Universiteit Leiden

Master Computer Science

Automation of Crack Assessment in Masonry Facades using Deep Learning

Name: Kelvin Kleijn
Student ID: 1179071
Date: 26/07/2022

Specialisation: Computer Science & Advanced Data Analytics

1st supervisor: Hao Wang
2nd supervisor: Wyke Pereboom-Huizinga (TNO)

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Contents

Automation of Crack Assessment in Masonry Facades using Deep Learning

Kelvin Kleijn

July 2021

Chapter 1

Introduction

This master thesis is the result of a collaboration between the Leiden Institute of Advanced Computer Science (LIACS), the Intelligent Imaging and Structural Reliability Departments of the Dutch Organisation of Applied Scientific Research (TNO), and the Faculty of Civil Engineering and Geo-sciences of the TU Delft. It can be viewed as a sub-project of TNO's efforts to partially automate the assessment of cracked facades.

In 2020, I joined the Intelligent Imaging department of TNO as a student intern to help advance TNO's efforts to automate the assessment of cracks in masonry facades. The ultimate aim of the project is to develop automated models that aid masonry experts in the process of assessing damaged masonry structures which feature one or multiple cracks. Though, we do not expect that it is feasible to fully automate this process in the foreseeable future, this thesis marks another step towards the attainment of this overarching goal.

1.1 Motivation

Unfortunately, the number of damage reports of masonry dwellings has risen annually over the past decade, and given that most buildings in the Netherlands were built in the mid twentieth century, many buildings are reaching the end of their life cycle [1], and thus the incidence of similar damage reports is expected to rise even further in the following decades. Furthermore, buildings make up a significant part of the built environment which constitutes a large portion of the national wealth in developed countries [2]. And to top it off, inspection of cracked facades is very costly, in some cases even requiring excavation of sites. These facts reflect the economic relevance of a well-maintained built infrastructure, and thereby justify our efforts to develop automated models to aid masonry experts in their assessment which will serve to support the maintenance of the country's rich built environment.

1.2 Problem Definition

In support of the goal to automate crack assessment, I focused on the development of a model to automate the assessment of the degree of similarity of pairs of images of cracked masonry facades as determined by masonry experts. Thus, the goal is to learn a function which takes in two separate images, each featuring a cracked-facade, and outputs the degree of similarity in close accordance with expert assessment. Formulated as such, the output can comprise of multiple components, depending on what kind of output would be most useful to aid masonry experts in their assessment of damaged facades. Formally, the task can be defined as follows

$$f : (X, X) \rightarrow (Y_1 ... Y_n); (x_1, x_2) \mapsto (y_1 ... y_n). \tag{1.1}$$

Where X denotes the set of all cracked facade images, and Y_1 through Y_n denote the sets of all possible values of output components 1 through n, respectively.

Note that a pair of images can map to multiple variables (output components), each of which conveys some additional information regarding the similarity of the two images provided as input. In our case, The function f is defined by the architecture and weights of the Neural Network that establishes it. The first neural network developed to regress the degree of similarity, yielded a scalar between 0 and 1. Thus, $n = 1$ and $Y_1 = [0, 1] \subset \mathbb{R}$.

However, numerous discussions have resulted in the decision to shift from a single output to three outputs, each of which captures a different aspect of the notion of similarity in the context of pairs of cracked masonry facades. Furthermore, the decision was made to shift from continuous similarity scores to ordinal similarity scores. In order of increasing similarity, the following ordinal categories were introduced: Very Dissimilar; Dissimilar; Similar; and Very Similar. Thus $n = 1$ and $Y_{1,2,3} = \{$*Very Dissimilar, Dissimilar, Similar, Very Similar*$\}$. The questions, the data, and the labelling task are covered in more detail in chapter 3.

Due to a shortage of real-world data, that is images of cracks in masonry facades, we have resorted to the use of synthetic cracks. Hence, the assessment of cracked facades was performed on synthetic cracks that were generated with two different crack simulation models, the Markov-Walk model and the FEM model. These simulation models will be discussed in the remainder of this thesis.

1.3 Project Background

As mentioned previously, I joined TNO in January 2021, while the project was initiated in 2019. This section presents a concise overview of efforts that were made prior to my involvement. First, I discuss the sheer lack of data, as well

as how this issue was addressed. This is followed by a brief account of early efforts to automate crack assessment. After that, the research questions will be presented, followed by a description of the approach, and this chapter closes with an outline of the remainder of the thesis.

1.3.1 Shortage of Data

Due to a shortage of images of cracked masonry facades, the researchers from TNO were faced with a serious challenge, that is the acquisition of data needed to support efforts to automate crack assessment. It is not easy to find thousands of cracked facades, and it is even harder to find thousands of cracks that belong to a predefined category of cracks. Only a handful of damage reports of cracked masonry facades were available at the time, and the team ultimately decided to resort to synthetic data to achieve their aim. Two methods for modelling and generating synthetic cracked facades based on the twelve aforementioned crack archetypes were proposed

- A Markov-Walk model to simulate cracks on a predefined set of facades. [3]
- A simulation model based on finite element analysis, which is much more extensive than the Markov-Walk model. Contrary to the Markov-Walk model, it enables one to select different aspect ratios, different settlement profiles and different facades sizes, to name just a few additional parameters. [4]

Initially, only the Markov-Walk model was developed by Dr. Arpad Rozsas. In June 2021, I updated Rozsas' original model and integrated the width as an additional parameter. Figures 1.1 through 1.12 show a sample crack that was synthetically generated with the Markov-Walk model for each of the aforementioned crack archetypes. Efforts to develop the finite element analysis simulation model (henceforth also referred to as **FEM**) started in January 2021.

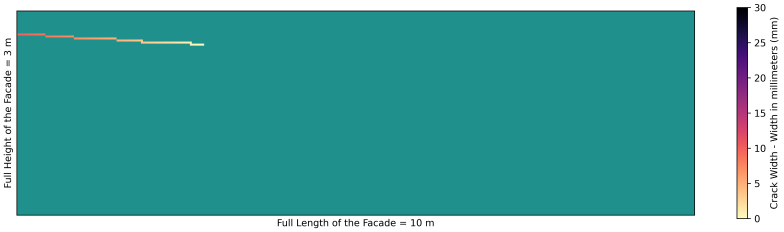

Figure 1.1: Crack Archetype 18

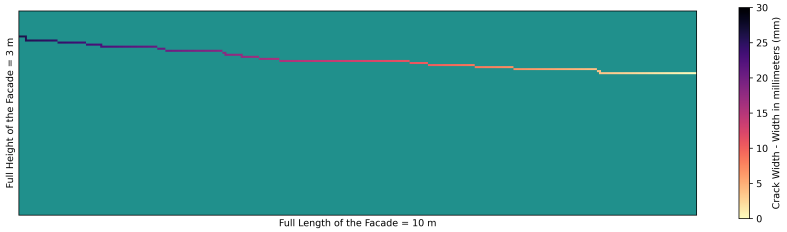

Figure 1.2: Crack Archetype 20

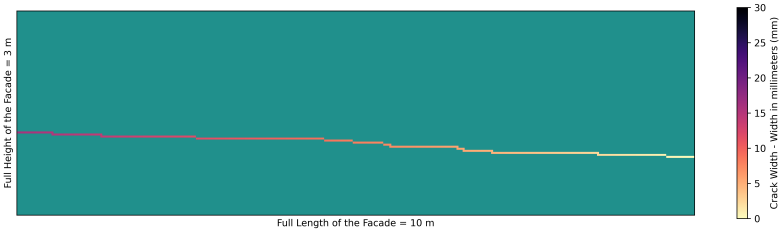

Figure 1.3: Crack Archetype 21

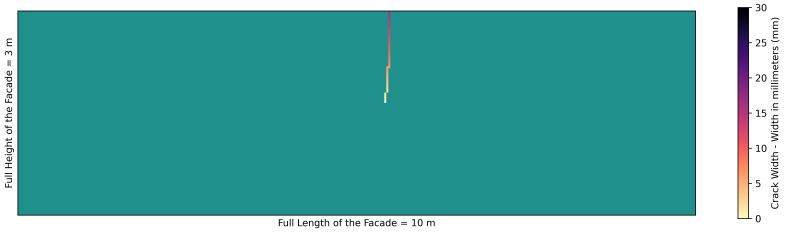

Figure 1.4: Crack Archetype 23

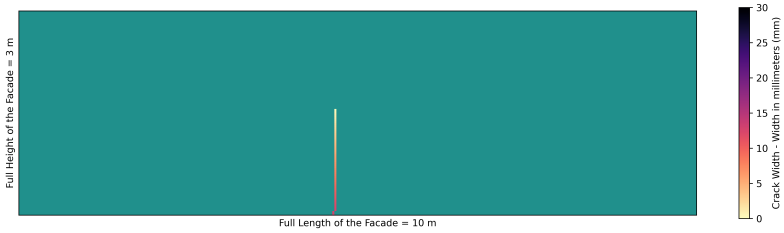

Figure 1.5: Crack Archetype 24

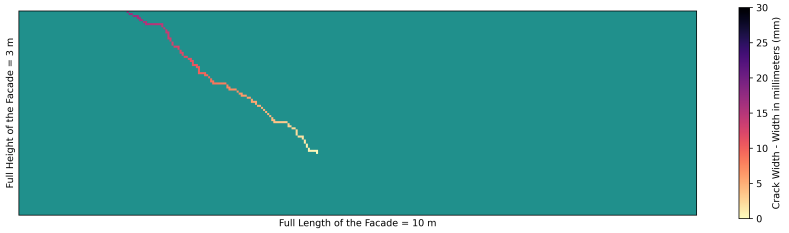

Figure 1.6: Crack Archetype 30

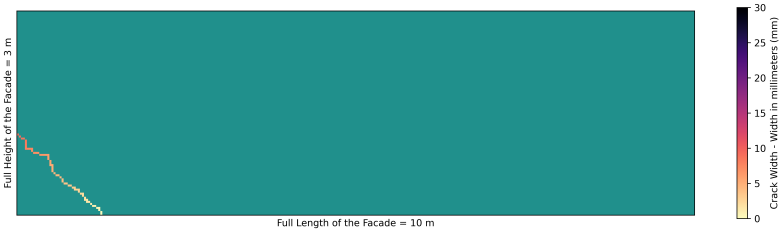

Figure 1.7: Crack Archetype 31

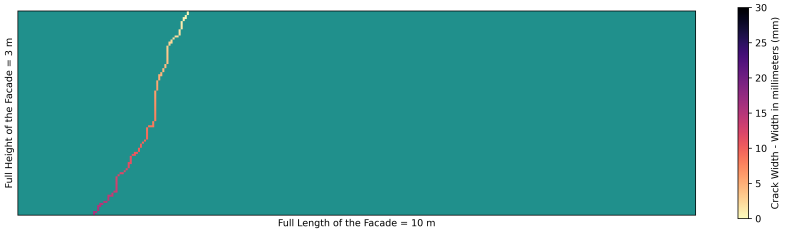

Figure 1.8: Crack Archetype 32

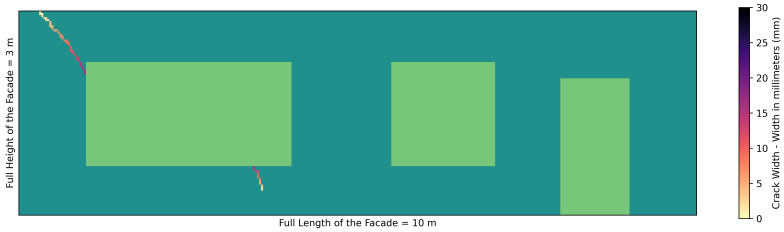

Figure 1.9: Crack Archetype 101

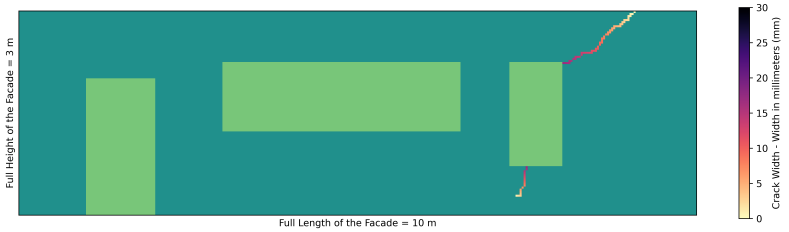

Figure 1.10: Crack Archetype 102

1.4 Automated Procedure

Early efforts focused on automation of the classification of Markov-Walk synthetically generated cracked masonry facades into one of eight crack archetypes,

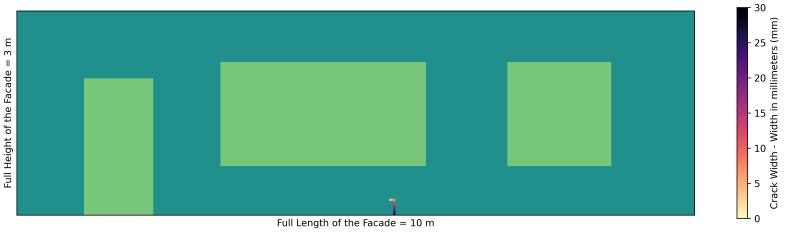

Figure 1.11: Crack Archetype 103

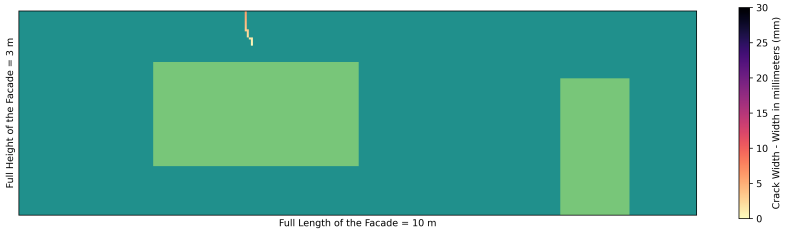

Figure 1.12: Crack Archetype 201

focusing solely on facades without openings (Figures 1.1 through 1.8). State-of-the-art methods from computer vision and pattern recognition failed, but a convolutional neural network (CNN), a model from the realm of deep learning, proved to be an effective alternative [3]. The resulting classification model yielded an accuracy of around 99% accuracy [3], the architecture of which is depicted in figure 1.13.

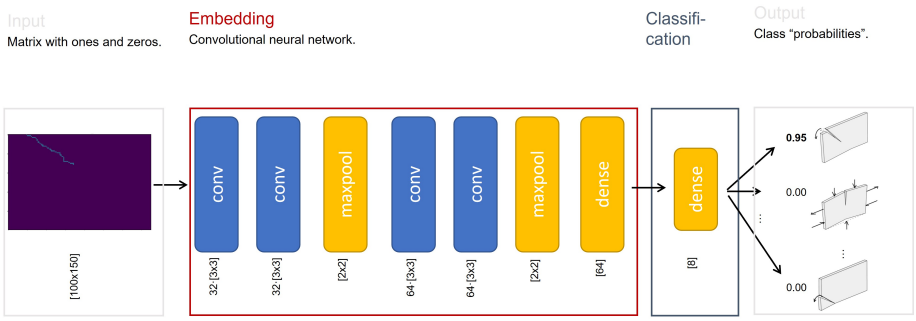

Figure 1.13: The architecture of the Neural Network is depicted. The used activation function is the Rectified Linear Unit (ReLU)

In addition to classifying the cracked facades into one of the eight crack archetypes, the masonry experts decided that it would be instructive to determine the degree of similarity between pairs of cracked facades. For this purpose the output of the second last (dense) layer (64-dimensional) was used to determine the similarity between cracked facades.

However, one major problem arose. Since, The convolutional network is designed to perform a single task, that is to classify the different cracked facades into the correct archetype class, it was entirely optimized to do just that, and to that end, the network is optimized to cluster samples belonging to the same class together. As a result, pairs of cracked facades that belong to different crack archetypes were mapped to points in the 64-dimensional space that were quite far apart, while pairs from the same crack archetype were invariably mapped to

points that were very close in 64-dimensional space. Thus, while the network accurately classified cracked facades into their corresponding cause classes, the embeddings from its second-last layer proved to be of little use for the purpose of predicting the degree of similarity between pairs of cracked facades. [5]

In order to accurately regress the degree of similarity between pairs of cracked facades, a siamese network was built. Unlike the classification network, this network was specifically designed to be optimized for the task of regressing the degree of similarity between pairs of cracked facades. This is accomplished by updating its weight in order to promote the convergence of its predictions to the actual similarity scores using gradient descent and the mean squared error loss to guide the optimization (to be discussed later). Its architecture is illustrated in Figure 1.14.

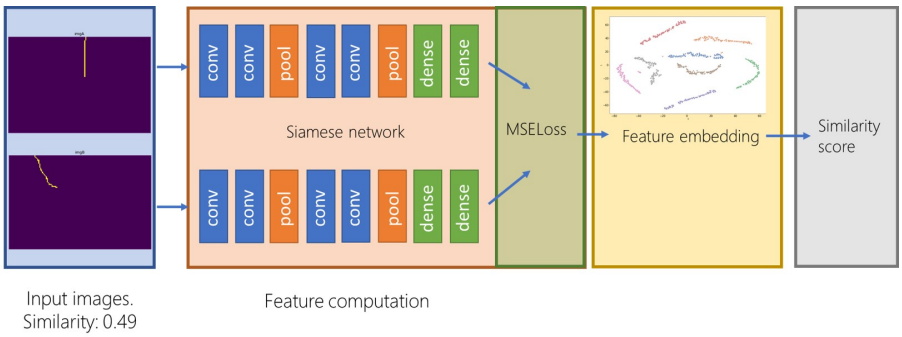

Figure 1.14: The architecture is similar to that of the classification network shown in Figure 1.13. Note that the parallel sets of layers represent the same network. Furthermore, the second-last (dense) layer has size 2048, while the last layer has 64 neurons.

