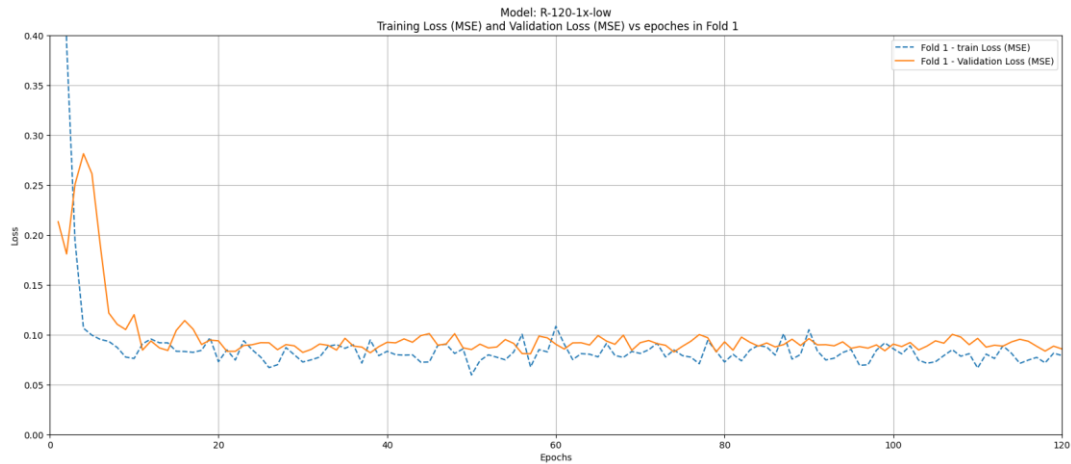


Figure 6.3: R-120-1x-low fold 1 training and validation MSE loss across epochs

The training and validation MSE losses across epochs show a smaller gap between the two curves, indicating better generalization and reduced overfitting. The convergence is smoother during the early epochs, suggesting that a lower learning rate helps stabilize the initial training phase. At later epochs, both the training and validation losses remain stable, indicating that the model has likely converged and is learning consistently.

Overall validation MAE loss:

Table 6.4: Validation MAE loss for R-120-1x-low and R-120-1x-low-c, with and without clamping

R-120-1x-low un-clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.393	0.138	0.265	0.257
Fold 2	0.250	0.196	0.223	
Fold 3	0.346	0.219	0.283	
clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.268	0.138	0.203	0.229
Fold 2	0.250	0.196	0.223	
Fold 3	0.302	0.219	0.261	

R-120-1x-low-c un-clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.401	0.156	0.279	0.264
Fold 2	0.245	0.183	0.214	
Fold 3	0.376	0.223	0.299	
clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.311	0.156	0.233	0.240
Fold 2	0.245	0.183	0.214	
Fold 3	0.320	0.223	0.272	

The results show that angle MAE remains considerably higher than velocity MAE across all folds, consistent with earlier observations. Although the MSE loss curves have improved, indicating reduced impact from outlier predictions, the MAE values have not shifted significantly. This suggests that while the final MAE validation loss did not improve substantially, the overall training behavior has become more robust.

The models exhibit cleaner convergence patterns, reduced overfitting tendencies, and greater stability during training. Based on these findings, the current learning rate is considered appropriate, and no further adjustments are necessary for subsequent experiments.

6.5 Data augmentation to increase range of angle

From previous models, it was consistently observed that the validation MAE loss for angle was higher than that for velocity. This can be due to the dataset containing only four distinct angle values while velocity values covering a broader range. Hence, data augmentation is applied.

Data augmentation refers to artificially expanding the dataset by applying transformations to existing samples.

In this case, horizontal flipping was used to double the size of the dataset.

For each original image with labels (y_1, y_2) an augmented version is created with a horizontally flipped image and a new set of labels (y_3, y_2) , where:

- y_1 = normalised angle
- y_2 = normalised velocity
- y_3 = normalised augmented angle
- where $y_3^{\text{raw}} = y_1^{\text{raw}} * -1$

The models R-aug and R-aug-c are using the new S_{nc} (augmented) and new S_{c} (augmented), across 120 epochs and compared to the previous R-120-1x-low and R-120-1x-low-c model. Since the cropped and non-cropped versions yielded similar comparative results, only the non-cropped (R-aug) model will be analyzed in detail.

Training and validation MSE loss across epochs

The training and validation MSE loss across epoches for R-aug fold 1 model is shown in figure 6.4:

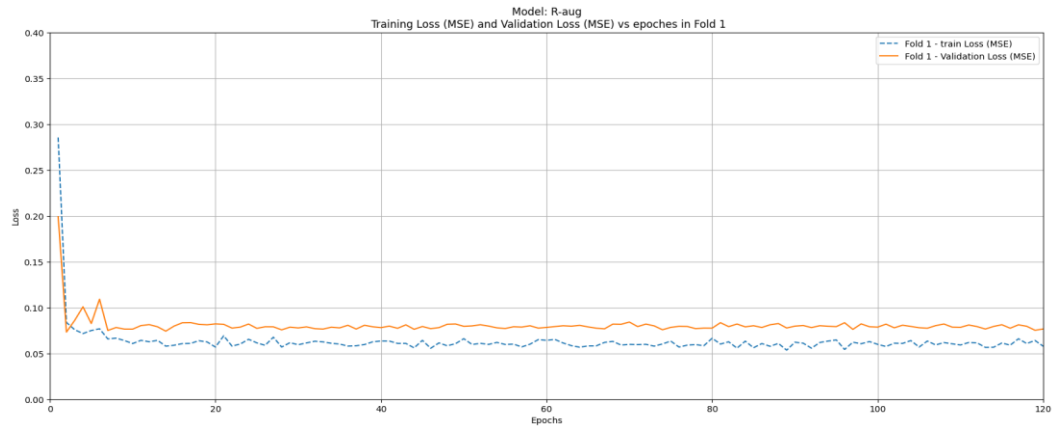


Figure 6.4: R-aug fold 1 training and validation MSE loss across epochs

R-aug fold 1 training and validation MSE loss across epochs

Compared to the previous lower learning rate models, the training and validation losses seems to converge faster and more smoothly. Next, the overall MSE losses across epochs are observed to be lower. Moreover, a slightly larger gap between training and validation loss is observed, indicating slightly increased overfitting.

Overall validation MAE loss:

Table 6.5 shows the overall validation MAE loss for both R-aug-mid and R-aug-c, with and without clamping of output values prediction to (0,1)

MAE validation loss for R-aug-mid and R-aug-c, with and without clamping

Table 6.5: Validation MAE loss for R-aug and R-aug-c with and without

R-aug un-clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.228	0.220	0.224	0.225
Fold 2	0.300	0.203	0.251	
Fold 3	0.244	0.156	0.200	
clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.228	0.220	0.224	0.220
Fold 2	0.269	0.201	0.235	
Fold 3	0.244	0.156	0.200	

clamping

R-aug-c un-clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.285	0.218	0.251	0.242
Fold 2	0.328	0.204	0.266	
Fold 3	0.254	0.162	0.208	
clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.245	0.208	0.227	0.223
Fold 2	0.271	0.198	0.234	
Fold 3	0.254	0.162	0.208	

The results show that the overall clamped MAE validation loss for R-aug is better than that of previous models. Specifically, the angle MAE validation loss has decreased, while the velocity MAE validation loss has slightly increased. The differences between clamped and unclamped angle MAE losses are smaller compared to previous models, indicating a reduction in outlier predictions outside the valid range.

Overall, the model shows stronger performance in predicting the impact angle compared to previous reduced learning rate models, although this improvement comes at a slight expense of increased overfitting for velocity, since each velocity value is effectively used twice in the augmented dataset.

6.6 Investigation of prediction accuracy on individual data points

After refining the CNN models, R-120-1x-low was identified as the strongest performing model. To assess the model's capability in predicting impact parameters, graphs were plotted to show the differences between the actual and predicted impact velocities and angles for each data point. For each prediction, the model corresponding to the fold where the data point appeared in the validation set was used. This approach avoids bias from training data exposure and ensures a fair evaluation of model performance on unseen samples.

The graphs for both angle and velocity predictions are shown in Figure 6.5. Enlarged versions of the graphs are provided in the Appendix.

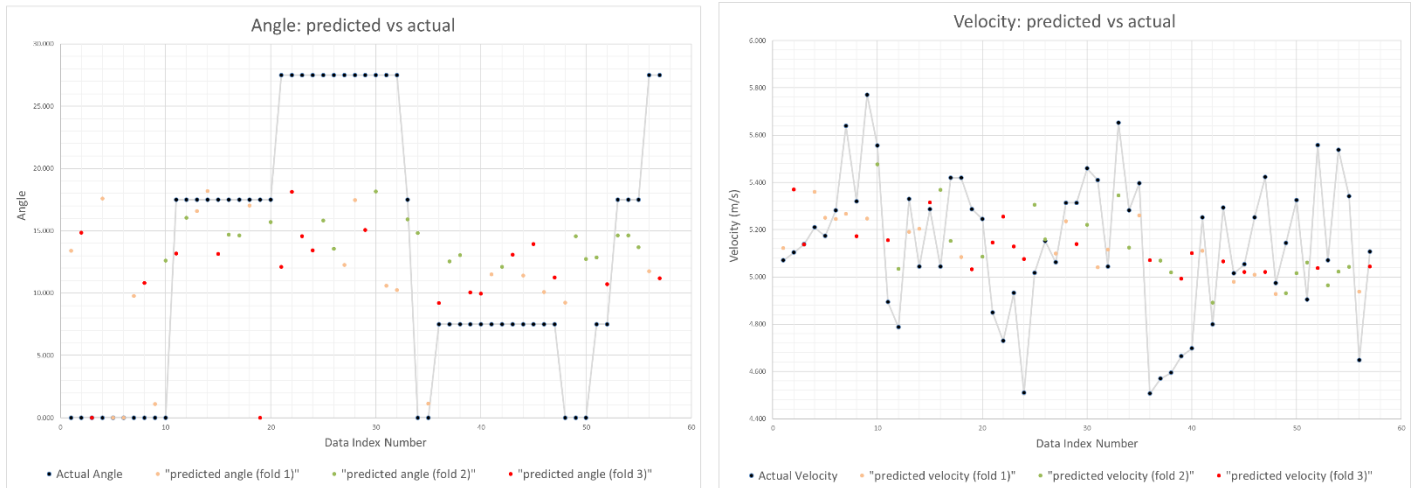


Figure 6.5s: Comparison between actual and predicted angle (left) and velocity (right) using R-120-1x-low model

From Figure 6.6, it is observed that the model predicts impact angles loosely.