1.5 Research Questions

What I have aimed to achieve specifically during my internship is the automation of the assessment of the degree of similarity between pairs of cracked facades as given by masonry experts, which is described as a task of high practical relevance in [5]. This would allow for a more in-depth assessment of the causes of observed damage in masonry structures. To achieve this goal, three considerations should be kept in mind, which form the basis of the research questions addressed in this work

- We need a program to automate the assessments. For this purpose we use a siamese neural network. To this end, I use the same network architecture that was proposed and implemented by Wyke Pereboom-Huizinga and Maarten Kruithof [5].

- Ideally, the siamese network should accuractely predict the similarity even for pairs of cracked facades that are somehow different from those that it encounters during the training phase. This deserves special attention since there is a huge variety of different types of cracks such that we cannot expect to have all of those represented in our dataset. To this end, we should prioritize experiments which serve to assess the model's ability to generalize across different types of cracked facades.
- In order to learn the expert assessment of the degree of similarity between pairs of cracked facades by means of a neural network, it is important to determine how consistent these assessments are w.r.t. one another before we use them to fit a neural network.

As stated previously, a shortage of real-world images of cracks has necessitated the acquisition of synthetic cracks. Bearing in mind that the ultimate goal remains the automation of the assessment of real-world cracks, we strive to develop automated methods that can learn to assess the similarity of pairs of real-world cracks in accordance with masonry experts, and since the Markov-Walk simulation model is quite simplistic, taking only a handful of parameters, the decision was made to develop a second, more fine-grained crack simulation model. This has prompted the development of the so-called finite element model (FEM). As a result, three data-sets have been amassed, one for each of the two, and one that mixes data from both, and this work considers all three of them. Therefore, many experiments and analyses described herein are applied in a similar fashion to the three different data-sets. In this work I strived to address several research questions. These questions roughly revolve around two themes:

- How well can the model regress the similarity between pairs of cracked facades as determined by masonry experts?
- To what extent do masonry experts agree in their assessment of cracked masonry facades?

The specific Research Questions that I aim to address in this work are outlined below:

RQ 1: To what degree do masonry experts agree in their assessments on the degree of similarity of pairs of cracked masonry facades?

RQ 2: On which samples do the experts disagree most? More specifically, I strive to pinpoint characteristics of samples on which experts disagree in their assessment.

RQ 3: How well can a Siamese Neural Network learn the similarity as assessed by masonry experts?

RQ 4: How well can a Siamese Neural Network

- ***learn the degree of similarity between pairs of cracked facades as determined by experts***
- ***generalize to samples that contain crack archetypes unseen in training***
- ***generalize to samples with similarity scores that do not occur in the training set***

1.6 Approach

In this section, a brief description is provided of the approach that I will take in order to address the research questions that were outlined in the previous section. A more extensive account is given in the Inter-Rater Reliability and the Neural Network Performance and Generalizability chapters.

1.6.1 Inter-Rater Reliability

To determine the extent to which the experts agree on their assessment of similarity, I performed an extensive analysis of the inter-rater reliability, which comprises several subanalyses. Each of these will be discussed in depth in the Inter-Rater Reliability chapter. A brief summary is shown below:

- **Rater Biases**:
  For each of the masonry experts involved in the labelling task, the mean of all the ratings that he/she provided is computed, along with the standard deviations.
- **Relative Rater Biases**:
  For all pairs of raters that have performed the labelling task, the difference between their biases is analyzed.
- **Correlations between Questions**:
  The labelling task consists of three different questions. The extent to which the ratings given for the questions are correlated is measured per rater
- **Overall Agreement**:
  The overall agreement among the raters is quantified using various statistics.
- **Agreement Distributions**:
  The distribution of the agreement as measured between pairs of raters is computed and visualized.
- **Agreement per Sample**:
  An account is given of the agreement per sample. Particularly, specific samples with low agreement are identified and presented.

Furthermore, in our analysis we distinguish between three categories of raters, roughly based on their level of expertise and experience. We distinguish between experts who hold a Ph.D. and have extensive experience, Ph.D. candidates and Master Students.

1.6.2 Neural Network Performance and Generalizability

In order to address RQ 3 and RQ 4, as outlined in the previous section, I will use the same Siamese Network that was built by Dr. Kruithof and Dr. Pereboom-Huizinga. In order to gauge how well the model performs on the given data, I apply a 75/%. 25% random train/test split.

In order to assess the model's ability to generalize across different types of samples, that is to address RQ4, I conducted two experiments:

1. **The Leave One Crack Out Experiment**
   All samples that consist of one or two cracked facades belonging to the crack archetype to be left out, are assigned to the test set, and all other samples to the training set.

2. **The Leave One Class Out Experiment**
   All samples belonging to the leave out class are assigned to the test set, all other samples are assigned to the training set.

The overarching aim of the two leave one out experiments is to determine if, and to what extent, the logic underlying the assessment of the similarity between pairs of cracked facades is universal across different crack archetypes as well as different degrees of similarity. In other words, is the task of determining the degree of similarity between cracks of archetypes A and B similar to determining the similarity between cracks of archetypes Y and B? And what about determining the similarity between highly similar cracks compared to determining the similarity between highly dissimilar cracks? This is the question that we strive to address by means of these experiments.

1.7 Outline

The remainder of this work is organized as follows. Chapter 2 provides a discussion of the related academic literature. Chapter 3 covers the Inter-rater Reliability, in which the three datasets will be described in fine detail. Chapter 4 covers the neural network experiments, including an analysis of the performance of the neural network when using a random-split, as well as some specific experiments that help to gauge the model's ability to generalize across different types of cracks. Chapter 5 covers suggestions for future work and conclusions, including a review of the research questions in light of the results that we obtained. The bibliography followed by the appendix can be found at the end of this thesis.

Chapter 2

Background

2.1 Introduction

In this section, I shed light on some of the relevant academic literature. Since my work involves both a deep learning and artificial intelligence component as well as a statistics component, relevant writings that are related to this thesis from both domains will be discussed in the following.

2.2 Neural Networks and Performance Evaluation

In [6], the authors demonstrate that different performance measures can yield different results in model selection of artificial neural networks. A model that optimizes the percentage of correctly classified samples, may not yield the lowest root mean squared error (RMSE) or the lowest mean absolute error (MAE). This is demonstrated through the use of histograms to illustrate how frequently errors of certain magnitude were observed in prediction. The reader is essentially warned that different performance measures can give rise to different evaluations of Neural Network Performance.

Early efforts by TNO to partially automate the assessment of damage in masonry structures are described in [5]. The classification network depicted in Figure 1.13 was introduced in this work. While the model yielded an accuracy in excess of 99%, a very good result, the proposed method in order to compute the degree of similarity between pairs of cracks, namely to calculate the distance between their 64-dimensional embeddings (from second layer) proved to be flawed. More specifically, within-class similarities were not properly predicted by this method. All in all, this work marks a promising first step towards the automation of damage assessment in masonry structures.

In [7] the notion of similarity between neural network representations is discussed. In other words, how similar are the layers of different neural networks, or even subsequent layers of the same deep neural network? The authors propose a central kernel alignment (CKA) as a method to measure similarity between neural network representations, and compare it to more popular methods for measuring such similarity, including canonical correlation analysis (CCA).

In [8], the authors seek to explain how Neural Networks tend to be able to generalize well to different samples unseen in training. More specifically, the authors aim to empirically assess which complexity measure can best explain differences in the ability to generalize to different samples between different neural network models, be it models with different architectures or models with the same architecture but with different weight configurations. The experiments show that a combination of sharpness coupled with a norm-based measure best explains the extent to which a model can generalize.

In [9], the authors apply one shot learning for character recognition. Rather than having a predefined set of classes to compare new and unseen samples to, the authors strive to learn a function through the use of a siamese network that can determine whether two images contain the same character. Their method is trained and tested on pairs of characters from different alphabets. Their model differs from ours in that the output is binary, while in our case the model outputs a continuous value in the real-valued interval $[0, 1]$

2.3 Correlation Coeffieicnts and Inter-rater Reliability

This section highlights some of the relevant literature regarding the domain of statistics that is concerned with measuring the degree of correlation or agreement among raters as well as some of my considerations regarding the herein presented literature.

2.3.1 Difference between Agreement and Association

Throughout the twentieth and late nineteenth centuries, statisticians have developed numerous statistical tools and methods to measure the degree of association between variables of interest, test reliability and validity, as well as inter-rater reliability, often referred to as agreement. A clear distinction is made between coefficients which are appropriate to establish whether a correlation exists between a set of variables, and coefficients which can be used to measure the amount of agreement between sets of ratings.

To illustrate this difference, suppose we have the following pair of sets of ratings:

1. $[1, 2, 3, 4, 5]$
2. $[3, 4, 5, 6, 7]$

Clearly, the corresponding data points of both lists suggest that a linear relationship exists, nonetheless, the agreement is quite low, since the ratings disagree across all five samples by a constant factor of 2. Some coefficients mentioned in the following measure the degree of correlation while others measure the rate of agreement.

2.3.2 History of Correlation Coefficients

In this subsection, a brief overview of the history of the development of statistics and correlation coefficients is given.

Early Methods for Measuring Association

Throughout the twentieth and late nineteenth centuries, statisticians have developed numerous statistical tools and methods to measure the degree of correlation between variables of interest, test reliability and validity, as well as inter-rater reliability, often referred to as agreement. Early methods used to measure agreement include Pearson's R and percent agreement. These measures, however, are of limited value, Pearson's R could only establish the degree to which ratings of two different raters are linearly associated, while percent agreement does not account for the possibility of chance agreement and does not account for ordinality of the label categories [10] [11].

First Statistic for Measuring Agreement: Kappa

In 1960, Jacob Cohen famously introduced Cohen's Kappa statistic as a means to measuring agreement among pairs of raters for nominal scales **[11]**. Suppose that 2 raters assign one of k nominal classes to N samples. Cohen's Kappa can be used to determine the degree to which the raters agree, on a scale from -1 to 1, where $k = -1$ signifies complete disagreement, $k = 0$ signifies the amount of agreement one would get if all labels were assigned randomly, and $k = 1$ signifies perfect agreement.

Weighted Kappa

Several years later, Cohen proposed a weighted version of the kappa statistic which he had proposed earlier, which marked an early step towards the development of coefficients that could be used to measure the extent of agreement between pairs of ratings on ordinal scales (i.e. likert scales). Nowadays, several variants of the weighted kappa exist, of which the most commonly used in the literature are the linearly weighted kappa and the quadratically weighted kappa. As the names suggest a difference of 2 is weighted as 2 and $2^2 = 4$ in linearly weighted kappa and quadratically weighted kappa, respectively.

Other Coefficients and Statistics

Several other coefficients have been proposed throughout the late twentieth century, including Lin's Concordance Correlation Coefficient [12] which applies to both ordinal and continuous variables. Various variants of most of the proposed coefficients exist, some apply to pairs of raters while others can be aplied to more than two raters, such as Fleiss Kappa[13], the Intraclass Correlation Coefficient (ICC) which was first proposed by R.A. Fisher in one of the most pivotal writings on statistical methods [14] and later proposed as measure of reliability by Bartko [15], and a variant of Lin's Concordance Correlation Coefficient which applies to more than two raters and was proposed by Barnhart [16]. Finally, Klaus Krippendorff proposed the Krippendorff's Alpha [17], which provides and overall estimate of the agreement among multiple raters in the case that n raters assign one of k classes to N samples, and not all raters have rated all samples, or stated differently, some values are missing. [17]

2.3.3 Interpretation of Coefficients and Statistics

This section sheds light on some of the proposed criteria for interpreting the strength of correlation as expressed by the aforementioned coefficients and statistics.

Pearson's R and Spearman's Rho

One can verify that Pearson's R will only take values 1 or -1 in the case that the two variables are perfectly linearly correlated or perfectly inversely linearly correlated, respectively. The stronger the evidence that a linear correlation exists, the closer Pearson's R and Spearman's Rho will be to 1 or -1. Guidelines on how to interpret the strength of the linear relationship between two variables as measured by the two have been suggested by Chan [18] and by Dancey and Reidy [19]. Chan suggested that $\rho > 0.8$, $0.6 < \rho < 0.8$, $0.3 < \rho < 0.5$, and $\rho < 0.3$ shall be interpreted as strong, moderately strong, fair and poor linear correlation, respectively.

Kappa Statistics

Early guidelines for Cohen's unweighted kappa suggested that agreements within the ranges $0.61 - 0.80$ and $0.81 - 1.00$ should be viewed "substantial agreement" and almost "perfect agreement", respectively.

A different guideline for interpreting both the Kappa statistic as well as the intraclass correlation coefficient was proposed later by Domenic V. Cicchetti [20]. According to Cicchetti a Kappa of $0.50 - 0.75$ should be interpreted as moderate agreement, $0.75 - 0.90$ as good agreement, and $\kappa > 0.9$ as excellent agreement.

According to Fleiss et al [21], the interpretation of the magnitude of agreement as expressed by weighted kappa is similar to that of the unweighted kappa, $k > 0.75$ signifies excellent agreement. This should, however, be taken with caution, since it has been shown empirically that quadratically weighted kappa tends to yield a higher value than linearly weighted kappa, which in turn tends to yield a higher value than unweighted kappa [10], that is $\kappa_2 > \kappa_1 > \kappa_0$. Moreover, as stated previously, the original guideline for interpreting Cohen's Kappa statistic has been criticized, and more stringent criteria have been suggested for its interpretation [22].

ICC

Various guidelines on the interpretation of the ICC have been suggested, but the most widely accepted include Cicchetti's guideline [20], and the guideline proposed by Li and Koo. [23]

Lin's CCC

The most widely adopted criteria for the interpretation of Lin's CCC were proposed in 2005 by Mc.Bride. [24]. Mc.Bride's criteria suggest that $0.90 \leq CCC \leq 0.95$ signifies moderate agreement, $0.95 \leq CCC \leq 0.99$ substantial agreement and $CCC \geq 0.99$ signifies excellent agreement. Therefore, I will henceforth interpret $CCC < 0.90$ as evidence of poor agreement.

Krippendorff's Alpha

Regarding the interpretation of the Krippendorff's Alpha, $0.67 \leq \alpha \leq 0.80$ is considered as somewhat acceptable agreement, while $0.80 \geq \alpha$ is generally considered to be indicative of a good degree of agreement.

2.3.4 Choice of Agreement Statistics

This section sheds light on considerations and criteria regarding which statistic to use in order to measure agreement or association.

When one conducts an extensive review of the existing scientific literature, one will likely conclude that there is no clear consensus on how to precisely interpret statistics and coefficients, nor a guideline on which method to choose for a specific problem and dataset, and while some studies highlight that different statistics yield similar interpretations of the degree of inter-rater reliability [25], other studies reveal that the use of different statistics lead to different results. [26]. Based on this observation, I have decided to use several statistics in order to gauge the inter-rater reliability of our datasets.

Nonetheless, there are some criteria based on which researchers can make a somewhat informed decision regarding which statistics to use to conduct their analysis. Firstly, one should determine whether the degree of association or

the degree of agreement is to be measured. In the former case, Pearson's R, Kendall's Tau and Spearman's Rho are probably more suitable, whereas in the latter case, the Kappa statistics, Lin's CCC, and the ICC are more appropriate. Furthermore, the nature of the data should also guide the decision-making, if the data is nominal, the kappa statistics are useful, while in the case of ordinal data, the weighted kappa as well as Lin's CCC and ICC are useful, and in the case of continuous data, Lin's CCC and the ICC are recommended. Lastly, Krippendorff's Alpha is useful when dealing with missing data.

2.4 Statistics and Coefficients

This section provides an intuition for each of the statistics and coefficients I have used throughout my Inter-Rater Reliability Analysis. Each of the used statistics will be discussed extensively in the following, and in some cases, examples will be given of how a statistic is used to compute the rate of agreement or correlation, including some examples for which certain statistics yield counter-intuitive results. The purpose of this section is to familiarise the reader with the statistics that I have used, and to provide an intuition for these statistics.

- ***Proportionate Agreement (Percent Agreement)***
- ***Cohen's Kappa Statistic***
- ***Average Difference.***
- ***Mean Difference***
- ***Lin's Concordance Correlation Coefficient***
- ***Krippendorff's Alpha***
- ***Kendall's Tau***
- ***Pearson's R***

In the following, each of the eight statistical measures used to measure agreement will be described.

- **Proportionate Agreement (Percent Agreement):**
  Given two raters who each rated the same set of N samples, in what fraction of their ratings do they agree?

  Example:
  Let P_0 denote the proportionate agreement. Suppose, Jack and Tim have rated 5 items in the same order, assigning a value ranging from 1 to 5 to each item. And suppose $\{1, 2, 1, 4, 5\}$ and $\{1, 3, 2, 5, 5\}$ represent the lists of their ratings. Regardless of which list corresponds to whom, the proportionate agreement is $P_0 = .4$, because their ratings match for item 1 and 5. Thus, $P_0 = 2/5 = .4$

Extension to Multiple Raters:
Note that this measure can be easily adapted to the case where n raters ($n \geq 3$) have rated N samples, including the case where each of the n raters has only rated a subset of the N samples. One can simply compute the set of commonly rated samples for all pairs of raters, and scan through the lists of commonly rated samples while keeping counters for both the number of observed agreements as well as the total number of comparisons. The proportionate agreement will simply be the number of observed agreements divided by the total.