Firstly, the model predicts 0° impacts relatively well. However, the model is only able to vaguely differentiate between 7.5° and 17.5° instead of predicting the values. Next, predictions for 27.5° are highly inaccurate.

In contrast, the model shows better performance for velocity prediction. It can differentiate between different velocity ranges and predict them loosely. However, it does struggle to accurately predict values at the extreme ends of the velocity range.

7 Conclusions

Through the iterative development and analysis of various CNN models, this research demonstrates that while CNNs are able to abstract some features from fractured tempered glass images and gain a basic sensing of the impact parameters, they are not able to predict these parameters accurately. In particular, impact angle prediction is notably more challenging than velocity prediction, with a larger number of outlier predictions observed. This difficulty may be attributed to the limited variation in angle values within the experimental dataset. Additionally, the results indicate that increasing the image resolution beyond a certain point does not enable the CNN models to extract additional useful features from the tempered glass crack patterns, suggesting that resolution alone is insufficient to improve model performance.

8 Recommendations for further work

While this research shows that it is not feasible to accurately predict tempered glass impact parameters using CNNs with the current dataset, it serves as a valuable starting reference for future studies on the analysis of glass crack patterns using CNN modelling.

Future work can explore predicting impact parameters for other types of glass, such as float glass and laminated glass, which may exhibit different crack characteristics.

Expanding the size of the dataset and increasing the variation in impact angles would also be beneficial to improve the learning capabilities of CNN models.

Moreover, alternative parameters such as impact energy and angle could be used instead of velocity and angle, potentially offering better predictive performance.

During CNN model training, different optimization algorithms, such as Stochastic Gradient Descent (SGD), could be tested to explore their impact on convergence behaviour.

Further investigations could also be conducted to determine the most optimal image resolution for training. If the ideal resolution is known, researchers can visually inspect images and prioritize the preservation of relevant fracture features for better model learning.

9 References

- Ernsberger, F. M. (1980). Techniques of strengthening glasses. In F. M. Ernsberger, *Elasticity and Strength in Glasses (Vol. 5 of Glass: Science and Technology)* (pp. 133–144). Academic Press, Inc.
- Hug, N. (n.d.). *Pytorch Vision GitHub repository*. Retrieved from Github: <https://github.com/pytorch/vision/tree/main/torchvision/models>
- International, A. (2015). *Standard Terminology of Glass and Glass Products (ASTM C162-05)*.
- International, A. (2019). *ASTM E2330-19: Standard test method for determination of concentrations of elements in glass samples using inductively coupled plasma mass spectrometry (ICP-MS) for forensic comparisons*. ASTM International.
- International, A. (2019). *ASTM E2330-19: Standard test method for determination of concentrations of elements in glass samples using inductively coupled plasma mass spectrometry (ICP-MS) for forensic comparisons*. ASTM International.
- Kadri Turgut Gürsel, F. Ö. (2014). Investigation of impact characteristics of tempered vehicle and vessel glasses. *Materialwissenschaft und Werkstofftechnik*.
- Long Bin Tan, N. D. (2022). Prediction and Optimization of Process Parameters for Composite Thermoforming Using a Machine Learning Approach. *polymers*.
- Neelesh Tiwari, A. H. (2019). Evidential significance of multiple fracture patterns on the glass in forensic ballistics. *Egyptian Journal of Forensic Sciences*, 1–5.
- NVIDIA. (n.d.). *ResNeXt101* . Retrieved from pytorch: https://pytorch.org/hub/nvidia_deeplearningexamples_resnext/
- Pytorch. (n.d.). *Models and pre-trained weights*. Retrieved from Pytorch: <https://pytorch.org/vision/main/models.html>
- Remya, S., Dwaz, G. C., Pillai, P. G., & Navami, S. K. (2023). A novel framework for forensic bone age estimation using CNN architecture and multimodal fusion. *7th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*. IEEE.
- Roberto Ballarini, G. P.-C. (2016). The lower bound for glass strength and its interpretation with generalized Weibull statistics for structural applications. *Journal of Engineering Mechanics*.
- Stewart, R. J., & Prakash, N. (2013). Modeling dynamic fragmentation of tempered glass. *Engineering Fracture Mechanics*, 27–35.
- Trejos, T. (2022). *Forensic glass examinations—A review focused on elemental spectrochemical analysis*.
- Tyagi, S., & Yadav, D. (2023). ForensicNet: Modern convolutional neural network-based image forgery detection network. *Journal of Forensic Sciences*.

10 Appendix

Glass impact test parameters

Dataset Index	height (cm)	angle (degree)
1	0.000	5.070
2	0.000	5.103
3	0.000	5.138
4	0.000	5.209
5	0.000	5.173
6	0.000	5.282
7	0.000	5.640
8	0.000	5.320
9	0.000	5.770
10	0.000	5.556
11	17.500	4.893
12	17.500	4.786
13	17.500	5.330
14	17.500	5.043
15	17.500	5.287
16	17.500	5.043
17	17.500	5.419
18	17.500	5.419
19	17.500	5.287
20	17.500	5.245
21	27.500	4.849
22	27.500	4.729
23	27.500	4.932
24	27.500	4.508
25	27.500	5.017
26	27.500	5.152
27	27.500	5.061
28	27.500	5.313
29	27.500	5.313
30	27.500	5.459

Dataset Index	height (cm)	angle (degree)
31	27.500	5.410
32	27.500	5.043
33	17.500	5.652
34	0.000	5.282
35	0.000	5.396
36	7.500	4.506
37	7.500	4.568
38	7.500	4.594
39	7.500	4.664
40	7.500	4.697
41	7.500	5.252
42	7.500	4.798
43	7.500	5.293
44	7.500	5.015
45	7.500	5.053
46	7.500	5.252
47	7.500	5.422
48	0.000	4.973
49	0.000	5.143
50	0.000	5.325
51	7.500	4.904
52	7.500	5.558
53	17.500	5.069
54	17.500	5.538
55	17.500	5.341
56	27.500	4.647
57	27.500	5.106

Glass impact test parameters

Dataset Index	height (cm)	angle (degree)	Dataset Index	height (cm)	angle (degree)
1	0.000	5.070	31	27.500	5.410
2	0.000	5.103	32	27.500	5.043
3	0.000	5.138	33	17.500	5.652
4	0.000	5.209	34	0.000	5.282
5	0.000	5.173	35	0.000	5.396
6	0.000	5.282	36	7.500	4.506
7	0.000	5.640	37	7.500	4.568
8	0.000	5.320	38	7.500	4.594
9	0.000	5.770	39	7.500	4.664
10	0.000	5.556	40	7.500	4.697
11	17.500	4.893	41	7.500	5.252
12	17.500	4.786	42	7.500	4.798
13	17.500	5.330	43	7.500	5.293
14	17.500	5.043	44	7.500	5.015
15	17.500	5.287	45	7.500	5.053
16	17.500	5.043	46	7.500	5.252
17	17.500	5.419	47	7.500	5.422
18	17.500	5.419	48	0.000	4.973
19	17.500	5.287	49	0.000	5.143
20	17.500	5.245	50	0.000	5.325
21	27.500	4.849	51	7.500	4.904
22	27.500	4.729	52	7.500	5.558
23	27.500	4.932	53	17.500	5.069
24	27.500	4.508	54	17.500	5.538
25	27.500	5.017	55	17.500	5.341
26	27.500	5.152	56	27.500	4.647
27	27.500	5.061	57	27.500	5.106
28	27.500	5.313			
29	27.500	5.313			
30	27.500	5.459			

Processed dataset specification

Index	Image name	y1 Normalized angle	y2 Normalized velocity	Index	Image name	y1 Normalized angle	y2 Normalized velocity
1	P01	0.00	0.45	31	P31	1.00	0.72
2	P02	0.00	0.47	32	P32	1.00	0.42
3	P03	0.00	0.50	33	P33	0.64	0.91
4	P04	0.00	0.56	34	P34	0.00	0.61
5	P05	0.00	0.53	35	P35	0.00	0.70
6	P06	0.00	0.61	36	P36	0.27	0.00
7	P07	0.00	0.90	37	P37	0.27	0.05
8	P08	0.00	0.64	38	P38	0.27	0.07
9	P09	0.00	1.00	39	P39	0.27	0.13
10	P10	0.00	0.83	40	P40	0.27	0.15
11	P11	0.64	0.31	41	P41	0.27	0.59
12	P12	0.64	0.22	42	P42	0.27	0.23
13	P13	0.64	0.65	43	P43	0.27	0.62
14	P14	0.64	0.42	44	P44	0.27	0.40
15	P15	0.64	0.62	45	P45	0.27	0.43
16	P16	0.64	0.42	46	P46	0.27	0.59
17	P17	0.64	0.72	47	P47	0.27	0.72
18	P18	0.64	0.72	48	P48	0.00	0.37
19	P19	0.64	0.62	49	P49	0.00	0.50
20	P20	0.64	0.58	50	P50	0.00	0.65
21	P21	1.00	0.27	51	P51	0.27	0.31
22	P22	1.00	0.18	52	P52	0.27	0.83
23	P23	1.00	0.34	53	P53	0.64	0.45
24	P24	1.00	0.00	54	P54	0.64	0.82
25	P25	1.00	0.40	55	P55	0.64	0.66
26	P26	1.00	0.51	56	P56	1.00	0.11
27	P27	1.00	0.44	57	P57	1.00	0.47
28	P28	1.00	0.64				
29	P29	1.00	0.64				
30	P30	1.00	0.75				