- **Cohen's Kappa Statistic:**
Cohen's Kappa Statistic is a measure which reflects the degree of agreement among two raters who have rated the same set of items. Cohen's Kappa differs from proportionate agreement in that it accounts for the possibility of chance agreement. That is, two raters who rate a set of items randomly will likely agree in some of their ratings by chance. The value of this measure ranges from -1 to 1, with values -1, 0 and 1 reflecting, complete disagreement, the amount of agreement that purely results from chance, and complete agreement, respectively. It is important to note that the original Cohen's Kappa is insensitive to whether a nominal, ordinal or interval scale is used. It is given by the following:

$$k = \frac{P_o - P_e}{1 - P_e}$$

where P_o denotes the proportionate agreement as defined above, and P_e denotes the chance agreement. In this context, P_e is computed as follows:

$$P_e = \frac{1}{N^2} \cdot \sum_k n_{k_1} n_{k_2}$$

where N denotes the number of items that were rated by both raters, k denotes the category and iterates over all possible categories, and n_{k_i} denotes the number of times that rater i rated an item as belonging to the k'th category.

Example:
Suppose, Jack and Tim assign a score to a set of four items. To each item, they can assign either 1, 2 or 3. Now suppose they have rated the items in the same order and their ratings can be represented as {1, 2, 2, 3} and {1, 1, 2, 3}. Let P_1, P_2 and P_3 denote the chance probabilities for classes

(scores) 1, 2 and 3, respectively. Then, $P_1 = \frac{1}{4} \cdot \frac{2}{4} = \frac{1}{8}$, $P_2 = \frac{2}{4} \cdot \frac{1}{4} = \frac{1}{8}$, and $P_3 = \frac{1}{4} \cdot \frac{1}{4} = \frac{1}{16}$, thus $P_e = P_1 + P_2 + P_3 = \frac{5}{16}$. $P_o = \frac{3}{4}$. The value of κ is given by:

$$k = \frac{P_o - P_e}{1 - P_e} = \frac{\frac{3}{4} - \frac{5}{16}}{1 - \frac{5}{16}} = 0.636$$

- **Average Difference:**
  In this context, the term "Average Difference" refers to the extent to which two given raters disagree, whenever they disagree. Intuitively, when dealing with ordinal labels, it matters not only whether two raters disagree in their assessment of a certain sample, but also by how much. If the first rater labels the featured pair of cracks as Very Dissimilar and the second rater labels it as Very Similar, this indicates that their judgments are far more different than they would have been if the second rater had labelled the pair as Dissimilar. Since the five similarity labels, ranging from Very Dissimilar to Very Similar, are ordinal, we can conveniently represent them as integers from 1 to 5, respectively. This enables one to quantify how much, on average, two given raters disagree whenever their judgments do not match.

  Example:
  Suppose each of two raters label the same set of five items in the same order. Their judgments can be represented as {1, 2, 3, 3, 5} and {2, 2, 3, 4, 1}. Their ratings differ for the first, fourth and last item by 1, 1 and 4, respectively. Thus, the average difference is computed as:

$$\frac{1 + 1 + 4}{3} = 2$$

- **Mean Difference:**
  Unlike the Average Difference, the Mean Difference is the mean of the sum of all the differences between the ratings provided by two raters for each sample.

  Example:
  To illustrate the difference between the two measures, suppose again that two raters have provided the following lists of ratings {1, 2, 3, 3, 5} and {2, 2, 3, 4, 1}. The mean difference is computed as:

$$\frac{(2-1) + (2-2) + (3-3) + (4-3) + (5-1)}{5} = 6/5 = 1.2$$

- **Lin's Concordance Correlation Coefficient:**
  Lin's CCC was first proposed by Lin [12]. It is based on Pearson's Correlation Coefficient which was introduced in the late 19'th century[27]. It

offers advantages in estimating the agreement over Pearson's R in that it not only measures whether a linear relation exists between two variables, but also the deviation from this fitted line, which makes it more suitable for measuring agreement.

Lin's CCC for pairs of ratings is given by the following formulae:

$$\hat{\rho}_c = \frac{2s_{xy}}{s_x^2 + s_y^2 + (\bar{x} - \bar{y})^2}$$

$$\textit{where } s_x^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})^2 \textit{ and } s_{xy} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(y_n - \bar{y})$$

Examples:

1) $\begin{Bmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 & 7 \end{Bmatrix} \rightarrow \hat{\rho}_c = 0.5$ **2)** $\begin{Bmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 \end{Bmatrix} \rightarrow \hat{\rho}_c = -1$

3) $\begin{Bmatrix} 2 & 2 & 2 & 3 \\ 2 & 1 & 2 & 3 \end{Bmatrix} \rightarrow \hat{\rho}_c = 0.667$ **4)** $\begin{Bmatrix} 2 & 1 & 2 & 3 \\ 2 & 1 & 2 & 3 \end{Bmatrix} \rightarrow \hat{\rho}_c = 1.0$

5) $\begin{Bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 7.5 \end{Bmatrix} \rightarrow \hat{\rho}_c = 0.124$ **6)** $\begin{Bmatrix} 1 & 2 & 3 & 4 \\ 1 & 8 & 27 & 64 \end{Bmatrix} \rightarrow \hat{\rho}_c = 0.047$

7) $\begin{Bmatrix} 1 & 3 & 5 & 7 & 9 \\ 2 & 4 & 6 & 8 & 10 \end{Bmatrix} \rightarrow \hat{\rho}_c = 0.941$ **8)** $\begin{Bmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 1 & 2 \end{Bmatrix} \rightarrow \hat{\rho}_c = 0$

9) $\begin{Bmatrix} 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \end{Bmatrix} \rightarrow \hat{\rho}_c = NaN$

Lin's CCC yields a much lower value than do Kendall's τ and Pearson's R for examples 1, 5 and 6. This is unsurprising in the sense that Lin's CCC is designed to measure agreement rather than correlation. However, a close relation yet exists between Lin's CCC and the two coefficients. This relation is somewhat reflected by example 1, but even more so by example 7. In example 7, the observations can be fitted perfectly by a line, and Lin's CCC is very high while the values of both sets of ratings do not agree in any position of the lists. Furthermore, it is remarkable that Lin's CCC suggests that the ratings presented in example 7 are more in agreement than those presented in example 3, which is quite counter-intuitive. Clearly, the observations from example 3 can not be fitted by a line, which provides counter-evidence for the existence of a linear relationship between the variables. Therefore, examples 3 and 7 further highlight

the association between Lin's CCC and Pearson's R. Additionally, one can observe that Lin's CCC equals 0 in the case that exactly one of the sets of ratings contains only a single value (Example 8), because that implies that either $x_n = \bar{x}$ or $y_n = \bar{y}$ which implies that $(x_n - \bar{x})(y_n - \bar{y}) = 0$. Note that both Pearson's R and Kendall's τ are undefined for example 8. And finally, Lin's CCC is undefined in the case that both sets of ratings are equal and contain repetitions of the same value (Example 9).

- **Pearson's R:**
  As mentioned in the Literature Review of this thesis, Pearson's R can be used to determine whether two variables are linearly correlated. Only in cases for which a perfect positive or negative correlation exists between two sets of ratings, Pearson's R will be 1 and -1, respectively. However, it is important to observe that Pearson's R tends to be high in cases where a positive non-linear relation exists between two variables.

  Pearson's R is given by the following formulae:

$$\frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})/(n-1)}{s(x)s(y)}$$

Examples:

1) $\begin{Bmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 & 7 \end{Bmatrix} \rightarrow P_r = 1$ **2)** $\begin{Bmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 \end{Bmatrix} \rightarrow P_r = -1$

3) $\begin{Bmatrix} 2 & 2 & 2 & 3 \\ 2 & 1 & 2 & 3 \end{Bmatrix} \rightarrow P_r = 0.791$ **4)** $\begin{Bmatrix} 2 & 1 & 2 & 3 \\ 2 & 1 & 2 & 3 \end{Bmatrix} \rightarrow P_r = 1$

5) $\begin{Bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 7.5 \end{Bmatrix} \rightarrow P_r = 0.990$ **6)** $\begin{Bmatrix} 1 & 2 & 3 & 4 \\ 1 & 8 & 27 & 64 \end{Bmatrix} \rightarrow P_r = 0.943$

7) $\begin{Bmatrix} 2 & 2 & 1 & 2 \\ 2 & 2 & 2 & 2 \end{Bmatrix} \rightarrow P_r = NaN$

One can observe that when one can fit a line through all the data points, $P_r = 1 \vee P_r = -1$ depending on whether the two variables are positively or negatively correlated (See examples 1, 2 and 4). When a datapoint that deviates from the fitted line is added, P_r decreases (See examples 3 and 5). Depending on whether the anomalous datapoint merely countersuggests the existence of a linear relationship between the two variables (See example 5) or that it even suggests that they are not correlated at all (example 3), the decrease in P_r will be minimal or substantial, respectively.

Example 6 demonstrates that two variables that are perfectly exponentially related will still yield a high P_r, therefore one must be cautious to conclude that a linear relationship exists in case of a P_r that is close to 1 or -1, although it does provide strong evidence that the two are somehow correlated. Pearson's R is undefined for example 7, because one of the sets of ratings contains only one value, which implies that the standard deviation for the corresponding variable is 0, which results in a division by zero.

- **Kendall's Tau:**
  Kendall's Tau (equivalently τ) only considers the relative frequencies of concordant and discordant pairs of data points. A concordant pair is a pair of observations $(x_i.y_i)$ and $(x_j.y_j)$ such that either $x_i < x_j \wedge y_i < y_j$ or $x_i > x_j \wedge y_i > y_j$. By contrast, a discordant pair is a pair of observations $(x_i.y_i)$ and $(x_j.y_j)$ such that either $x_i < x_j \wedge y_i > y_j$ or $x_i > x_j \wedge y_i < y_j$. Similarly to most other coefficients, $-1 \leq \tau \leq 1$. The more concordant pairs, the higher the value of τ and the more discordant pairs, the lower the value of τ.

Kendall's τ is given by:

$$\tau = \frac{C - D}{\frac{1}{2}n(n-1)}$$

Where, C denotes the number of concordant pairs, and D denotes the number of discordant pairs.

Examples:

1) $\begin{Bmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 & 7 \end{Bmatrix} \rightarrow P_r = 1$ **2)** $\begin{Bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 8 & 27 & 64 & 125 \end{Bmatrix} \rightarrow P_r = 1$

3) $\begin{Bmatrix} 2 & 2 & 2 & 3 \\ 2 & 1 & 2 & 3 \end{Bmatrix} \rightarrow P_r = 0.775$ **4)** $\begin{Bmatrix} 2 & 1 & 2 & 3 \\ 2 & 1 & 2 & 3 \end{Bmatrix} \rightarrow P_r = 1.0$

5) $\begin{Bmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 1 & 2 \end{Bmatrix} \rightarrow P_r = NaN$ **6)** $\begin{Bmatrix} 5 & 4 & 3 & 2 \\ 5 & 6 & 7 & 7.5 \end{Bmatrix} \rightarrow P_r = -1.0$

While Kendall's τ is generally believed to be similar to Pearson's R, the examples shown above clearly reflect some key differences. So long as the two variables are either perfectly correlated or inversely correlated $\tau = 1 \vee \tau = -1$. That is, if all pairs of datapoints are concordant or all

are discordant, $\tau = 1$ and $\tau = -1$, respectively. similarly to what we observed for Pearson's R, if one of the ratings contains only one value, τ is undefined.

- **Krippendorff's Alpha:**
  Lastly, we consider the Krippendorff's Alpha (equivalently α), introduced by Klaus Krippendorff [17]. Of all the statistics presented in this section so far, the Krippendorff's Alpha along with the Percent Agreement are the only statistics that can be used directly in order to measure the agreement across entire datasets. Krippendorff's α is given by the following [28]:

$$\alpha = 1 - \frac{D_o}{D_e} = \frac{1 - \sum_{v=1,v\prime=1}^{V} o_{vv'}\,\delta(v, v^{'})}{1 - \sum_{v=1,v\prime=1}^{V} e_{vv'}\,\delta(v, v')}$$

$$where\ o_{vv'} = \sum_{u=1}^{N} \frac{\sum_{i\neq i'}^{m} I(v_{iu} = v) \times I(v_{i'u} = v^{'})}{m_u - 1} = o_{v'v}$$

$$and\ e_{vv'} = \frac{\sum_{i\neq i'}^{m} I(v_{iu} = v) \times I(v_{i'u} = v^{'})}{n - 1}$$

where v iterates over the values (i.e. label categories, 1, 2, 3 and 4 in our case), and u iterates over the so-called **units**, and in this context, unit means the same as sample. Note that $\delta(v, v^{'})$ depends on the scale that is used, which can be **nominal**, **ordinal** or **continuous**. Furthermore, $o_{vv'}$ aims to quantify how often a combination of ratings as provided by a pair of raters is observed. For instance $o_{1,2}$ quantifies how often it has occurred that rater assigned label 1 to a sample, while rater 2 assigned label 2 to that same sample, over all pairs of raters. $e_{1,2}$ is an estimate of the probability that the first rater assigns label 1 and the second rater assigns label 2, based on how often this pair of values occurs across the dataset. One could say that in that regard, the Krippendorff's α is at least partly inspired by Cohen's κ. For further reading I recommend [17] and [28].

Chapter 3

Interrater Reliability

3.1 Motivation

In this section, we shed light on the variability among assessments of the degree of similarity between pairs of cracked facades, as provided by masonry experts during the labelling task. Earlier discussions on the assessment of damage in masonry structures have highlighted the complexity of this task [29]. Additionally, those whose research relates to masonry structures mostly agree that there is a wide variety of potential causes of damage to these settlements [30]. Furthermore, it has been suggested that crack assessment can be somewhat subjective, which implies that significant differences between assessments provided by different masonry experts exist.

It is important to note that the automation of similarity assessment of pairs of images of cracked facades requires a neural network to be trained on similarity assessments as provided by masonry experts, and that a low rate of agreement among these masonry experts will most likely result in a poor level of performance of the neural network. Krishna Ajithkumar Pillai provided evidence for this in her Master Thesis [4]. Her results show that the degree to which experts agree in their assessments, as measured by Krippendorff's Alpha, seems to be correlated with the predictive power of the neural network.

It stands to reason that in order to successfully automate the task of assessing the degree of similarity between pairs of cracked facades, those who provide the similarity assessments of the data used to train the neural network should strongly agree. Even if a fitted siamese network yields a decent performance on assessments provided by masonry experts who exhibit poor inter-rater agreement, it is questionable whether the network has learnt anything that will be useful to a masonry expert. First of all, since we do not know what the network has exactly learnt in such a case. The learnt function will be based on some average of a set of divergent ratings provided by multiple masonry experts who apparently hold different views on the concept of similarity in the context of

cracked masonry facades. And secondly, since masonry experts' opinions are quite divergent, it is doubtful whether such an automated tool will be useful to said experts, since any assessments or views provided by the tool might strongly conflict with their own assessments and views, and therefore cause confusion rather than aid them in their decision-making.

This motivates efforts to not only determine the extent to which masonry experts agree on the assessment of the similarity of pairs of cracked masonry facades, but also to investigate the nature of the disagreement that we observe in the data in order to better understand the causes thereof, and hopefully what could be done to improve the rate of agreement.

Hence, I strive to address the first research question, RQ1, by means of this extensive analysis. Additionally, to further enhance our understanding of the variability among raters, we find it instructive to investigate for which types of samples raters tend to disagree more strongly. Thus, this analysis also addresses the second research question, RQ2.

Additionally, we will investigate the degree of correlation between sets of ratings provided for each of the three similarity questions by single raters, in order to verify whether correlations exist between the similarity questions. The three questions will be described in the following section.

3.2 Scope of Statistical Analysis

It is important to note that this discussion is limited to similarity assessments performed on pairs of synthetically simulated cracked facades that are modelled based on the crack archetypes that were mentioned earlier in this work. This work reports on the interrater variability as measured on three datasets (D_1, D_2 and D_3), henceforth referred to as the **Markov-Walk Dataset**, the **Mixed Dataset** and the **FEM Dataset**. Each dataset will be described in more detail in the following section (3.3).

The labelling task consists of three questions. I have decided to perform separate statistical analyses for the annotations provided for each of the three questions. This granular approach will allow us to determine the degree of agreement for each of the questions separately, and will also allow for maximal flexibility in deciding how to combine these three degrees of agreement into one, should we deem it appropriate to combine them into a single measure. Questions are shown below in the order in which they were shown to the participants during the labelling task. Note that a brief account of how the three questions are related, and why they were formulated as such is postponed to section 3.7 (on Q-to-Q Correlations).

1. ***How similar are the settlement damage cause and the pattern of these two crack patterns?***
2. ***How similar is the severity of damage in both of the crack patterns?***
3. ***Overall, how similar are the cracks?***

For each of the three questions, each of which concerns a comparison between two synthetic images of crack patterns, the raters assigned one of the following five labels: ***Very Dissimilar***, ***Dissimilar***, ***Similar***, ***Very Similar*** and ***I Cannot Say***

Note: *The "I Cannot Say" label only occurs in the Markov-Walk Dataset, and is viewed as an "outside category" rather than it being on the likert scale from very dissimilar to very similar.*

Note: *For the Markov-Walk dataset, we excluded all samples that were rated as "I Cannot Say" at least once (for any of the three questions).*

3.3 Datasets

In this section I explain how each of the three data-sets were obtained, and provide insight into their underlying structure. The identities of those who were involved in rating the similarity of the samples will not be mentioned here, for privacy reasons.

In total, 28 raters were involved in labelling the samples of the three datasets. Based on their level of expertise and experience, we divided the 28 raters into three categories, each of which will be described below.

- ***MSc.*** *Is currently enrolled in a Master's program at the faculty of Architecture and the Built Environment at the TU Delft.*
- ***Ph.D.*** *Had been, or is currently a Ph.D. candidate in a topic which is very closely related to the assessment of masonry strucures.*
- ***Expert*** *Has extensive experience in the assessment of damaged masonry structures, and holds a Ph.D. Many of the raters in this category are employed as researchers in the field of structural engineering, either in industry or in academics*

3.3.1 Markov Walk Dataset

This dataset consists of images that were generated using a Markov Walk Model developed and programmed by Dr. Arpad Rozsas in the popular programming language Python3. Each of the crack types that can be generated with the data simulation model are represented in the dataset [5]. As mentioned before the included crack archetypes are 18, 20, 21, 23, 24, 30, 31, 32, 101, 102, 103 and 201 as described by Ilse de Vent in her Ph.D. thesis [2]. Three examples of samples (pairs of crack patterns) that were rated, are shown in Figures 6.1, 6.2, and 6.3. The raters are indexed from 1 up to and including 28, raters 1 through 7 are experts, raters 8 through 14 belong to the Ph.D. category (described in the previous section) and the remaining raters are Master Students. In total 2968 samples were labelled at least once, 2587 of which were labelled at least three times, and 2466 were labelled three times and were never assigned the label "I cannot say". I have excluded all samples that were rated as "I cannot say", as well as all samples that were rated fewer than three times. There are two reasons for this. First of all, a single rater may accidentally select the wrong degree of similarity for a given sample, and secondly, since raters may disagree, ratings from multiple raters will provide a more reliable estimate. In total, 9372 labelling instances have been completed, 8 of which were duplicates, which yields a total of 9364 labelling instances (the first of each of the 8 duplicates were ignored). The distribution of the labels across each of the three questions is shown in table 3.1 below:

	Label: 1	Label: 2	Label: 3	Label: 4
Q1	1476	2399	2943	2423
Q2	1304	3012	2980	1945
Q3	1294	2989	3156	1802
Total	4074	8400	9079	6170

Table 3.1: Markov-Walk Dataset: Distribution of the ordinal similarity labels across the three different questions

The following pairs of crack archetypes are deemed to be more similar by the masonry experts: **(18, 20), (31, 32), (20, 21), (23, 30), (24, 32), (101, 102), (103, 201)**

The pairs of crack archetypes shown above were also considered as **similar** in the criteria based on which we formed pairs of cracked facades in order to construct the Markov-Walk data-set. The composition of this data-set is as follows:

- *25% of the samples consist of pairs of crack archetypes that are considered to be similar and have similar widths.*
- *20% of the samples consist of pairs of crack archetypes that are considered to be similar and have independently sampled widths (which may or may not be similar).*
- *25% of the samples consist of pairs of cracked facades that belong to the same crack archetypes and have similar widths.*
- *20% of the samples consist of pairs of cracked facades that belong to the same crack archetypes and have independently sampled widths (which may or may not be similar).*
- *10% of the samples consist of pairs of cracked facades that are independently, randomly picked with and have independently sampled widths (which may or may not be similar).*

3.3.2 FEM Dataset

This dataset consists of pairs of images that were generated using a simulation model that is based on Finite Element Analysis. It was implemented in Diana by Krishna Ajithkumar Pillai. In her efforts to develop this model, she was supervised by Giorgia Giardina,Arpad Rozsas and Arthur Slobbe.[4] In total, 7 of all of the 28 raters were involved in labelling samples from this data-set. It consists of 500 samples, and 1522 labelling instances were completed, 5 of which were duplicates. The distribution of the labels across each of the three questions is shown in table 3.2 below:

	Label: 1	Label: 2	Label: 3	Label: 4
Q1	654	232	331	300
Q2	487	422	327	281
Q3	686	311	212	308
Total	1827	965	870	889

Table 3.2: FEM Dataset: Distribution of the ordinal similarity categories across the three different questions

3.3.3 Mixed Dataset

This dataset consists of images from both aforementioned simulation models as well as real-world images. In total 3 of the 28 raters were involved in annotating the image pairs, all of whom are considered to be experts. In total, 150 labelling instances were completed, none of which were duplicates. Moreover, this dataset contains 50 samples in total, and each of the three raters rated all of the 50 samples.

	Label: 1	Label: 2	Label: 3	Label: 4
Q1	39	32	42	37
Q2	22	62	54	12
Q3	40	44	43	23
Total	101	138	139	72

3.4 Approach and Outline

In order to quantify the variability among the assessments provided by the 28 involved raters, and to investigate the nature of observed disagreement, I performed multiple analyses. Each of these will be described briefly in the following

3.4.1 Mean Biases

Separately for each dataset, for each of the three questions, and for each of the raters, I have computed the average of all similarity ratings given, and visualized these results with bar charts. These charts will clearly reflect whether specific raters tend towards one of the extremes of the similarity spectrum (Very Dissimilar and Very Similar) and to what extent.

3.4.2 Relative Biases

Whereas the Mean Biases serve to provide insight into the individual labelling behaviour of each of the raters, the relative biases are the differences in bias between pairs of raters, computed over sets of commonly rated samples.

3.4.3 Overall Agreement

In this analysis, the rate of agreement is measured by means of different statistics and coefficients for complete datasets. Whether thresholds are met that indicate whether the rate of agreement is sufficient will be reflected by using

different colors, green to indicate a satisfying rate of agreement, orange to indicate substantial yet insufficient agreement, and red to indicate poor agreement.

3.4.4 Question to Question Correlations

This section will provide insight into the degree of correlation between ratings given for each of the questions, computed separately for all users. Since the third question asks the rater to estimate the **Overall Similarity** while the first and second question address two different aspects of what could be thought to constitute the notion of similarity, one could argue that the rating provided for question three could represent some weighted average of the ratings provided for the first two questions. Whether the ratings provided by the raters actually support this hypothesis, and the precise relation between ratings provided for the three different questions, is discussed in this section.

3.4.5 Problematic Samples

I have strived to determine which samples, in terms of involved crack archetypes, are subject to disagreement among the raters, and identify what is common among those samples. This section presents my findings with regards to this effort. How I have quantified the rate of disagreement for individual samples is discussed, as well as the types of samples for which the raters exhibit a poor level of agreement that I have identified using this method.

3.4.6 Outline

In the remainder of this chapter, the correspondence of the indexed raters between the three datasets is discussed first, followed by a discussion of the aforementioned (sub)analyses in the order in which they have been presented previously.

3.4.7 Cross Data-Sets Rater Correspondences

In order to respect the privacy of the involved raters, non-suggestive names were used to represent them. Across all three datasets, all raters are referred to as *Rater(i)* for some $i \leq n$, where $n = 28$. However, it is not always the case that *Rater(i)* in one dataset refers to the same rater as *Rater(i)* in one of the other dataset. For instance, Rater5 represents a different person in the Markov-Walk dataset than it does in the FEM dataset. In fact, Rater5 represents an Expert in the Markov-Walk dataset, whereas it represents a Master Student in the FEM dataset, and the mixed dataset has only three raters. The correspondence between rater indices across the three datasets are shown in Figure 3.1.

3.5 Mean Similarity Assessments

In this section the mean of the ratings for each question and for each rater are presented, separately for each of the three datasets. Quite surprisingly, the mean of the ratings provided for each of the three questions vary significantly between raters. For the Markov-Walk dataset, the results are shown separately for MSc. students, PhD. students and expert raters, whereas for both the FEM and Mixed data-sets, the results are shown in single bar charts, since those data-sets involve only 7 and 3 raters, respectively.

3.5.1 Agreement vs. Difference in Bias

While a difference in bias for a particular question among a pair of raters reflects that the rate of agreement among those raters is suboptimal, the reverse does not hold, namely that equal biases among a pair of raters should reflect that they perfectly agree. To see this, consider the following pair of ratings, provided by Rater A and Rater B for the same question and for the same set of samples:

$$RaterA \rightarrow \{1, 5, 1, 5, 1, 5, 1, 5, 1, 5\}$$
$$RaterB \rightarrow \{2, 3, 4, 3, 3, 3, 3, 3, 3, 3\}$$

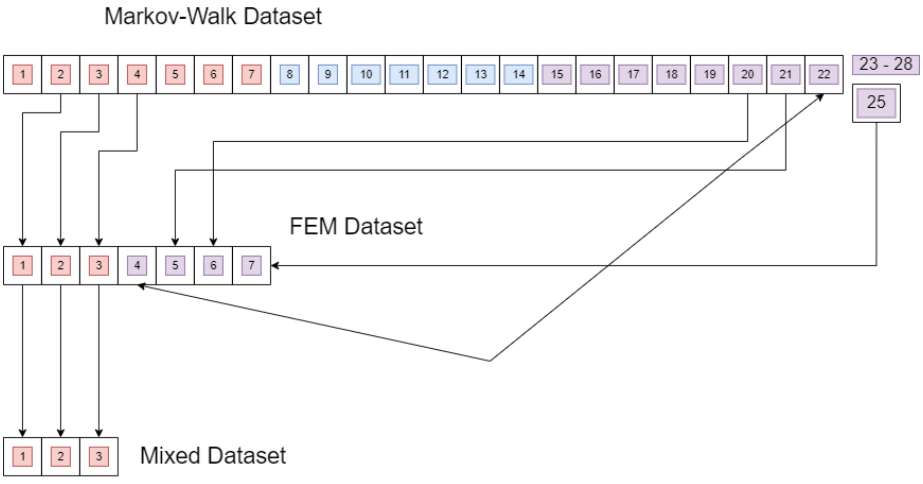

Figure 3.1: This diagram shows the correspondences between the rater indices across the three different datasets.

While both raters have the same bias (same average rating across the sample set), the level of agreement is clearly quite low. One might wonder if considering the rater biases is even useful, in addition to merely computing the amount of

agreement between raters. I deem it relevant to compute the average ratings and differences in average ratings, since these values clearly illustrate whether or not certain Raters are more likely to rate samples as **more similar** or **more dissimilar**, as well as whether such tendencies are present in all raters, and whether they vary between raters. Hence, this analysis provides a better understanding of how different raters have rated the data differently.

After all, the goal is not just to measure the amount of agreement, but to gain a better understanding of the differences between the labelling behaviours of the various raters, and what could be done to improve overall agreement. The overall agreement among raters, measured by means of the different coefficients discussed earlier, will be covered in the Overall Agreement section of this chapter. In the remainder of this subsection, first an example will be given of how the **mean** rating is computed for a specific rater and question. Then, the results of the analyses will be discussed.

Example
Suppose Rater A has provided the following n ratings for n samples for the second question: {2, 3, 2, 4, 3, 5, 4, 3, 4, 2, 1, 3, 2, 4, 3, 5}. The mean rating given for the second question by Rater A will be:

$$\frac{2+3+2+4+3+5+4+3+4+2+1+3+2+4+3+5}{16} = 3,67$$

3.5.2 Markov-Walk Data-Set

The average ratings for each of the three questions which were given by each of the 28 raters who were involved in labelling the Markov-Walk dataset, are shown in Figure 3.2. A close inspection of these results allow one to make various observations. First of all, the mean ratings provided by the experts seem to be more similar, then those provided by the Ph.D Raters, and much more similar than those provided by the MSc. Raters. Rater 19 who has labelled 907 of the 2466 samples, has rated the samples across all three questions roughly as dissimilar, whereas Rater 26 rated, who labelled 517 samples, roughly rated them as similar across all three similarity questions. That is a difference of one class, which is significant, given that there are only 4 ordinally scaled similarity classes. we also see a huge discrepancy between the average ratings provided by Raters 19 and 20.

Note: *One might argue that Figure 3.2 does not provide a reliable estimate of the difference in bias, since the biases are computed over different sets of samples. However, since the criteria based on which pairs are formed contains only 5 classes, and the smallest class accounts for 10%, if both raters have rated at least 50 samples. it is unlikely that the sample sets are wildly different, given that they come from the same distribution*

Note: *Raters 7 and 24 have only rated 1 and 4 samples respectively. Obviously, the standard deviation of the provided ratings equals 0 if only a single rating is present. Rater 24 has provided the following ratings, which yields a standard deviation of 0 for question 3.*

Relative Biases

A closer look at the Relative Biases (Figure 3.3), plotted separately for 1) all pairs of raters; 2) all pairs of expert raters; 3) all pairs of Ph.D. raters; and 4) All pairs of MSc. Raters, we see a somewhat similar trend. While the median differences across all three questions are comparable for the MSc. Raters and the Expert raters, one can observe that the amount of difference in the 25'th percentile and above, is remarkably higher among the MSc. Raters than for the Expert raters. Overall, across all subsets of Raters, we see a much higher variation among the average ratings given for the second question than for the first and third questions. One surprising observation, especially in light of the bar charts shown in figure 3.2, is that the differences in bias seem to be lowest among the Ph.D. population of raters. Note, however, that this anomalous result might be due to the fact that there are such few pairs of raters among the expert raters that have at least 8 commonly rated samples, only 6 such pairs exist. All in all, one observes that the rater's biases, for each question, tends to vary by almost half a class (almost .5). This is a significant difference, given that there are only 4 classes, and the ratings are supposed to reflect the same ground truth.

3.5.3 FEM Data-Set

The biases of each of the seven raters for all questions and the differences between the biases of pairs of these raters across commonly rated samples are shown in Figure 3.4. Perhaps the first observation one would make, is that the experts (Raters 1 through 3) all have rated the samples for the second question as 2.5, on average, which is right in the middle between the two extremes, whereas the MSc. raters (Raters 4 through 7) have provided an average rating of roughly 2 for the second question (2 = dissimilar).

The differences between the averages for pairs of Raters further confirm that the expert raters have similar averages for question 2. Across all questions, the averages of the ratings tend to vary more among the MSc. raters, while the differences between MSc. raters and expert raters tend to be greater (top-right boxplot).

It is interesting to note that Raters 1 through 3 have been involved in the crack assessment automation project since its initiation in 2019. Perhaps, close collaboration and joint discussions have led them to provide assessments that are more aligned.