CNN models parameters

$\mathbf{f_{\theta^n}}$	Model Name <i>R – epochs – pixels – learning rate – crop</i>	Epochs	Pixel dimension	Learning Parameters					Image dataset (S _c = cropped) (S _{nc} = non-cropped)
				Factor	Patience	Threshold	Learning rate (Lr)	Min Lr	
$\mathbf{f_{\theta^1}}$	R-060-1x-mid	60	336 x 448	0.5	5	1E-04	1E-05	1E-06	S _{nc}
$\mathbf{f_{\theta^2}}$	R-060-1x-mid-c	60	336 x 504	0.5	5	1E-04	1E-05	1E-06	S _c
$\mathbf{f_{\theta^3}}$	R-060-2x-mid	60	672 x 896	0.5	5	1E-04	1E-05	1E-06	S _{nc}
$\mathbf{f_{\theta^4}}$	R-060-2x-mid-c	60	672 x 1008	0.5	5	1E-04	1E-05	1E-06	S _c
$\mathbf{f_{\theta^5}}$	R-120-1x-mid	120	336 x 448	0.5	5	1E-04	1E-05	1E-06	S _{nc}
$\mathbf{f_{\theta^6}}$	R-120-1x-mid-c	120	336 x 504	0.5	5	1E-04	1E-05	1E-06	S _c
$\mathbf{f_{\theta^7}}$	R-120-1x-low	120	336 x 448	0.3	3	1E-03	1E-05	5E-08	S _{nc}
$\mathbf{f_{\theta^8}}$	R-120-1x-low-c	120	336 x 504	0.3	3	1E-03	1E-05	5E-08	S _c
$\mathbf{f_{\theta^9}}$	R-aug	120	336 x 448	0.3	3	1E-03	1E-05	5E-08	S _{nc} (augmented)
$\mathbf{f_{\theta^{10}}}$	R-aug-c	120	336 x 504	0.3	3	1E-03	1E-05	5E-08	S _c (augmented)

MAE validation loss for R-060-1x-mid and R-060-1x-mid-c, with and without clamping

R-060-1x-mid un-clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.537	0.139	0.338	0.275
Fold 2	0.261	0.200	0.230	
Fold 3	0.294	0.223	0.258	
clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.286	0.139	0.212	0.234
Fold 2	0.261	0.200	0.230	
Fold 3	0.294	0.223	0.258	

R-060-1x-mid-c un-clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.467	0.151	0.309	0.270
Fold 2	0.274	0.192	0.233	
Fold 3	0.338	0.200	0.269	
clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.286	0.151	0.218	0.233
Fold 2	0.274	0.192	0.233	
Fold 3	0.294	0.200	0.247	

MAE validation loss for R-060-2x-mid and R-060-2x-mid-c, with and without clamping

R-060-2x-mid un-clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.456	0.201	0.328	0.281
Fold 2	0.265	0.190	0.228	
Fold 3	0.380	0.197	0.288	
clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.285	0.167	0.226	0.236
Fold 2	0.265	0.190	0.228	
Fold 3	0.311	0.197	0.254	

R-060-2x-mid-c un-clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.323	0.151	0.237	0.255
Fold 2	0.280	0.181	0.230	
Fold 3	0.397	0.200	0.298	
clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.308	0.151	0.230	0.240
Fold 2	0.280	0.181	0.230	
Fold 3	0.323	0.200	0.261	

MAE validation loss for R-120-1x-mid and R-120-1x-mid-c, with and without clamping

R-120-1x-mid un-clamped				
Fold	MAE			Avg overall
	angle	velocity	overall	
Fold 1	0.583	0.200	0.392	0.302
Fold 2	0.264	0.191	0.227	
Fold 3	0.367	0.207	0.287	
clamped				
Fold	MAE			Avg overall
	angle	velocity	overall	
Fold 1	0.261	0.153	0.207	0.229
Fold 2	0.264	0.191	0.227	
Fold 3	0.301	0.207	0.254	

R-120-1x-mid-c un-clamped				
Fold	MAE			Avg overall
	angle	velocity	overall	
Fold 1	0.318	0.153	0.236	0.240
Fold 2	0.248	0.189	0.219	
Fold 3	0.337	0.198	0.267	
clamped				
Fold	MAE			Avg overall
	angle	velocity	overall	
Fold 1	0.304	0.153	0.229	0.238
Fold 2	0.248	0.189	0.219	
Fold 3	0.337	0.198	0.267	

MAE validation loss for R-120-1x-low and R-120-1x-low-c, with and without clamping

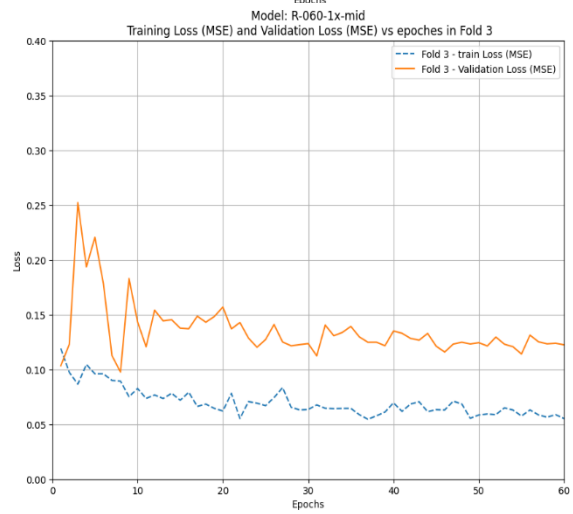
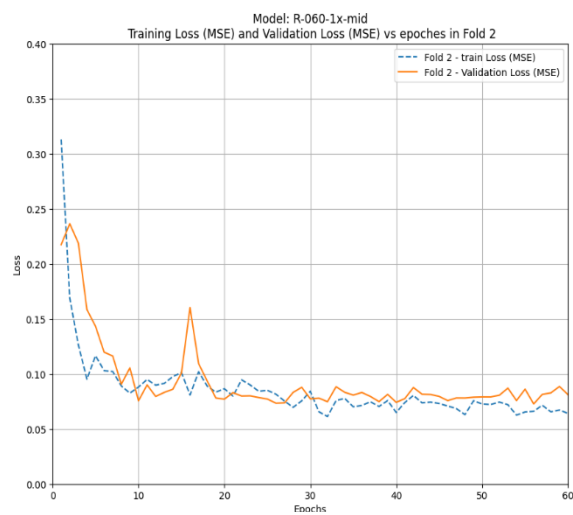
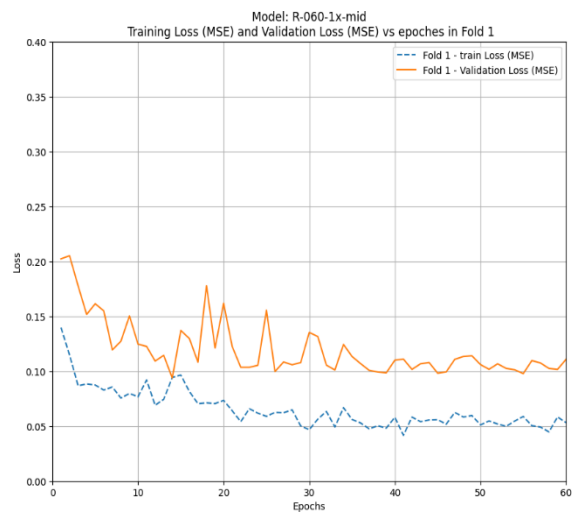
R-120-1x-low un-clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.393	0.138	0.265	0.257
Fold 2	0.250	0.196	0.223	
Fold 3	0.346	0.219	0.283	
clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.268	0.138	0.203	0.229
Fold 2	0.250	0.196	0.223	
Fold 3	0.302	0.219	0.261	

R-120-1x-low-c un-clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.401	0.156	0.279	0.264
Fold 2	0.245	0.183	0.214	
Fold 3	0.376	0.223	0.299	
clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.311	0.156	0.233	0.240
Fold 2	0.245	0.183	0.214	
Fold 3	0.320	0.223	0.272	