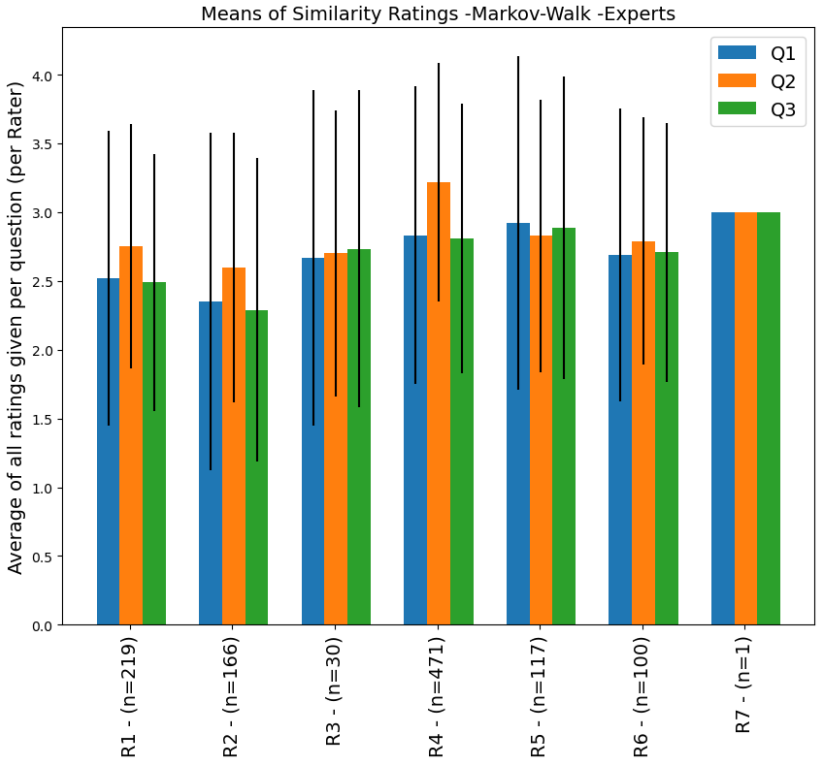

Figure 3.2: Markov-Walk: Mean Ratings per Question - Expert Raters

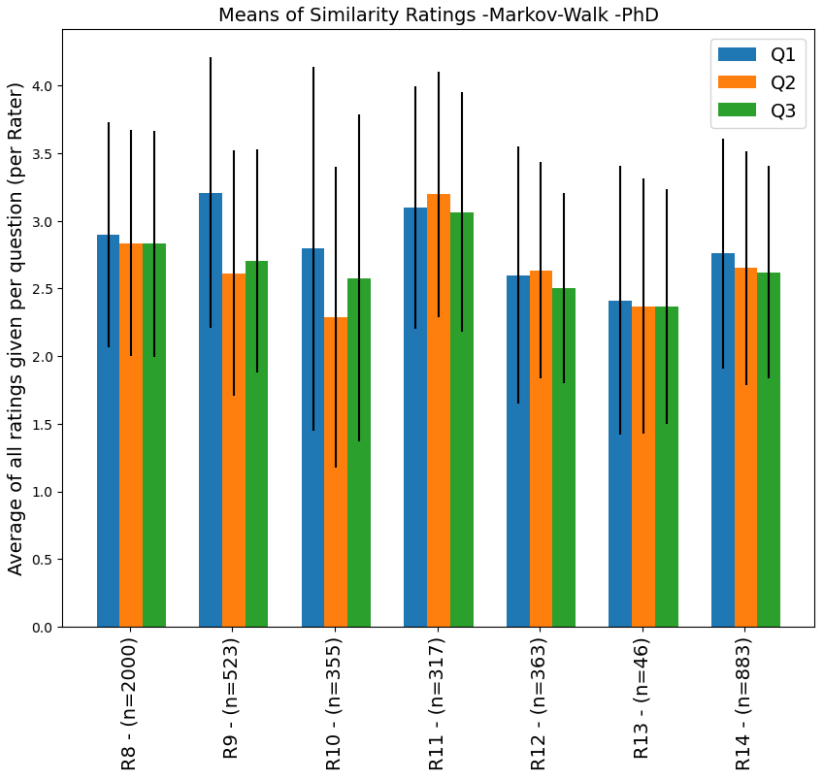

Figure 3.3: Markov-Walk: Mean Ratings per Question - Ph.D. Raters

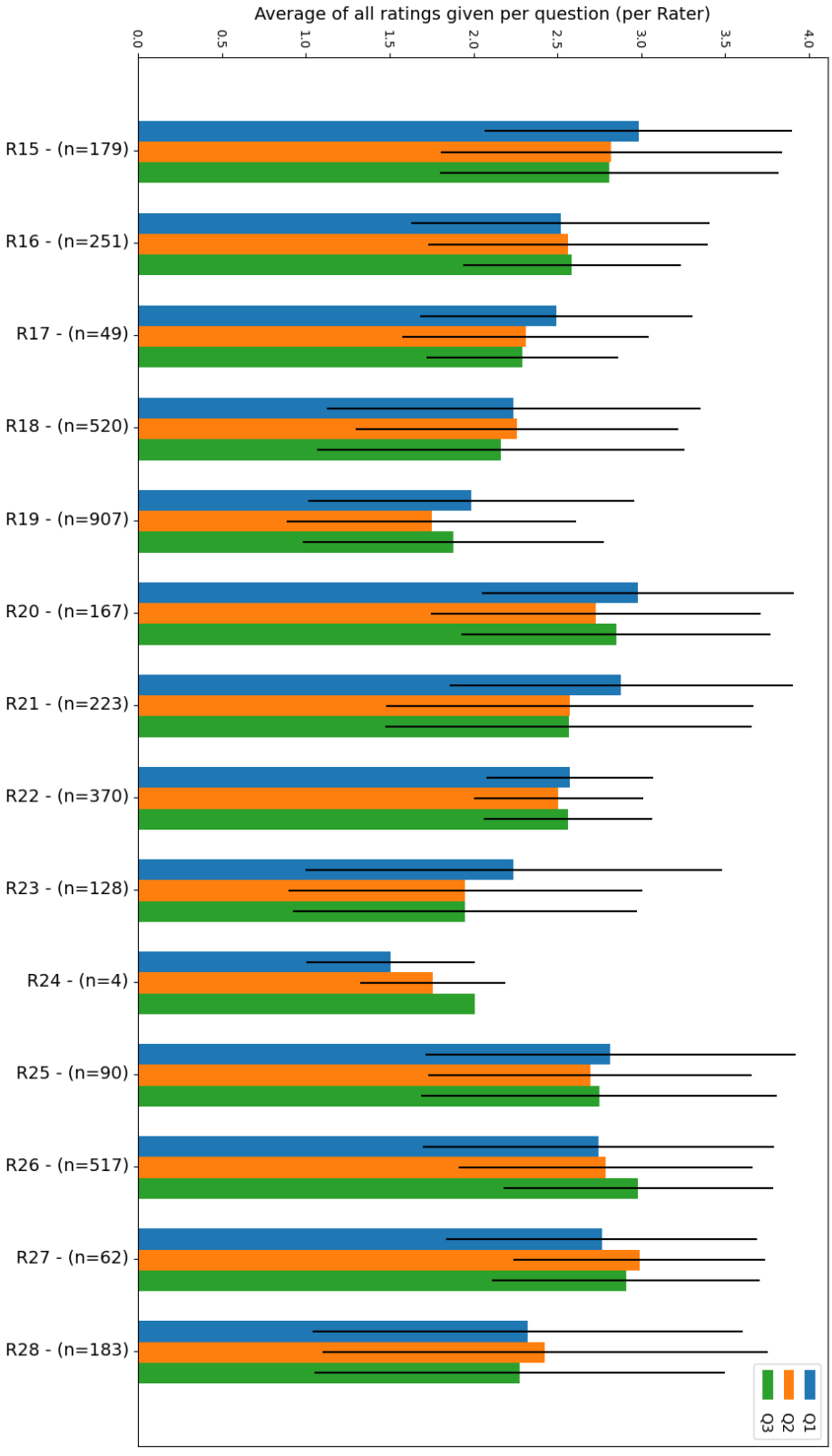

Figure 3.4: Markov-Walk: Mean Ratings per Question - MSc. Raters

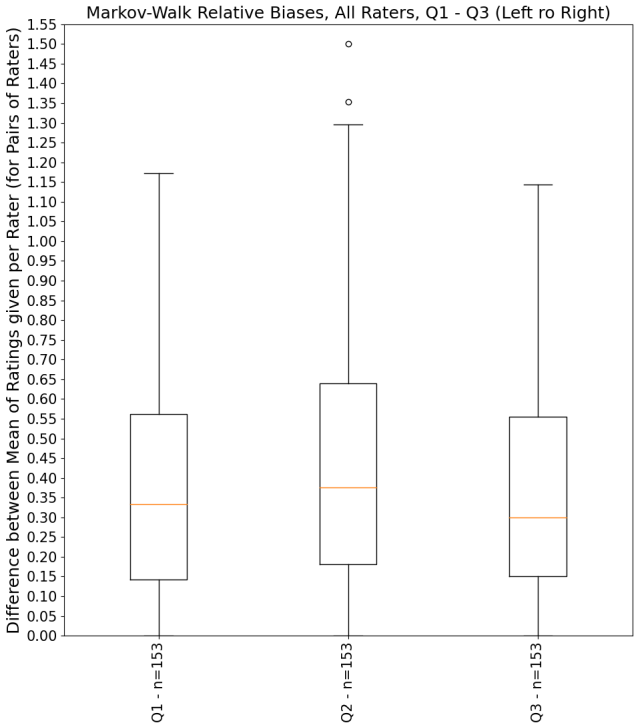

Figure 3.5: Markov-Walk: Relative Biases - All Raters

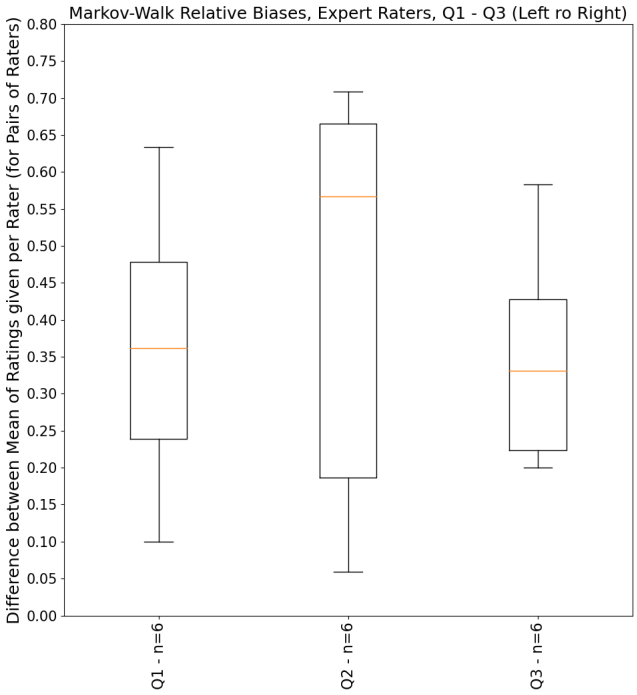

Figure 3.6: Markov-Walk: Relative Biases - Expert Raters

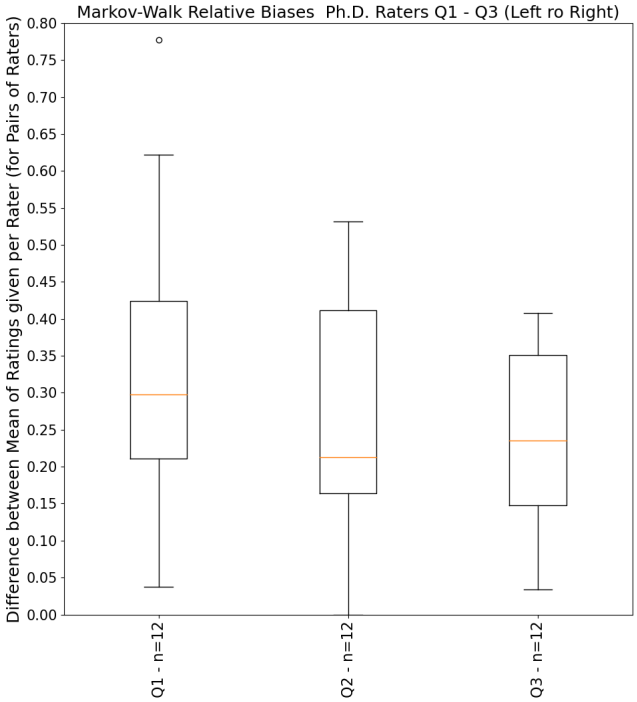

Figure 3.7: Markov-Walk: Relative Biases - Ph.D. Raters

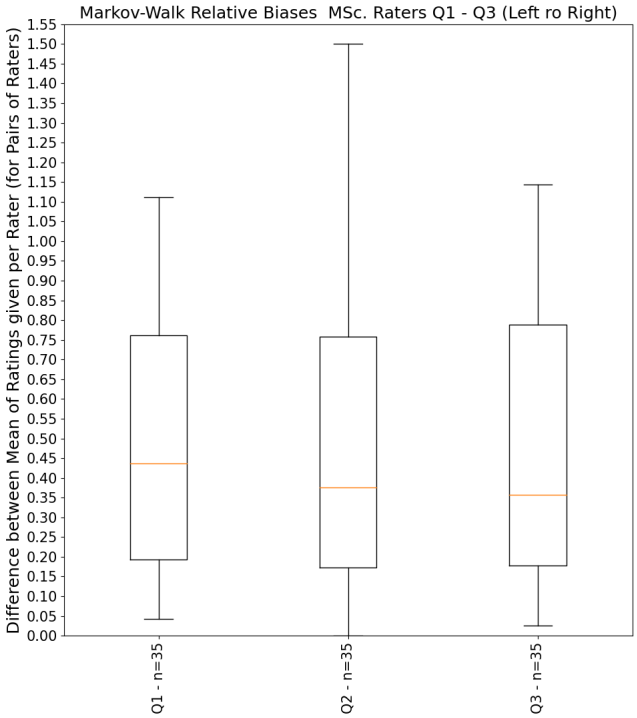

Figure 3.8: Markov-Walk: Relative Biases - MSc. Raters

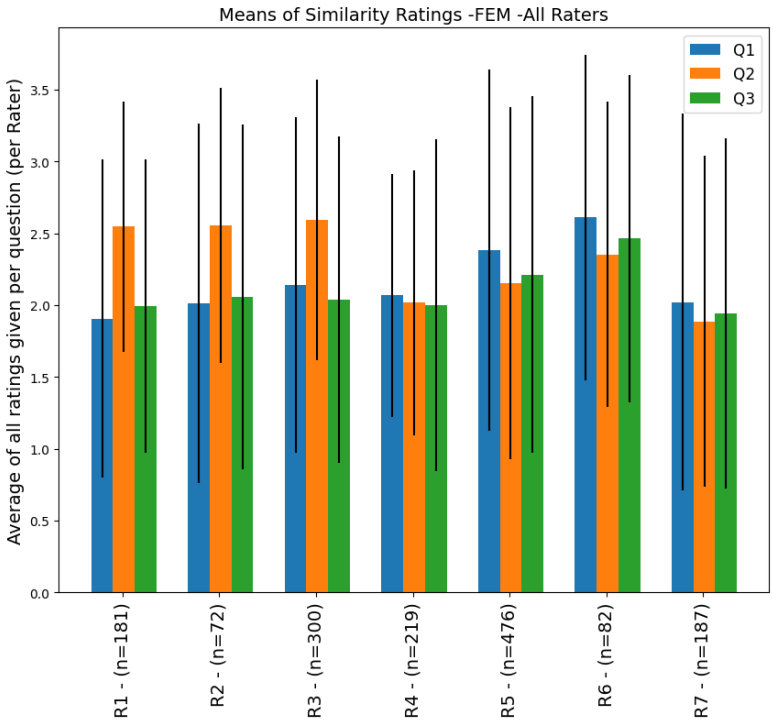

Figure 3.9: FEM: Mean Ratings per Question - All Raters

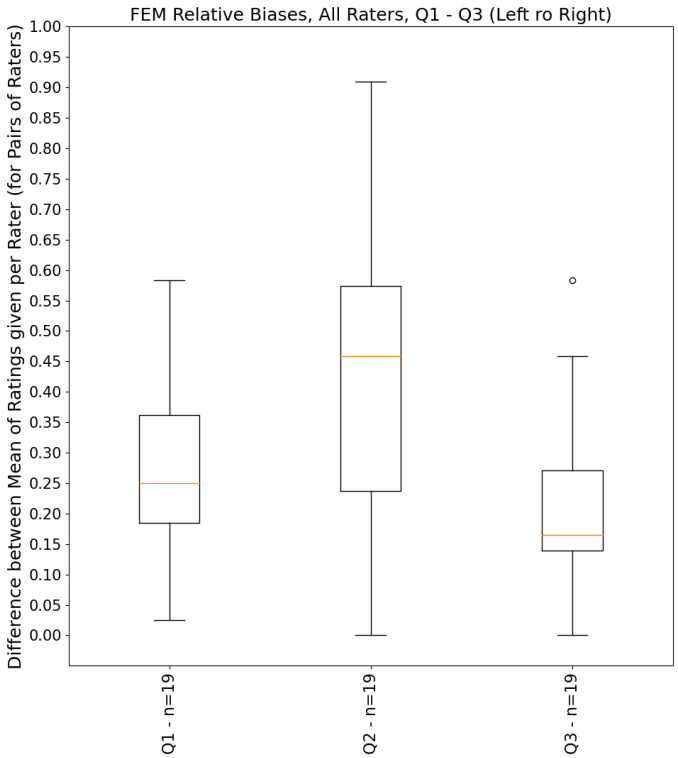

Figure 3.10: FEM: Relative Biases - All Raters

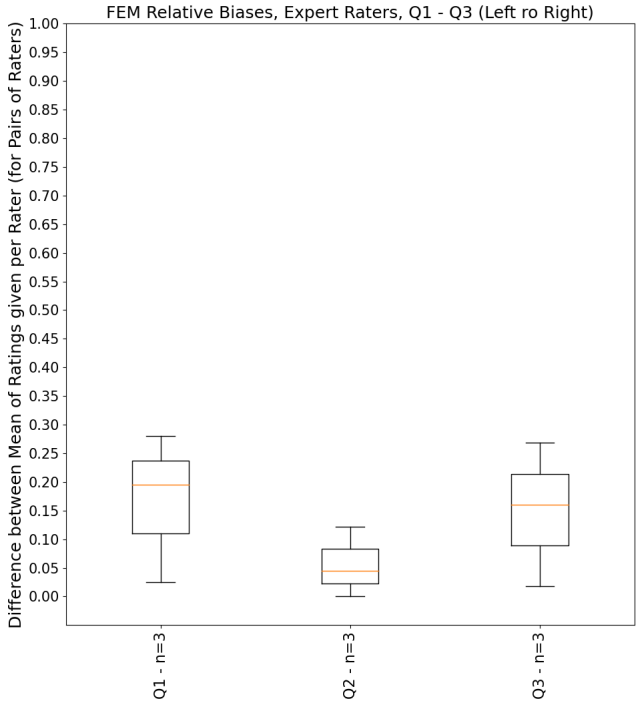

Figure 3.11: FEM: Relative Biases - Expert Raters

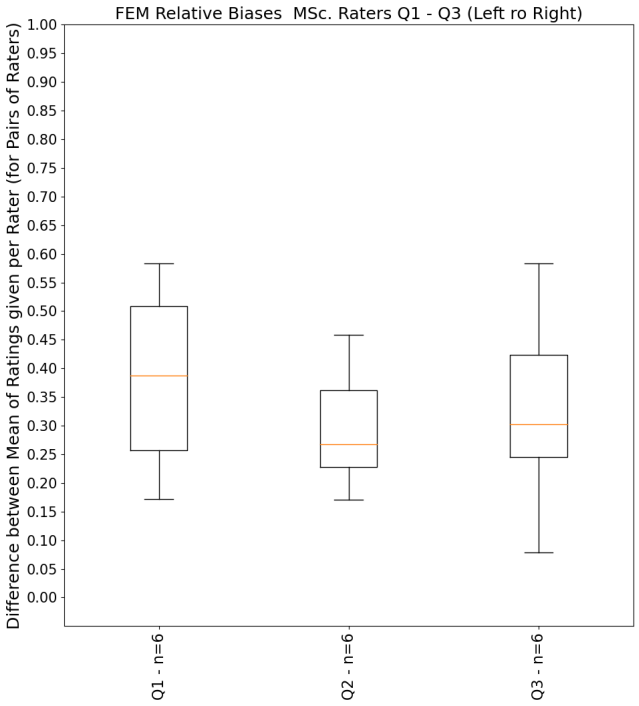

Figure 3.12: FEM: Relative Biases - Ph.D. Raters

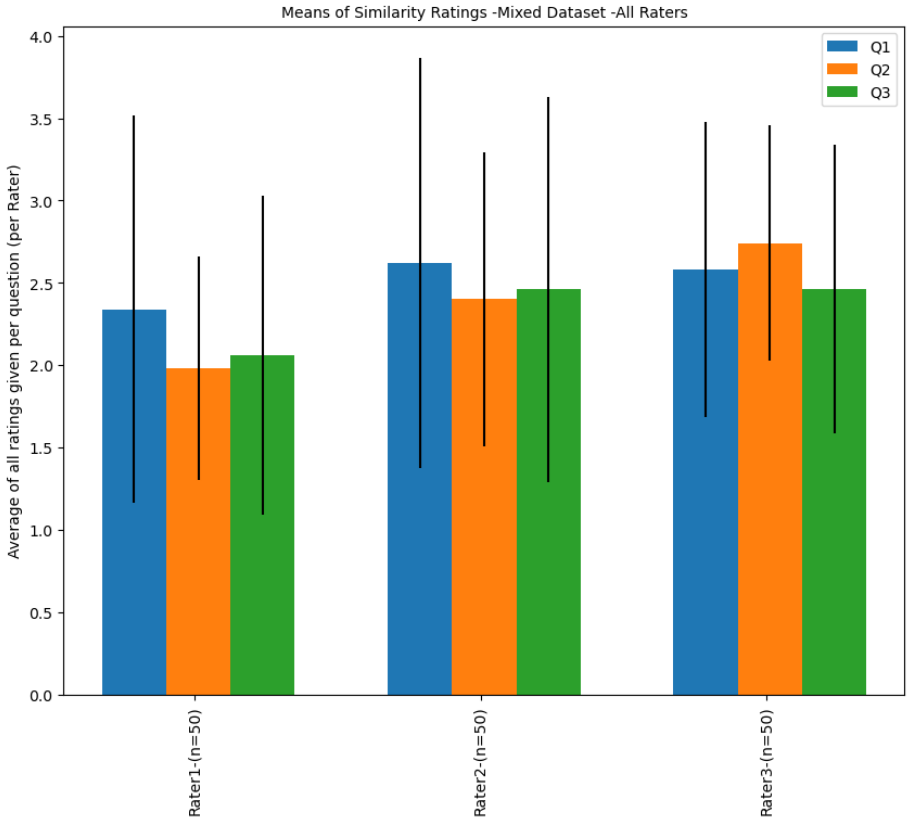

Figure 3.13: Mixed Dataset - Mean Ratings per Question - All (Expert) Raters

3.5.4 Mixed Data-Set

Interestingly, for the mixed dataset, no boxplots of the differences between the biases of pairs of raters are needed, because all three raters have labelled the same set of samples which consists of 50 image pairs. Hence, we will only consider the bar chart shown in Figure 3.5 to get an idea of how the averages of their ratings are distributed.