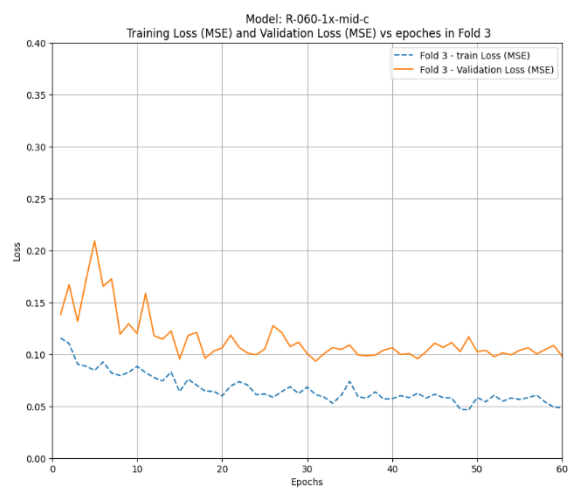
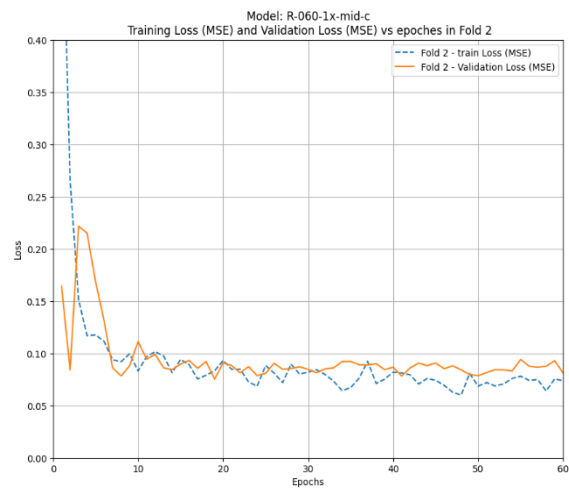
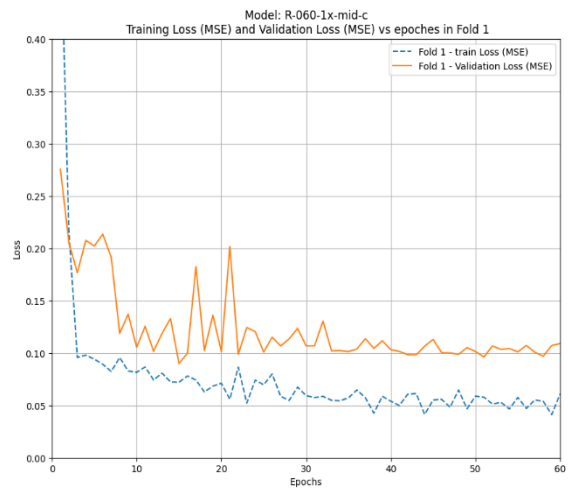
MAE validation loss for R-aug-mid and R-aug-c, with and without clamping

R-aug un-clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.228	0.220	0.224	0.225
Fold 2	0.300	0.203	0.251	
Fold 3	0.244	0.156	0.200	
clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.228	0.220	0.224	0.220
Fold 2	0.269	0.201	0.235	
Fold 3	0.244	0.156	0.200	

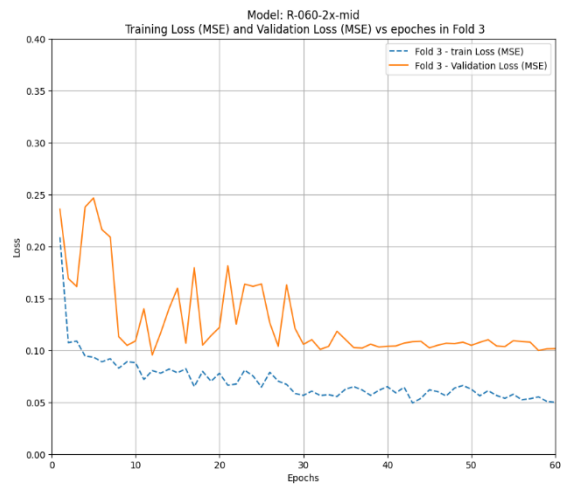
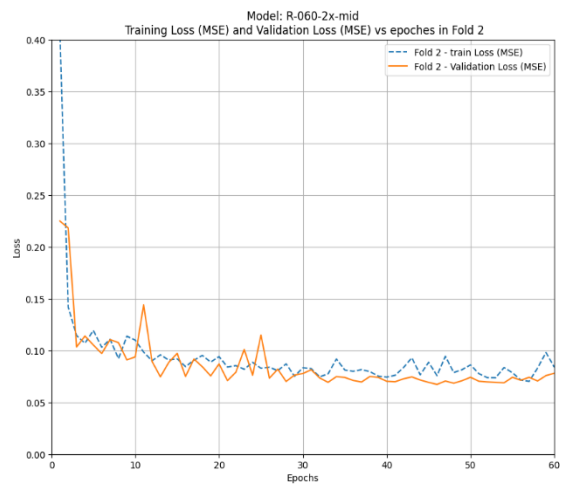
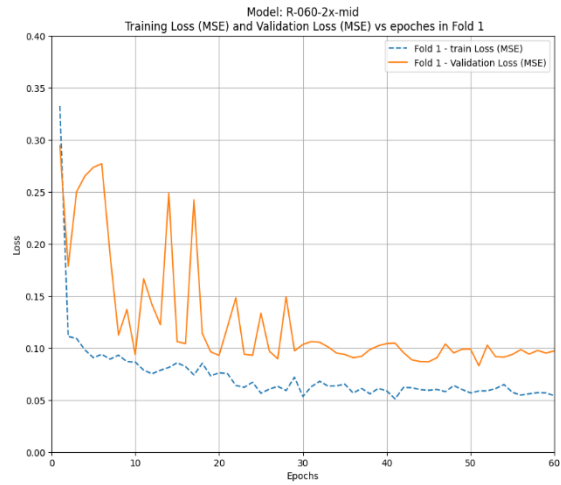
R-aug-c un-clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.285	0.218	0.251	0.242
Fold 2	0.328	0.204	0.266	
Fold 3	0.254	0.162	0.208	
clamped				
Fold	MAE			
	angle	velocity	overall	Avg overall
Fold 1	0.245	0.208	0.227	0.223
Fold 2	0.271	0.198	0.234	
Fold 3	0.254	0.162	0.208	



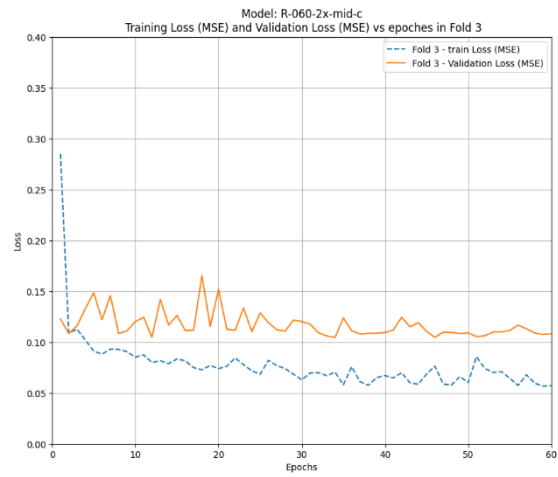
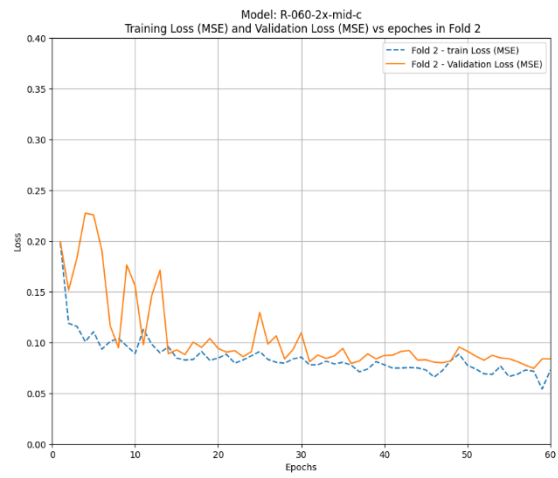
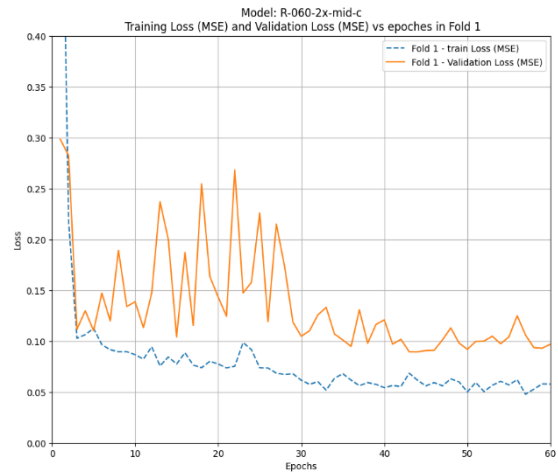
R-060-1x-mid training and validation MSE loss across epochs