We see some discrepancies between the averages of the provided ratings across all three questions. Whereas, Rater 1 assigned an average score of 2 (dissimilar) for question 1, both Rater 2 and Rater 3 tend to rate the overall similarity as 2.5 (between similar and dissimilar). The greatest difference that one can observe, is the difference between the average ratings provided for question 2 by Raters 1 and 3, approximately around 1.95 and 2.75, respectively. This clearly demonstrates that the raters have different biases. Note that all three of the raters are experts. Furthermore, one can observe that the standard deviation for the ratings provided for question 1 by Rater 3 is significantly

smaller than the standard deviation of the ratings provided by Rater 2 for the same question. All in all, the results shown in this subsection motivate a more thorough examination of why the raters demonstrate different biases in the labelling task.

3.6 Q-to-Q Correlations

In this section I shed light on the extent to which ratings provided by the same rater for the three different questions are correlated. In the summer of 2021 discussions were held in which masonry experts questioned the suitability of the then current set-up of the labelling task. At that particular time, only a single dataset of pairs of images of cracked facades existed (395 samples), the annotations of which had been provided by Dr. Giorgia Giardina. Each sample was annotated with a similarity interval. The similarity intervals were:

- **0.0 - 0.4**
- **0.4 - 0.59**
- **0.59 - 0.79**
- **0.79 - 0.89**
- **0.89 - 0.96**
- **0.96 - 1.00**

Upon close inspection of the two pairs of synthetically generated cracked masonry facades, Dr. Giardina suggested that the first pair has high similarity in terms of damage pattern/cause (See Figures 5.1 and 5.2 in the Appendix), whereas the cracked facades of the second pair were highly similar in terms of severity (See Figures 5.3 and 5.4).

This observation motivated Dr. Giardina to propose a more fine-grained approach to the assessment of the degree of similarity between pairs of cracked facades. The initial set-up was deserted, i.e. a single similarity score, expressed as one of the aforementioned similarity intervals, and three novel notions of similarity were proposed.

1. **Similarity in terms of Damage Pattern/Cause**
2. **Similarity in terms of Severity**
3. **Overall Similarity**

The general expectation was that the ratings provided for the third question would roughly represent a weighted average of the ratings provided for the first and second questions. However, no specific instructions were provided to raters involved in labelling our datasets on how to interpret the third question in relation to the first two questions. In that sense, it was argued, raters would be given the freedom to decide for themselves how these questions were to be viewed in relation to one another. Furthermore, we decided to switch to four similarity intervals instead of the original six. These are the following

- **0.0 - 0.25**
- **0.25 - 0.50**
- **0.50 - 0.75**
- **0.75 - 1.00**

In order to better understand the relations between these questions as evidenced by the provided ratings, I have used the mean difference as well as Lin's CCC. (See Figure 3.6)

In the following, the question to question correlations for all three pairs of questions are visualized separately for each rater. The results of this analysis are discussed separately for the Markov-Walk, the FEM and the Mixed datasets, in the order in which they are stated here.

3.6.1 Q-to-Q Markov-Walk

The Q-to-Q Correlations for each of the 28 raters of the Markov-Walk Data-Set are shown in Figures 3.6 to 3.8. As discussed previously, one would expect questions 1 and 3, and questions 2 and 3 to be at least somewhat correlated. When analyzing these charts, one will likely observe that for most raters, the ratings provided for question 1 and question 3 show the strongest correlations, but there are several exceptions to this rule (Raters 9, 12 (Ph.D.) 15, 17, 21 and 23 (MSc.)).

Quite surprisingly, questions 1 and 2, are also quite correlated. One could argue that cracks with similar width and length, two factors that are generally considered to underlie the notion of similarity in terms of severity, would also be considered slightly more correlated in terms of damage pattern/cause. This would imply that the concepts of damage pattern/cause and severity are not independent of one another (i.e. they somewhat overlap).

Another possible explanation for this phenomenon is that people are inherently biased to provide ratings for the different questions that are somewhat similar. Stated differently, people are less likely to provide wildly different assessments for the three different questions (e.g. Q1: Very Dissimilar, Q2: Very Similar). Unsurprisingly, one can observe that the correlations expressed by Lin's CCC and the Mean Difference are roughly inversely correlated.

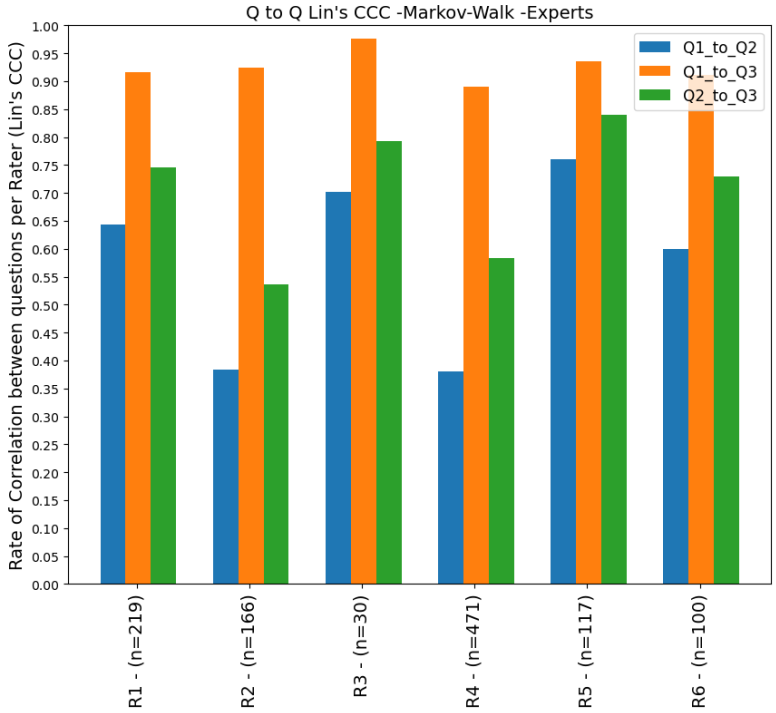

Figure 3.14: Markov-Walk: Correlations between Ratings for Question Pairs - Expert Raters - Lin's CCC

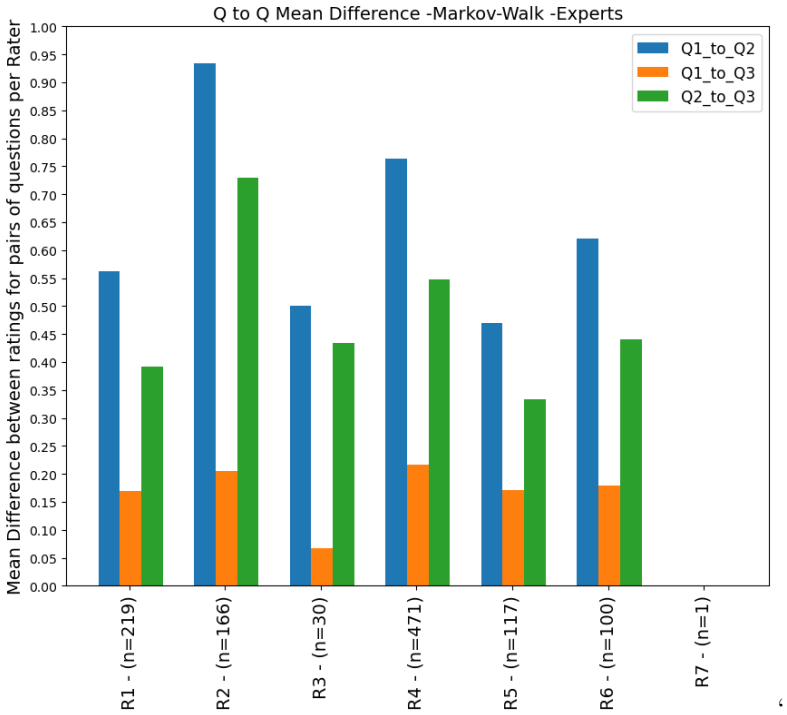

Figure 3.15: Markov-Walk: Correlations between Ratings for Question Pairs - Expert Raters - Mean Difference

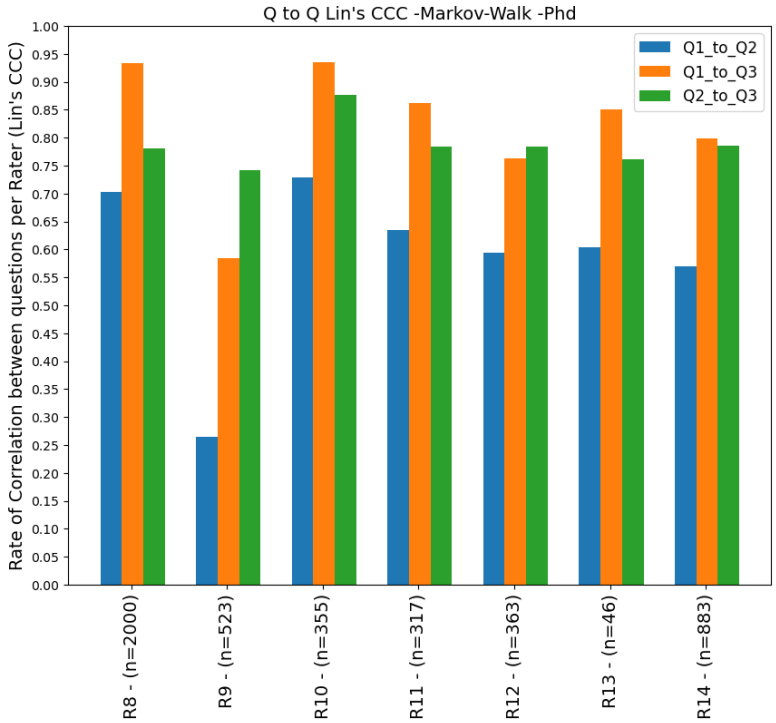

Figure 3.16: Markov-Walk: Correlations between Ratings for Question Pairs - Ph.D. Raters - Lin's CCC

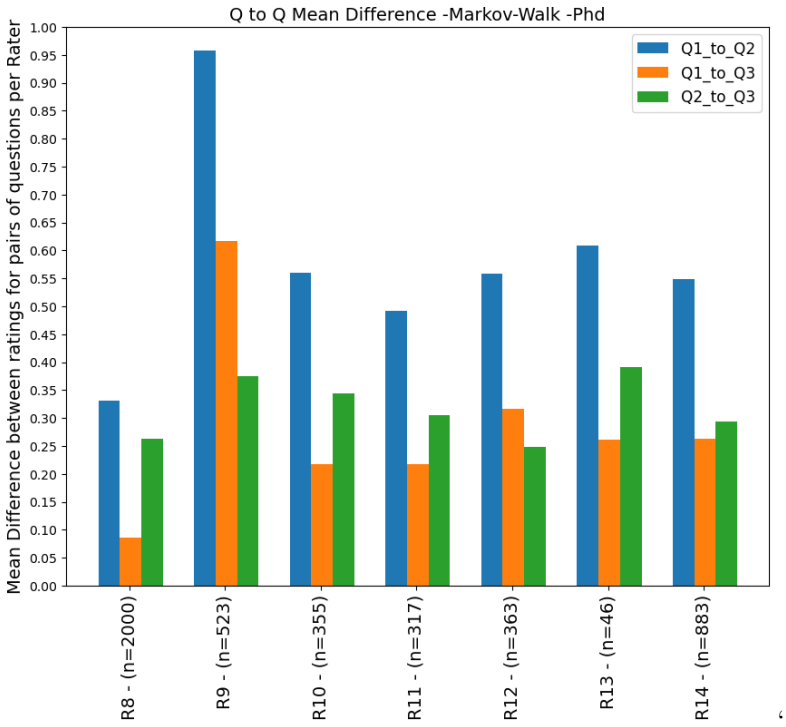

Figure 3.17: Markov-Walk: Correlations between Ratings for Question Pairs - Ph.D. Raters - Mean Difference

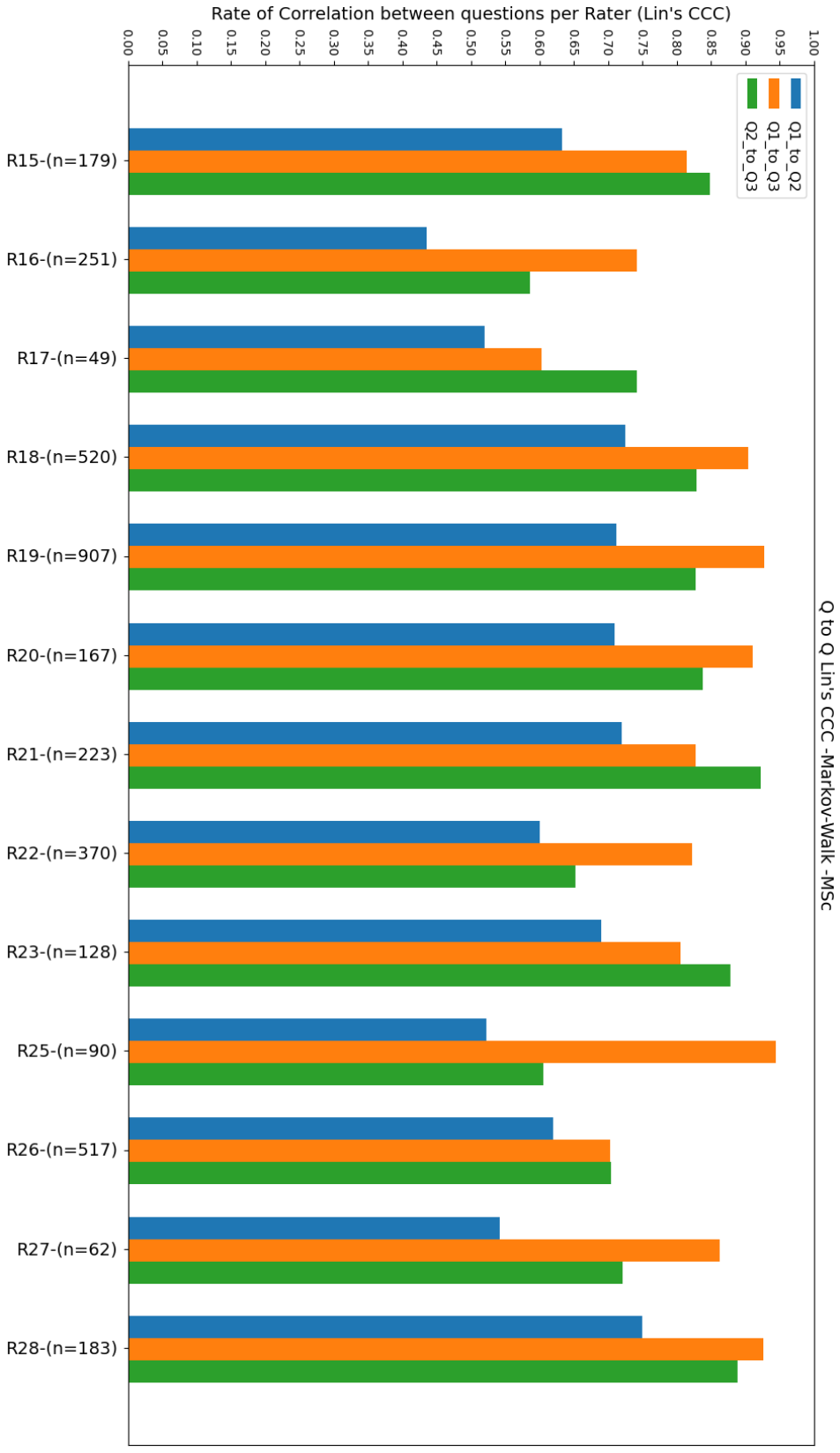

Figure 3.18: Markov-Walk: Correlations between Ratings for Question Pairs - MSc. Raters - Lin's CCC