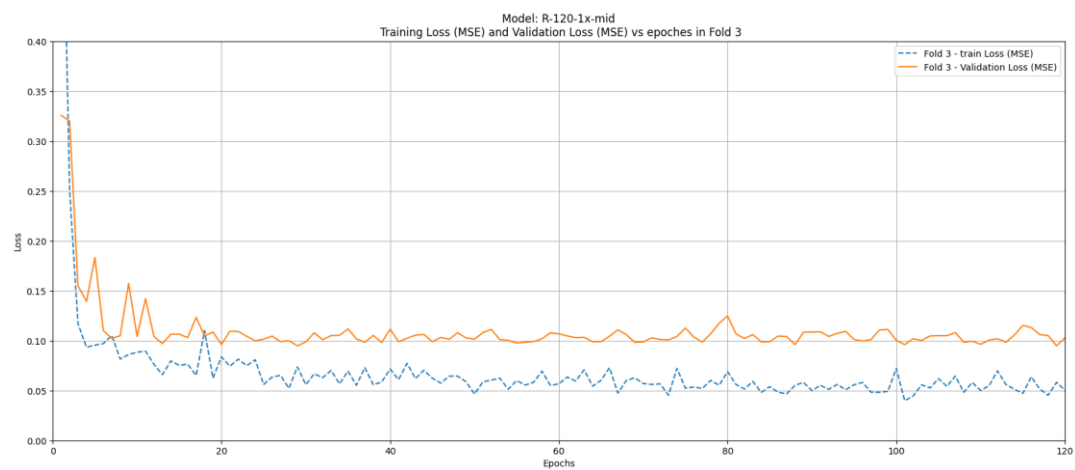
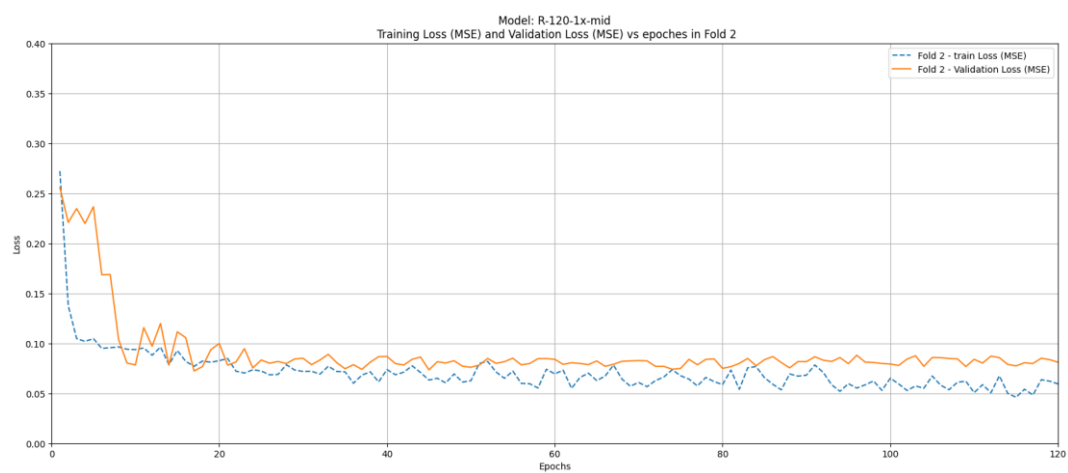
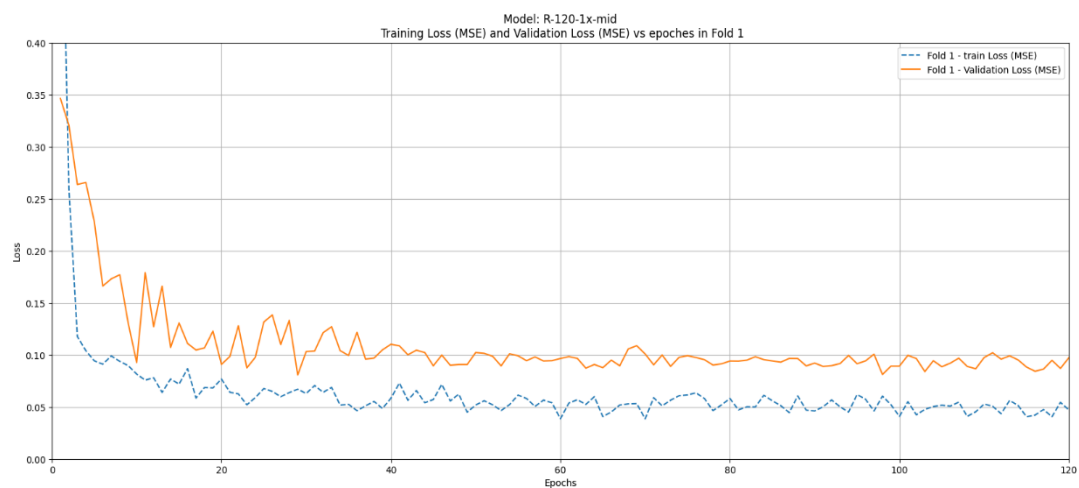
R-060-1x-mid-c training and validation MSE loss across epochs



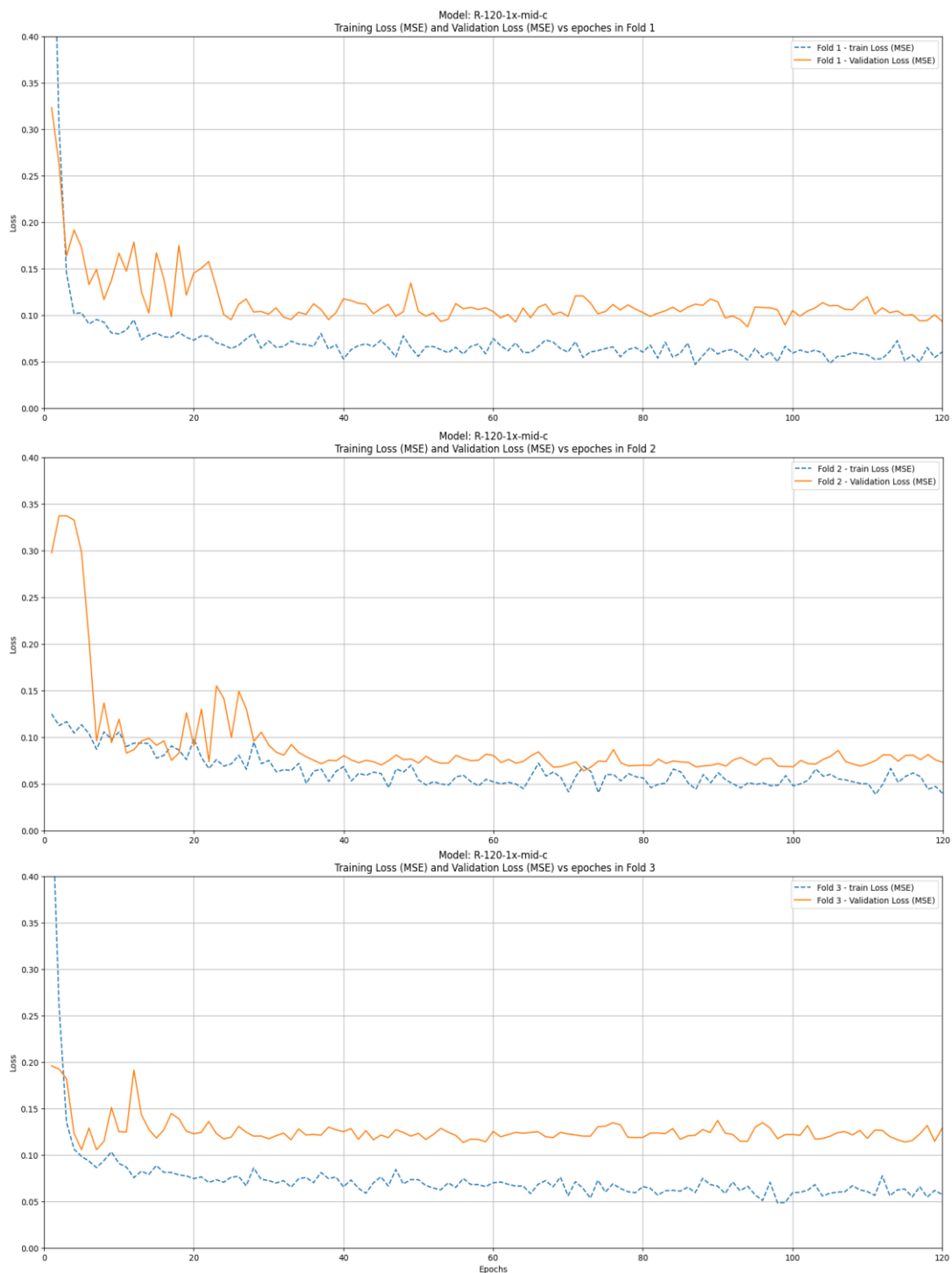
R-060-2x-mid training and validation MSE loss across epochs



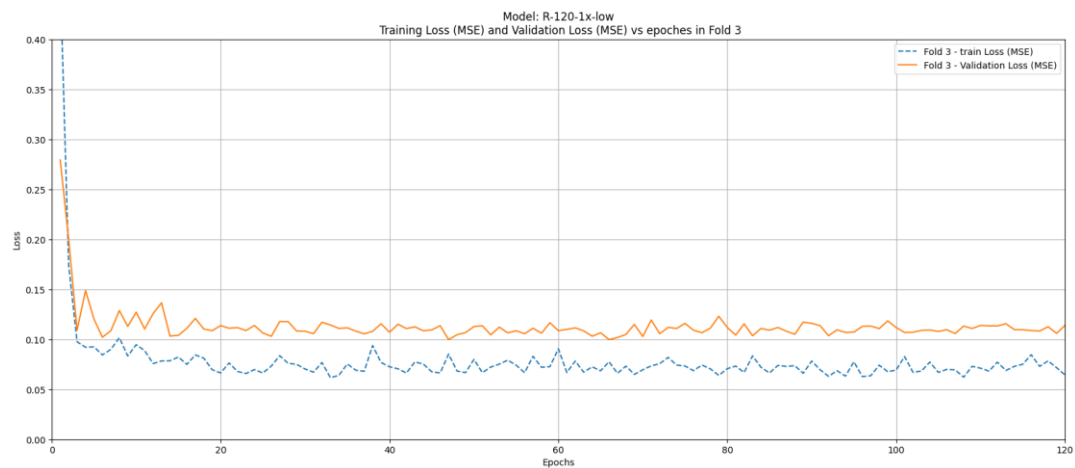
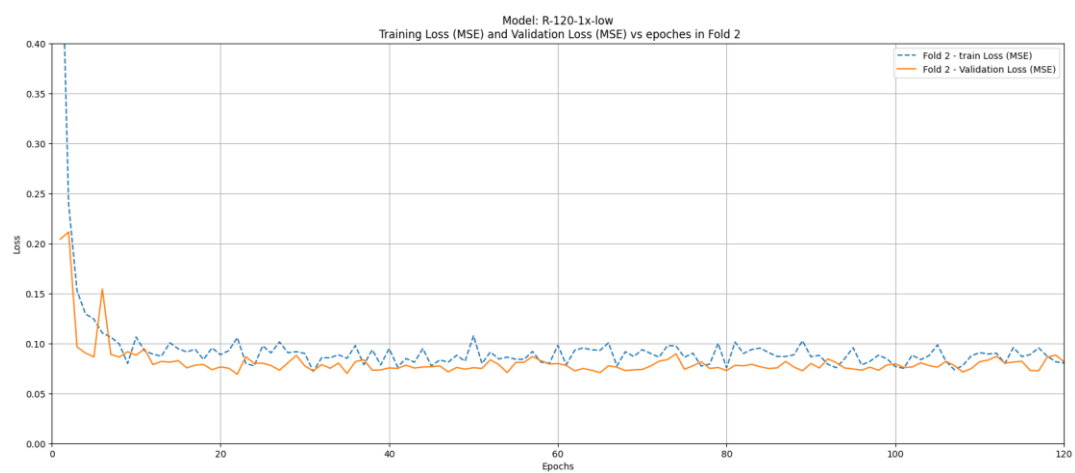
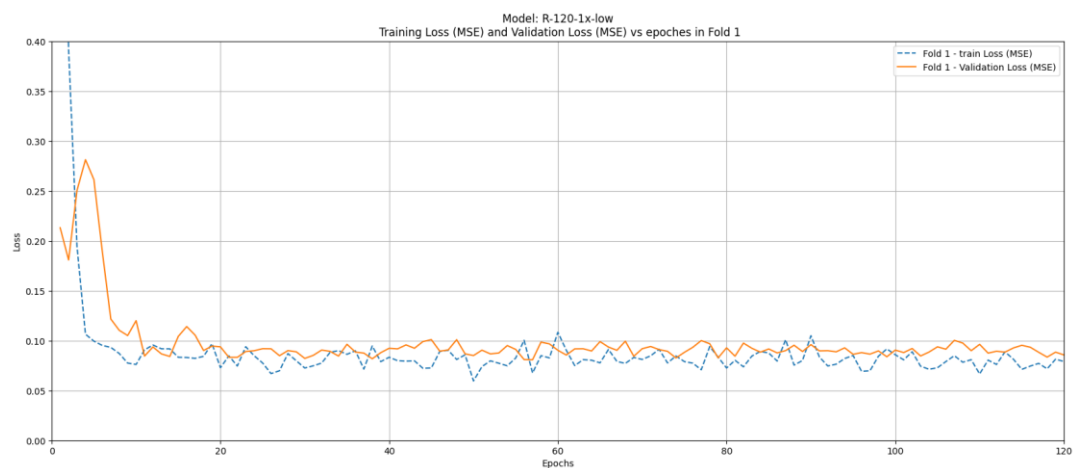
R-060-2x-mid-c training and validation MSE loss across epochs



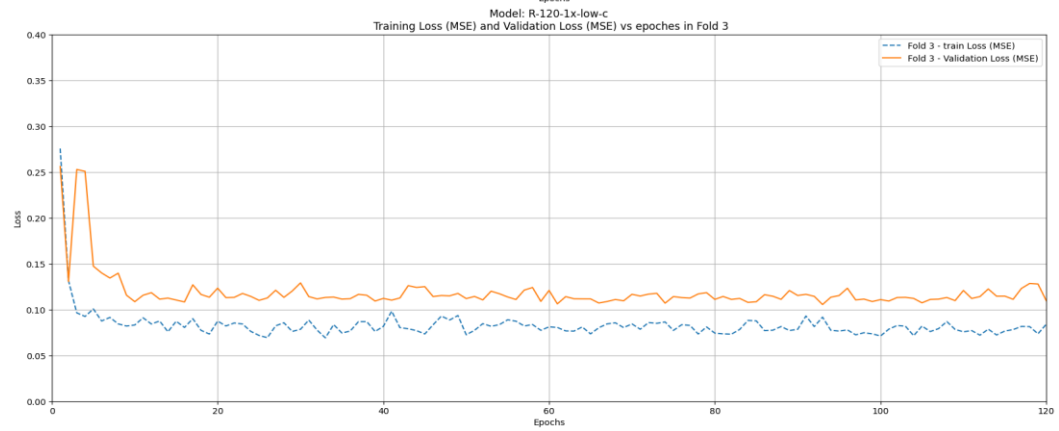
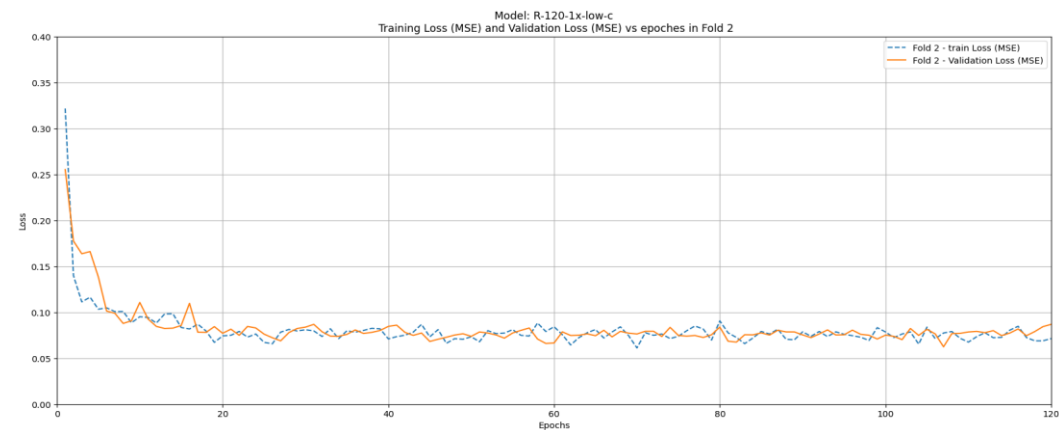
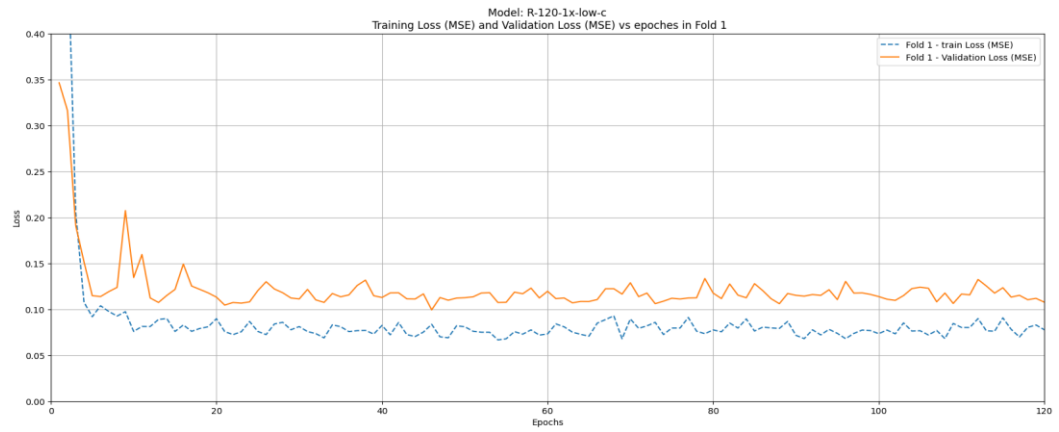
R-120-1x-mid training and validation MSE loss across epochs