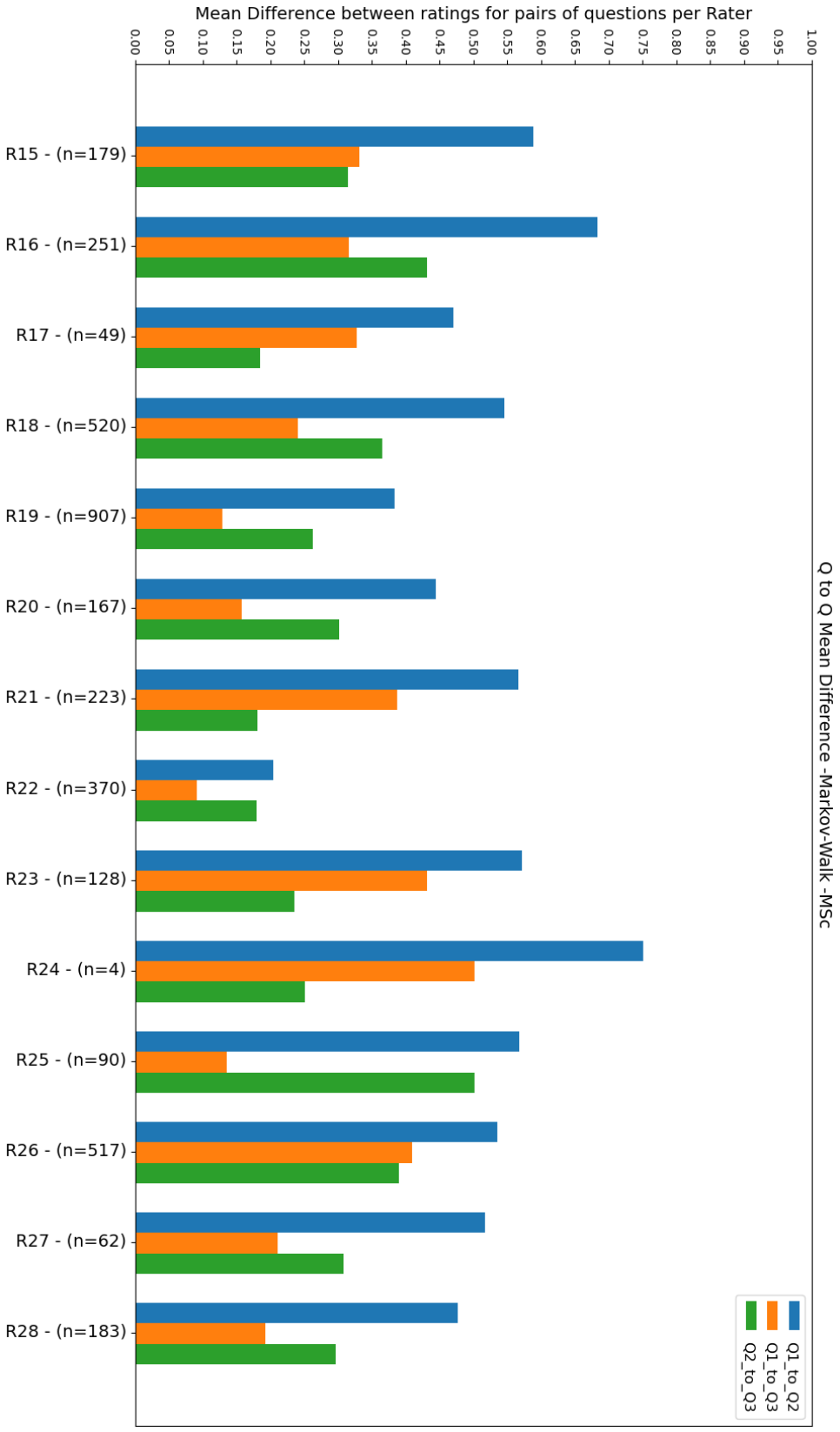

Figure 3.19: Markov-Walk: Correlations between Ratings for Question Pairs - MSc. Raters - Mean Difference

3.6.2 Q-to-Q FEM

The correlations between the different questions for each of the seven Raters are illustrated in Figure 3.9. Similarly to what we observed in the question to question correlations for the Markov-Walk dataset, the expert raters tend to provide highly similar ratings for questions 1 and 3. Note that the Lin's CCC scores for question 1 and question 3 are in excess of .9 for all Expert Raters. Especially for the third Rater, one finds that the Lin's CCC score is roughly .96, which is so high, that the ratings provided for questions 1 and 3 by Rater 3 are virtually interchangeable.

As is the case for the Markov-Walk dataset, one can observe that all pairs of questions are somewhat correlated, including questions 1 and 2. Perhaps the most remarkable observation is that there is a clear difference between the labelling behaviours of the Experts and that of the MSc. Raters. Across the 4 MSc. Raters, the correlations between all pairs of questions are .75 and above, and for Raters 4 to 7 .85 and above. Furthermore, questions 2 and 3 are more correlated than questions 1 and 3 across all MSc. Raters, whereas the reverse holds true for the Expert Raters. This is somewhat alarming.

One would be inclined to ask **Why is there such an enormous difference between the Experts and the MSc. Raters?** and **Why are all Q-to-Q correlations so high for all MSc. Raters?** As mentioned previously, regarding the former question, one could argue that the three involved Expert Raters have collaborated closely throughout this project and that this has somehow resulted in similar rating behaviour across these three raters. A possible explanation regarding the latter question is that the MSc. raters would get tired after rating a certain number of samples, and hence, as their labelling efforts proceeded, they increasingly tended to assign the same similarity label to all three questions, rather than to carefully inspect the two cracked facades and consider the three questions separately.

Regardless of what truly brought about these differences, these results strongly motivate a more thorough analysis of the labelling behaviour of the raters, and a reconsideration of the validity of the current set-up of the similarity assessment task at large.

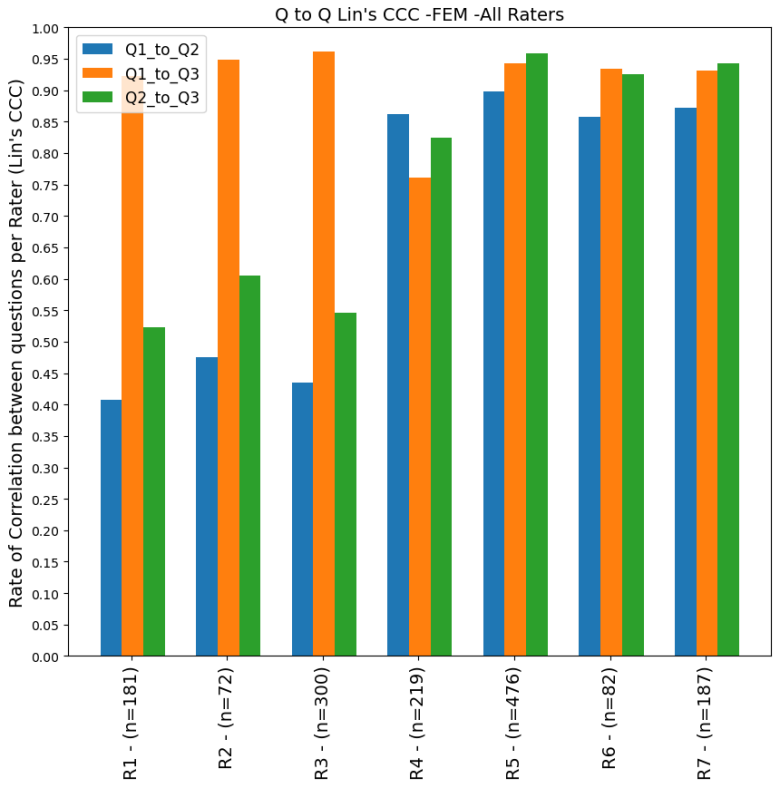

Figure 3.20: FEM: Correlations between Ratings for Question Pairs - All Raters - Lin's CCC

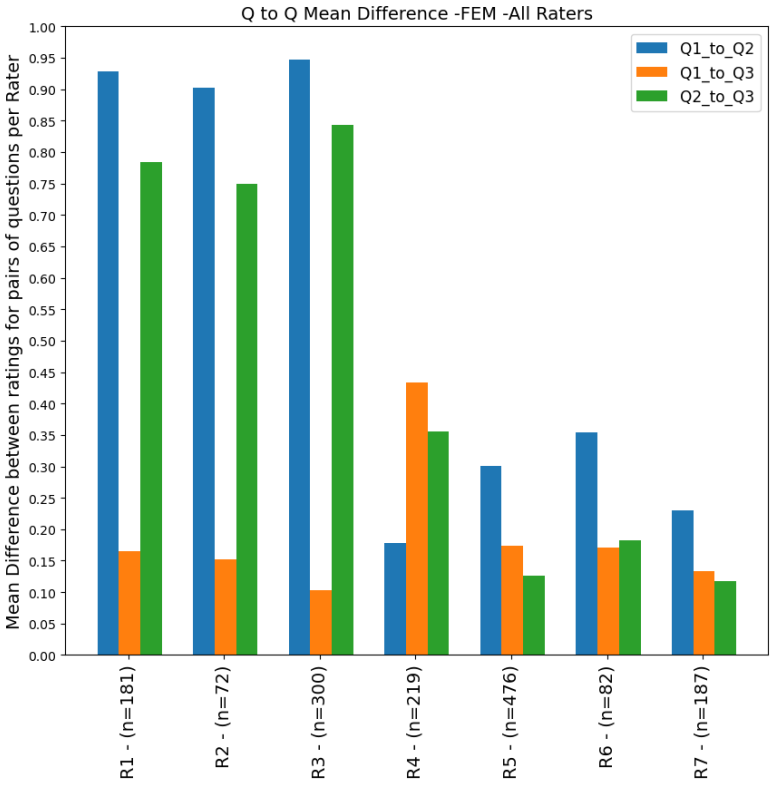

Figure 3.21: FEM: Correlations between Ratings for Question Pairs - All Raters - Mean Difference

3.6.3 Q-to-Q Mixed Data-Set

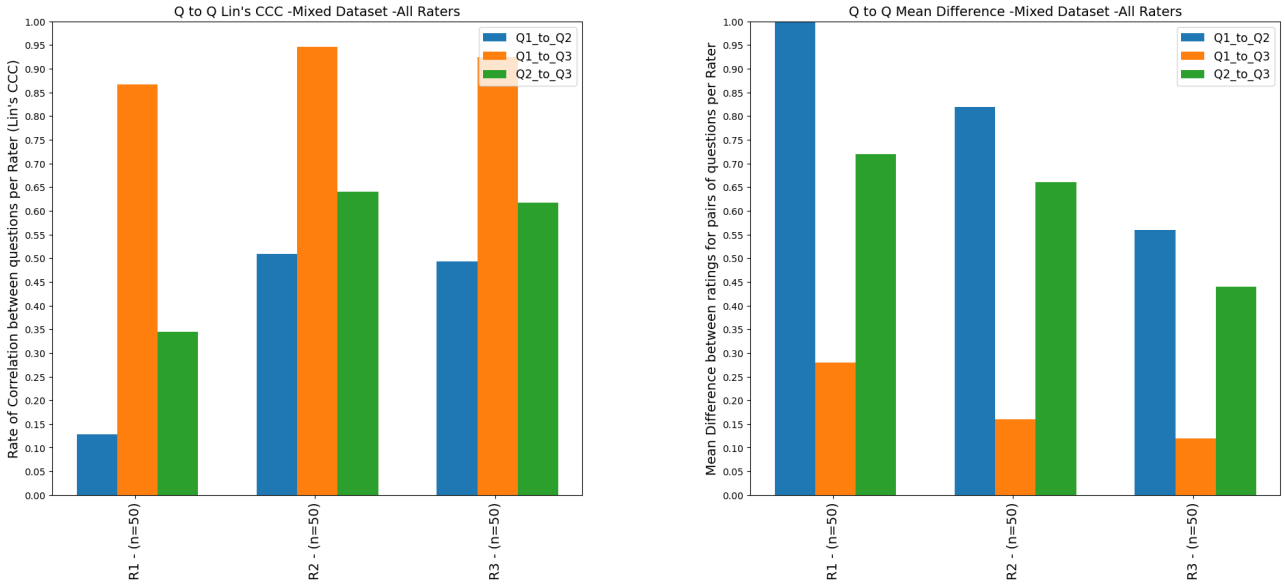

Figure 3.22: Mixed Dataset: Question to Question Correlations for each of the three Raters, expressed by Lin's CCC (Left) and Mean Difference (Right)

3.6.4 Q-to-Q Mixed Data-Set

The correlations between the different questions for each of the three Raters are illustrated in Figure 3.10. Upon close inspection of the two plots, one can conclude that for all three raters, questions 1 and 3 are most strongly correlated, a finding that is consistent across all three datasets. For Raters 2 and 3, the ratings provided for question 1 and question 3 are extremely similar. For the first Rater, the question to question correlations are slightly lower than for Raters 2 and 3, across all three pairs of questions. All in all, the results across all three datasets show that questions 1 and 2 are not given equal weightage in determining the degree of overall similarity.

Models	All Raters			Expert Raters			MSc. Raters		
	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3
Krippendorff's Alpha	0.674	0.562	0.733	0.694	0.527	0.737	0.629	0.654	0.673
Unweighted Kappa	0.342	0.189	0.380	0.282	0.170	0.329	0.395	0.333	0.335
Weighted Kappa	0.509	0.367	0.570	0.481	0.352	0.552	0.528	0.510	0.527
Lin's CCC	0.637	0.524	0.711	0.637	0.528	0.725	0.628	0.646	0.661
Pearson's R	0.671	0.587	0.730	0.663	0.598	0.737	0.662	0.676	0.688
Kendall's Tau	0.606	0.499	0.625	0.570	0.531	0.595	0.611	0.592	0.616
Mean Difference	0.615	0.775	0.534	0.588	0.715	0.518	0.676	0.623	0.652
Percent Agreement	0.545	0.400	0.589	0.551	0.420	0.551	0.580	0.538	0.538

Table 3.3: Overall Agreements and Correlations FEM Data

Models	All Raters			Expert Raters			Ph.D. Raters			MSc. Raters		
	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3
Krippendorff's Alpha	0.634	0.519	0.618	0.804	0.543	0.740	0.667	0.562	0.663	0.621	0.493	0.613
Unweighted Kappa	0.268	0.226	0.261	0.468	0.328	0.437	0.252	0.322	0.366	0.186	0.160	0.196
Weighted Kappa	0.458	0.363	0.437	0.657	0.446	0.613	0.445	0.459	0.500	0.375	0.304	0.378
Lin's CCC	0.621	0.499	0.601	0.804	0.573	0.764	0.624	0.596	0.640	0.545	0.447	0.546
Pearson's R	0.711	0.593	0.707	0.868	0.660	0.826	0.678	0.632	0.681	0.675	0.557	0.695
Kendall's Tau	0.648	0.538	0.654	0.811	0.601	0.764	0.622	0.573	0.639	0.614	0.501	0.641
Mean Difference	0.645	0.706	0.619	0.419	0.576	0.427	0.592	0.536	0.460	0.736	0.772	0.690
Percent Agreement	0.469	0.442	0.484	0.568	0.495	0.586	0.536	0.512	0.582	0.427	0.377	0.445

Table 3.4: Overall Agreements and Correlations Markov Walk Data

Models	All Raters		
	Q1	Q2	Q3
Krippendorff's Alpha	0.780	0.316	0.700
Unweighted Kappa	0.374	0.161	0.306
Weighted Kappa	0.600	0.259	0.517
Lin's CCC	0.774	0.361	0.700
Pearson's R	0.812	0.442	0.755
Kendall's Tau	0.739	0.383	0.686
Mean Difference	0.507	0.693	0.560
Percent Agreement	0.520	0.420	0.480

Table 3.5: Overall Agreements and Correlations Mixed Dataset

3.7 Overall Agreement (per Dataset)

To provide an overview of the inter-rater reliability (amount of agreement) between raters for all three datasets, I have computed the average correlations and agreements between all pairs of raters for each dataset, excluding pairs who share less than eight commonly rated samples. These results are shown in the three tables shown above (Tables 3.3 - 3.5). The three different colors that are used to display the values (green, yellow and red) are interpreted as follows:

- **Green**: *sufficient agreement/correlation*
- **Yellow**: *Moderate, yet insufficient agreement/correlation*
- **Red**: *Insufficient Agreement/Correlation.*

Furthermore, regarding the interpretation of the used coefficients, i.e. which range of values corresponds to green, yellow or red, and for which coefficient, I apply the following criteria for the following coefficients, mostly based on guidelines presented in the literature as covered in the Literature Review section of this thesis:

- Kendall's Tau/Pearson's R:
  - Moderate Agreement: $0.60 - 0.80$
  - Sufficient Agreement: $0.80 - 1.00$
- Krippendorff's Alpha:
  - Moderate Agreement: $0.67 - 0.80$
  - Sufficient Agreement: $0.80 - 1.00$
- Kappa Statistics
  - Moderate Agreement: $0.50 - 0.75$
  - Sufficient Agreement: $0.75 - 1.0$
- Lin's CCC
  - Moderate Agreement: $0.9 - 0.95$
  - Sufficient Agreement: $0.95 - 1.0$

Any values that are below the Moderate Agreement threshold are considered to provide evidence of poor agreement.

3.7.1 Discussion of Results

The results confirm, beyond any doubt, that the rate of agreement among the 28 raters is generally insufficient, and in many cases even poor. Especially for the second question there is poor agreement among the raters across all datasets, including all sub-populations of raters. One can observe that there is slightly more agreement among the expert raters than there is among any other sub-population of raters. However, this holds true mostly for the Markov-Walk dataset, and only for the first and third questions. Regarding the second question, there is no more agreement among experts than there is among MSc. raters, Ph.D. raters or all raters.

The most promising result is the amount of agreement measured among the expert raters for the first question for the Markov-Walk dataset. Its corresponding column contains three green-colored values. It also stands out that across all three datasets, the experts agree most on samples belonging to the Markov-Walk dataset. Their agreement is clearly higher for the Markov-Walk dataset

than for any other dataset, even for the second question, which suggests that the samples from the Markov-Walk dataset might visually be the clearest and most convenient to interpret. Another interesting observation is that the agreement among the experts is highest for the first question, except when one considers samples generated with the FEM, in the latter case the agreement is highest for the third question.