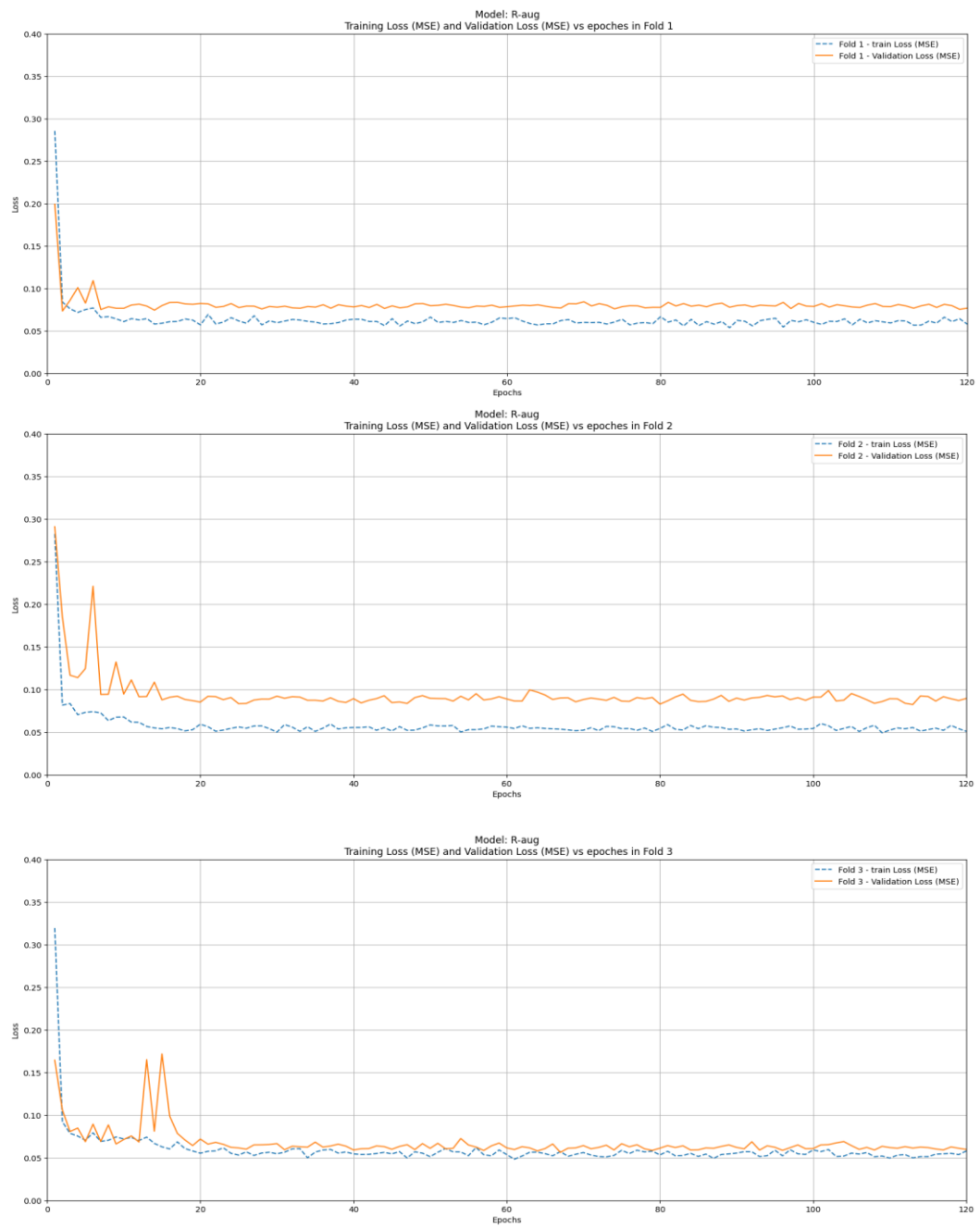
R-120-1x-mid-c training and validation MSE loss across epochs



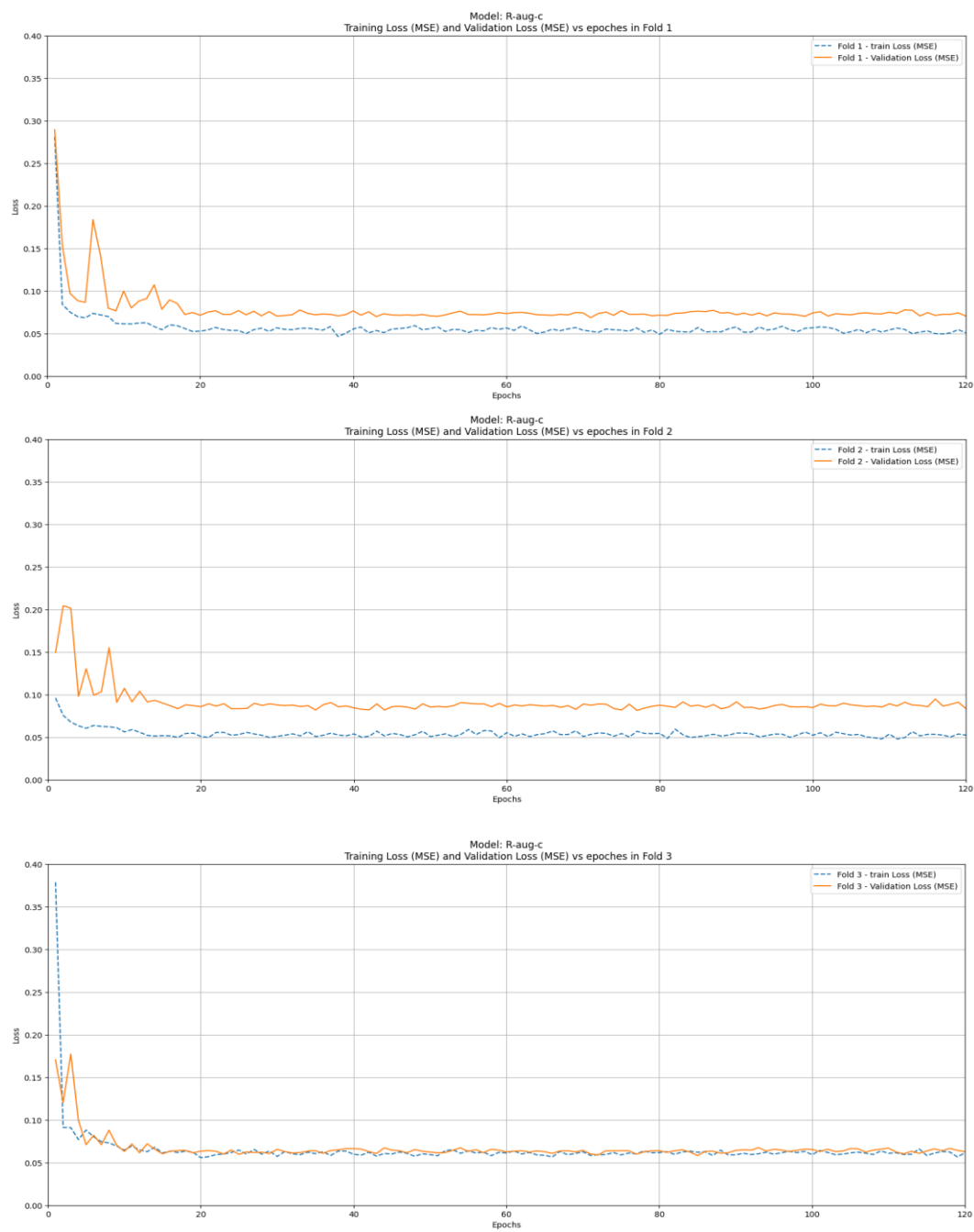
R-120-1x-low training and validation MSE loss across epochs



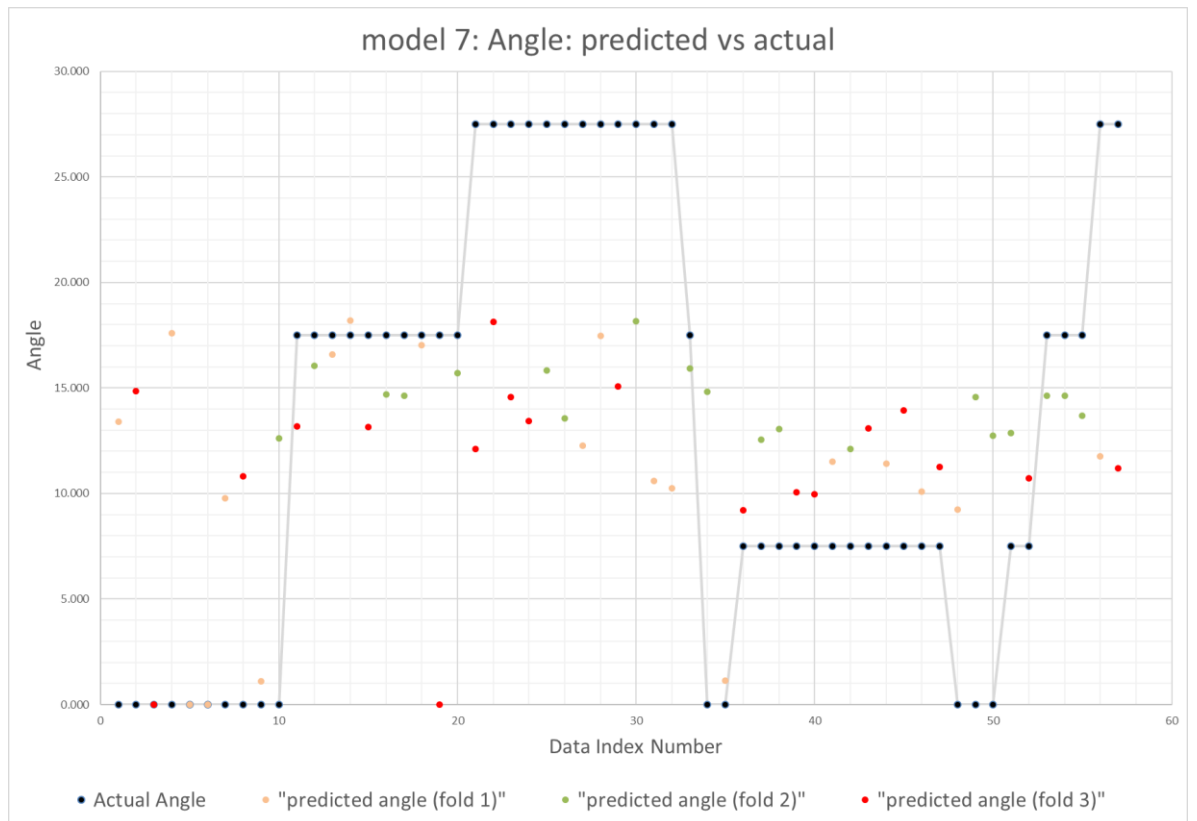
R-120-1x-low-c training and validation MSE loss across epochs



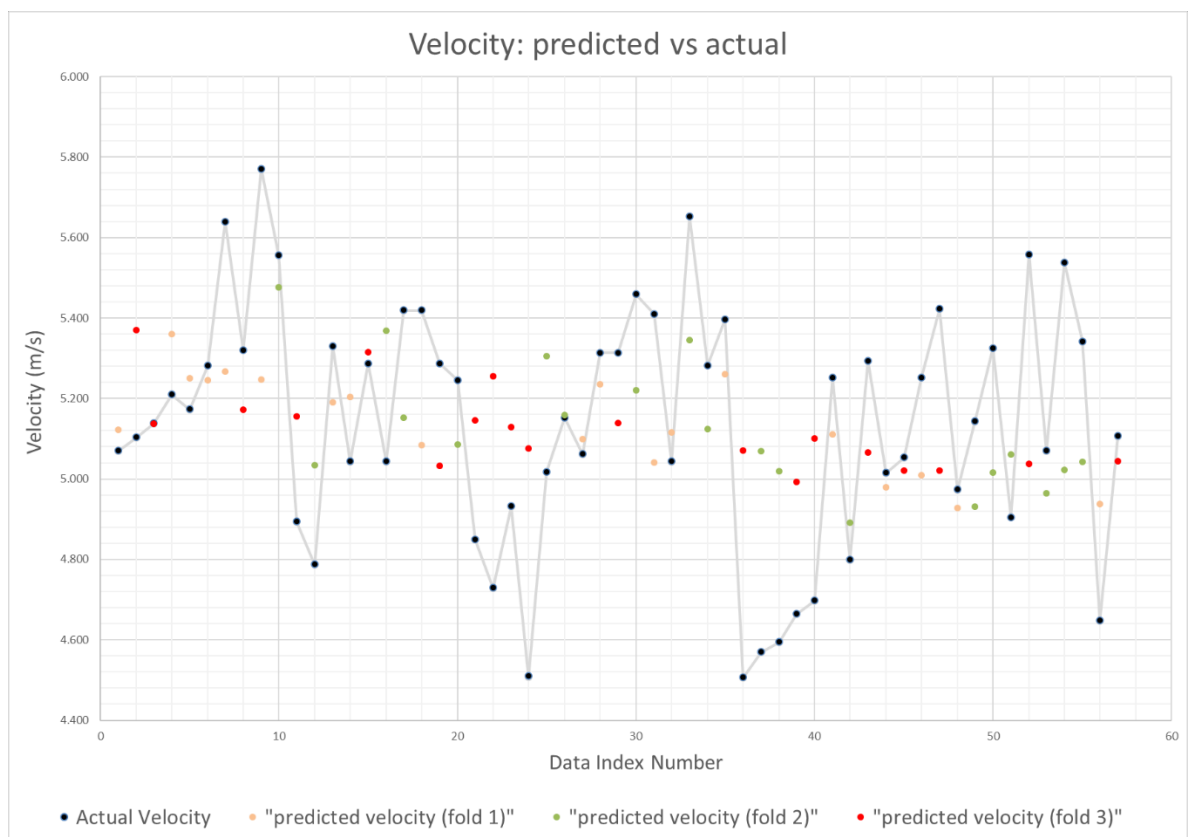
R-aug training and validation MSE loss across epochs



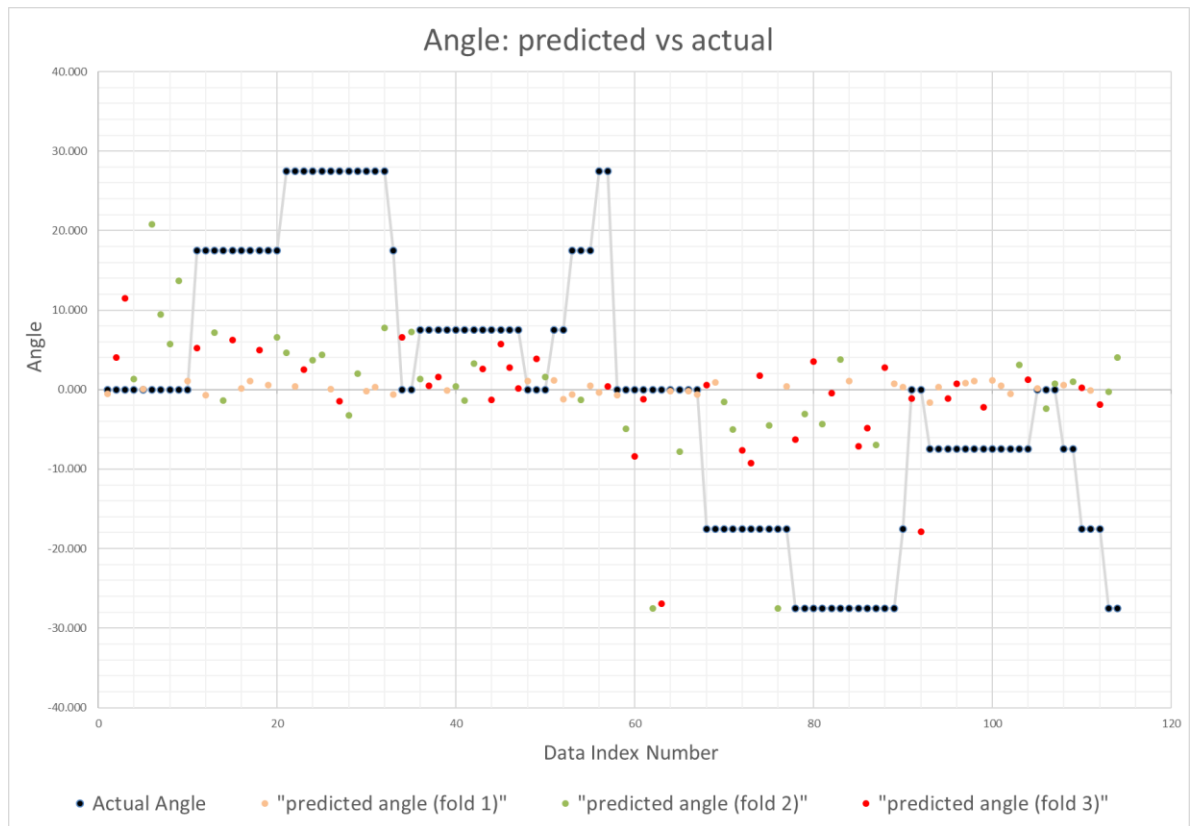
R-aug-c training and validation MSE loss across epochs



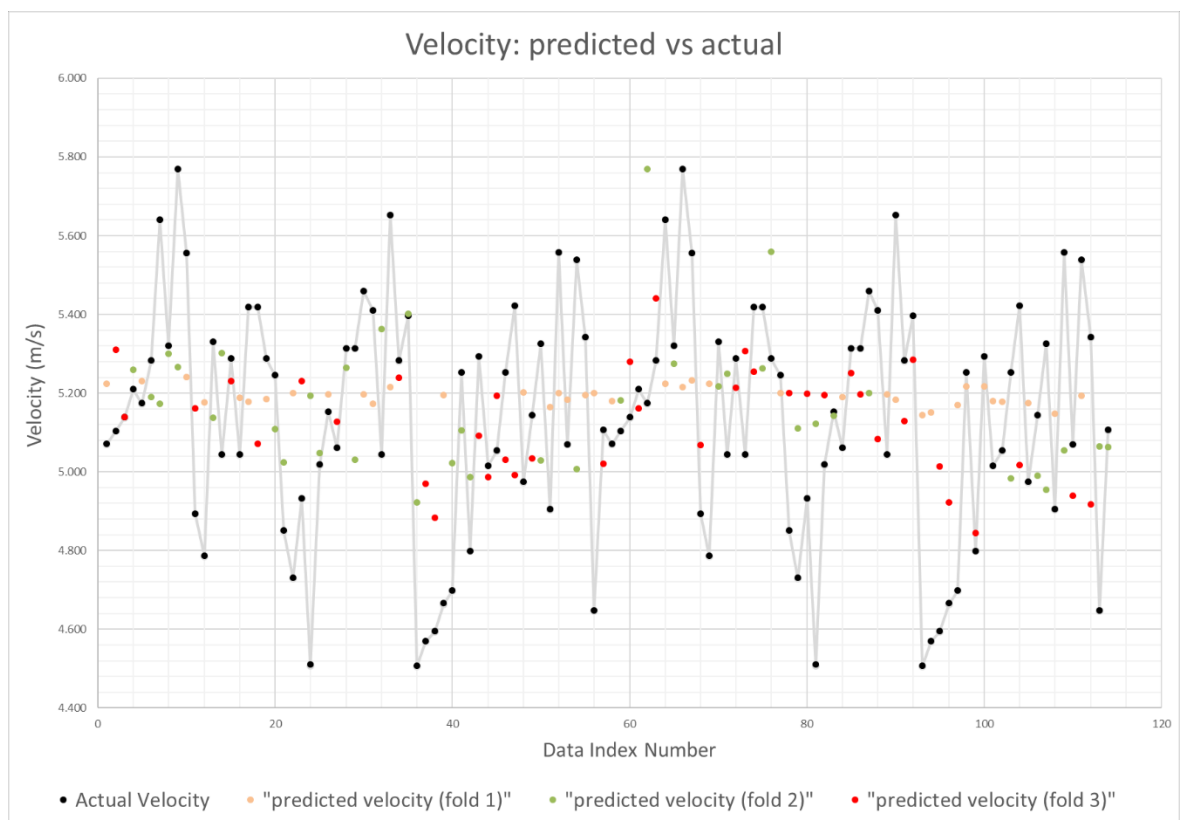
Comparison between actual and predicted angle using R-120-1x-low model



Comparison between actual and predicted velocity using R-120-1x-low model



Comparison between actual and predicted angle using R-aug model



Comparison between actual and predicted velocity using R-aug model