Furthermore. one notes that the results of the linear weighted kappa and the Krippendorff's Alpha are somewhat conflicting. As discussed in the Literature Review of this thesis, this sometimes happens, since no agreement coefficient is perfect, and most agreement coefficients fail to express the amount of agreement in alignment with one's intuition in at least some cases.

All in all, the agreement among raters, including expert raters, is clearly too low, but the agreement among the experts for the Markov-Walk dataset is hopefull. Especially the degree of agreement for the first question shows that this approach has potential, provided that effective steps are taken to further improve the rate of agreement.

3.8 Low Agreement Samples

As a part of my efforts, I have strived to quantify the amount of agreement for individual samples. That is, given a collection of samples, each of which was rated by multiple raters, how do we compute the relative amount of disagreement that we observe among ratings provided for single samples? The advantage of this endeavour is that it allows one to identify characteristics of samples for which the agreement is comparatively low.

The remainder of this section is organized as follows. First, I explain how I have quantified the amount of disagreement by means of an illustrative example. Then, I will discuss the findings and results for each of the three aforementioned datasets, separately. And finally, I provide a summary of the findings and conclusions.

3.8.1 Quantifying Disagreement per Sample

Suppose that the following set of ratings were provided for a pair of cracked facades:

- **Rater1:** {*2, 2, 3*}
- **Rater1:** {*3, 3, 3*}
- **Rater1:** {*2, 2, 4*}

These three sets of ratings (all w.r.t. the same sample) allow for three pairwise comparisons presented below:

$$\begin{Bmatrix} 2 & 2 & 3 \\ 3 & 3 & 3 \end{Bmatrix} \rightarrow \textit{Total Difference} = 2$$

$$\begin{Bmatrix} 2 & 2 & 3 \\ 2 & 2 & 4 \end{Bmatrix} \rightarrow \textit{Mean Difference} = 1$$

$$\begin{Bmatrix} 3 & 3 & 3 \\ 2 & 2 & 4 \end{Bmatrix} \rightarrow \textit{Mean Difference} = 3$$

$$\frac{\textit{Sum of Total Differences}}{\textit{Total number of Comparisons}} \rightarrow \frac{2+1+3}{9} = 0.67$$

3.8.2 Markov-Walk Dataset

As stated previously, the Markov-Walk Dataset consists of 2466 samples. For each of these 2466 samples, I have quantified the amount of disagreement. An overview of the pairs of crack archetypes that occur among the 250 samples with the highest degree of disagreement are shown in Table 3.6, along with their occurrence frequencies. We find that the following pairs of crack archetypes are especially problematic in terms of agreement. These are:

- **101, 102**
- **103, 103**
- **103, 201**
- **20, 21**
- **201, 201**
- **102, 102**

Some examples of problematic samples belonging to the categories listed above, are shown in the Appendix (FiguresWhen one purely determines how problematic a pair of crack archetypes is (in terms of maximizing disagreement) it would seem natural to consider the fraction of the samples corresponding to the pair over the entire dataset, that occurs among the x% most problematic pairs. For example, one might argue that the pair (23, 102) should have been listed as well since 40% of its samples are among the 10% of samples with the lowest agreement. Here, I have chosen a trade-off between the fraction and the total number of samples that occur in the set of samples that are least agreed on.

Archetypes	Occurrences	Total Occurrences
(101 , 102)	56	154
(103 , 103)	25	100
(103 , 201)	22	156
(20 , 21)	21	155
(101 , 101)	18	114
(23 , 30)	17	165
(102 , 102)	14	71
(201 , 201)	13	89
(24 , 32)	11	181
(31 , 32)	8	181
(30 , 30)	8	81
(24 , 24)	7	107
(32 , 32)	6	96
(18 , 20)	3	131
(30 , 102)	2	9
(23 , 102)	2	5
(18 , 18)	2	78
(31 , 31)	2	106
(20 , 20)	2	88
(32 , 103)	1	1
(31 , 101)	1	6
(102 , 103)	1	4
(23 , 23)	1	103
(23 , 24)	1	4
(31 , 103)	1	2
(30 , 32)	1	3
(18 , 101)	1	1
(24 , 103)	1	3
(21 , 21)	1	103
(101 , 103)	1	5

Table 3.6: Markov-Walk Dataset: All pairs of crack archetypes that occur (and how often) among the 250 lowest agreement samples, and how often these crack pairs occur in the entire dataset (2466 samples)

Archetypes	Occurrences	Total Occurrences
(23 , 24)	13	23
(24 , 103)	11	23
(20 , 21)	9	23
(18 , 102)	9	14
(23 , 103)	6	11
(101 , 102)	6	15
(18 , 20)	6	23
(18 , 101)	5	14
(103 , 103)	4	14
(18 , 21)	4	23
(24 , 24)	3	12
(23 , 102)	3	12
(101 , 101)	2	15
(20 , 102)	2	14
(102 , 102)	2	12
(101 , 103)	2	11
(102 , 103)	2	4
(20 , 101)	2	14
(20 , 20)	1	13
(23 , 23)	1	13
(18 , 18)	1	15
(18 , 23)	1	14
(21 , 102)	1	11
(23 , 101)	1	10
(24 , 102)	1	10
(18 , 103)	1	14
(21 , 101)	1	11

Table 3.7: FEM Dataset: All pairs of crack archetypes that occur (and how often) in the 100 poorest agreement samples, and how often these crack pairs occur in the entire dataset (500 samples)

3.8.3 FEM Dataset

The FEM Dataset consists of 500 samples. An overview of the pairs of crack archetypes that occur among the 100 samples with the highest degree of disagreement are shown in Table 3.7, along with their occurrence frequencies. For this dataset, one can identify the following pairs of crack archetypes that are problematic in terms of agreement:

- **23, 24**
- **24, 103**
- **20, 21**
- **18, 102**
- **23, 103**
- **101, 102**

Examples of samples with high disagreement are shown in the Appendix Figures 5.5 - 5.8.

Image1	Image2	Disagreement
32_0_A_123.npy	21_10x3_0.4_0_40000.npy	0.0
102_6x4_0.2_3_210.npy	21_0_A_106.npy	0.074
21_8x4_0.5_0_5000.npy	31_0_B_110.npy	0.074
30_0_A_113.npy	18_10x3_0.5_0_50004.npy	0.074
23_0_C_100.npy	Facade_1.npy	0.074
201_1_A_103.npy	23_10x3_0.5_0_50000.npy	0.074
20_0_A_101.npy	20_10x3_0.4_0_40000.npy	0.074
Facade_2.npy	101_1_A_106.npy	0.074
23_0_A_104.npy	23_10x3_0.5_0_50000.npy	0.074
201_1_A_103.npy	23_6x4_0.2_0_200.npy	0.074
103_2_C_103.npy	103_10x3_0.2_7_20000.npy	0.148
18_6x4_0.2_0_200.npy	20_0_A_101.npy	0.148
31_0_B_110.npy	24_4x4_0.4_0_41.npy	0.148
Facade_2.npy	Facade_3.npy	0.148
24_4x4_0.4_0_41.npy	31_0_B_110.npy	0.148
Facade_3_Real.npy	201_2_A_105.npy	0.148
18_10x3_0.5_0_50004.npy	103_2_C_103.npy	0.148
Facade_3_Converted.npy	30_0_A_113.npy	0.148
Facade_3_Real.npy	30_0_A_113.npy	0.148
32_0_A_123.npy	102_6x4_0.2_3_210.npy	0.148
102_1_B_102.npy	102_6x4_0.2_3_210.npy	0.148
23_0_A_104.npy	23_6x4_0.2_0_200.npy	0.222
21_8x4_0.5_0_5000.npy	20_0_A_101.npy	0.222
101_10x3_0.4_8_40010.npy	32_0_A_123.npy	0.222
18_0_B_119.npy	18_6x4_0.2_0_200.npy	0.222
23_0_C_100.npy	Facade_2.npy	0.222
Facade_3_Converted.npy	201_2_A_105.npy	0.222
20_0_A_101.npy	20_4x4_0.4_0_40.npy	0.222
101_6x4_0.3_2_301.npy	18_0_B_119.npy	0.22
30_0_A_113.npy	102_6x4_0.2_3_210.npy	0.22
103_2_C_103.npy	103_4x4_0.5_1_50.npy	0.22
101_10x3_0.4_8_40010.npy	102_1_B_102.npy	0.22
101_1_B_108.npy	101_6x4_0.3_2_301.npy	0.22
Facade_1.npy	201_2_A_105.npy	0.22
Facade_2.npy	32_0_A_100.npy	0.22
102_1_B_102.npy	102_10x3_0.3_9_30024.npy	0.22
21_0_A_106.npy	21_8x4_0.5_0_5000.npy	0.22
Facade_4.npy	101_1_A_106.npy	0.22
Facade_4.npy	201_1_B_100.npy	0.22
101_1_B_108.npy	101_10x3_0.4_8_40010.npy	0.296
18_0_B_119.npy	18_10x3_0.5_0_50004.npy	0.296
102_6x4_0.2_3_210.npy	18_0_B_119.npy	0.296
24_0_A_117.npy	24_4x4_0.4_0_41.npy	0.296
103_4x4_0.5_1_50.npy	103_2_C_103.npy	0.296
24_0_A_117.npy	24_10x3_0.5_0_50001.npy	0.296
31_0_B_110.npy	18_6x4_0.2_0_200.npy	0.296
21_0_A_106.npy	21_10x3_0.4_0_40000.npy	0.296
201_1_B_100.npy	Facade_1.npy	0.370
Facade_1.npy	201_1_B_100.npy	0.370
Facade_3.npy	101_1_A_106.npy	0.370

Table 3.8: Mixed Dataset: The names of the cracked facade matrices which form the 50 samples along with the corresponding amount of disagreement

3.8.4 Mixed Dataset

Since the Mixed Dataset contains only 50 samples, I have decided to compute the amount of disagreement for each of the 50 samples. The results are shown in Table 3.8.

One can observe that image pairs that consist of one cracked facade generated by one of the simulation model and one of the real-world cracked facades tend to maximize disagreement among the expert raters.

Chapter 4

Neural Network Experiments

This chapter covers the deep learning experiments that I have conducted to address the third and fourth research questions defined in chapter 1.

4.1 Introduction

As briefly discussed in the introduction of this thesis, the main goal is to help automate the assessment of cracked facades. In order to achieve this, I have fitted the neural network using various train/test splits, and evaluated its performance. As stated previously, three datasets have been amassed to support the goal of crack assessment automation, and I have fitted the model to both the Markov-Walk and FEM datasets, separately. One can distinguish three different types of train/test splits, all of which are outlined below:

- **Random-Split**: *75% of the samples are randomly assigned to the train set and the remaining 25% is assigned to the test set.*
- **Leave-One-Crack Out**: *All samples that contain the leave-out crack archetype will be assigned to the test set, and all remaining samples are assigned to the training set.*
- **Leave-One-Class Out**: *All samples that belong to the leave-out similarity class are assigned to the test set, and all remaining samples are assigned to the training set*

4.2 Outline

The remainder of this chapter is organized as follows. First, an overview of the architecture of the neural network is provided, as well as how it is trained. Secondly, I cover how multiple annotations given by multiple raters are converted

to a set of labels used to train the neural network, and how the cracked facades are fed to the network. And finally, I present and discuss the experiments and the corresponding results.

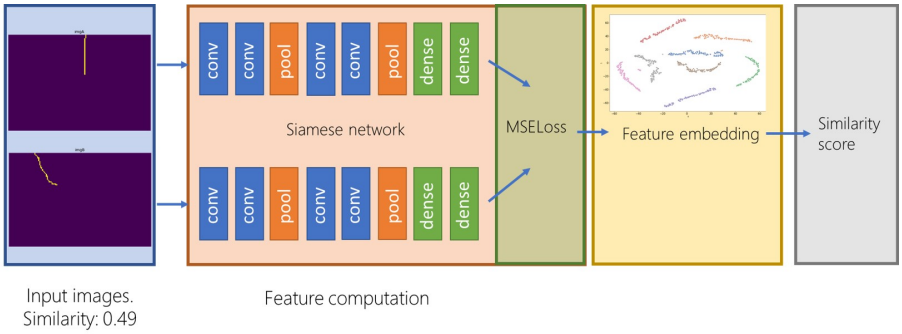

Figure 4.1: Architecture of the Siamese Network used for learning the similarity between crack archetypes.

4.3 Neural Network Architecture

As stated in the introductory chapter of this thesis, I have used the same neural network architecture that Wyke Pereboom and Maarten Kruithof have proposed [3]. An overview of this architecture is shown in Figure 4.1. The architecture is similar to that of the classification network shown in Figure 1.13. Note that the parallel sets of layers represent the same network. Furthermore, the second-last (dense) layer has size 2048, while the last layer has 64 neurons.

4.3.1 Similarity Computation

As shown in Figure 4.1, the images (81 pixels long and 161 pixels wide) are run through the network in paralell. In essence, both images are run through the same convolutional neural network (CNN), and each is mapped to a vector in 64-dimensional space. Then, we compute the Euclidean distance in this 64-dimensional space (implemented with torch.norm function from PyTorch), and subsequently quantify the similarity of two points in this space as follows: $1 - tanh(d)$, where d denotes the computed distance between the two vectors.

$$tanh(d) = \frac{sinh(x)}{cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Note that the co-domain of $tanh(x)$ is the interval $[-1, 1]$, and $d > 0 \rightarrow tanh(d) \geq 0$, and since $d \geq 0$, we have that $0 \leq 1 - tanh(d) \leq 1$.

The performance of the neural network is measured by computing the so-called **loss** which reflects how much the predicted similarity differs from the actual

similarity scores. In this case, the loss is computed by means of the **Mean Squared Error** (MSE), which is defined as $MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$, where n denotes the number of samples, Y_i denotes the ground truth similarity score for sample i and $\hat{Y}_i$ denotes the similarity score predicted by the Siamese Network for sample i. Note that in our particular case, the MSE is always a scalar from the interval $[0, 1]$. The MSE differs from the **Mean Absolute Error** (MAE) in that the differences between the true similarity and predicted scores are squared.

Additionally, I also compute the accuracy, F_1, the precision and the recall for both training and test sets. While it is more common to assess the performance of a regression model by means of the MAE or MSE, this case is unusual in the sense that the original annotations can be viewed as similarity intervals rather than scalars in the continuous interval $[0, 1]$. Hence, predicted scores will fall into one of a number of intervals that cover the domain $[0, 1]$ which justifies the treatment of neural network predictions as assignments to one of the four similarity classes mentioned earlier.

The network is trained in a number of so-called **epochs**. In each of these epochs, the network is updated, and a corresponding updated model is obtained. This process is also known as the **training phase**. In each epoch, the dataset is split up in so-called **mini batches**. Rather than running all n samples through the network at once, the samples are propagated through the network in a per batch fashion, and the weights of the network are updated for each batch, separately.

I have used batches of 16 samples for all experiments. The network is trained by updating its weights in each epoch, which is done by backpropagating the error signal, that is the loss on the training set, in a backwards manner from the last layer to the first layer. Note that only the loss over the training set affects the updation of the network.

It is customary to pre-define the criteria for termination of the training phase. In the original set-up of the training phase, as defined by Wyke Pereboom and Maarten Kruithof, the network was trained for 50 consecutive epochs and the model resulting from the last epoch was used for regression.

I have decided to take a slightly different approach. The model would usually improve iteratively for the first 20 epochs and then its performance would hoover around, getting better in some epochs, and worse in others.

I decided to set the number of epochs to 75, and terminate the process once no improvement is obtained over 20 consecutive epochs, as this set-up allows for incremental improvements, even after 20 to 30 epochs, while avoiding the pitfall of training indefinitely without achieving better performance. Once training has terminated, it is not the model resulting from the last epoch that is returned, but the model that best minimizes the loss on the training set.

4.4 Annotations

As discussed in the Inter-Rater Reliability chapter of this thesis, most samples were rated by multiple raters. As stated before, we have decided to eliminate all samples that were rated fewer than three times from the dataset (See Section 3.3.1). However, since the neural network can only handle a single true similarity score per sample, this calls for a policy on how to combine these multiple ratings into a single rating.

We have jointly discussed this matter and have identified three distinct ways in which this can be achieved.

1. **Majority Voting**:
   In the case that n raters have rated a given sample, one simply picks the majority vote as the true similarity score for this sample. Note that this is not always possible, since a majority vote may not exist. However, it has the nice property that all samples will fall nicely into the center of one of the similarity bins. Cases for which no majority vote exist will be handled differently (to be discussed later).

2. **Averaging**:
   A possible alternative to majority voting is to take the average rating of a set of n ratings (provided by n raters). The upside to this approach is that, unlike majority voting, it can always be applied. The downside is that it can prove to be problematic when one treats the similarity regression task as a classification task, especially when the average falls on the boundary between two similarity bins (to be discussed later).

3. **One Sample per Labelling Instance**:
   A third alternative is to have one sample for each labelling instance. That is, if n raters have labelled the i-th sample, one simply feeds the i-th sample to the network n times, once for each of the n ratings. This alternative was briefly discussed, but we quickly decided to drop it.

In the following, the conversion from the provided assessments to annotations, on a per sample basis, will be discussed in more detail.

4.4.1 Conversion Annotations to Labels

As stated previously, all assessments were provided as one of the four ordinal categories that were mentioned before. I have chosen to convert these to so-called similarity intervals (or ranges). The correspondences between the categories and intervals are shown below:

- **Very Dissimilar: 0.00 - 0.25, Center: 0.125**
- **Dissimilar: 0.25 - 0.50, Center: 0.375**
- **Similar: 0.50 - 0.75, Center: 0.625**

- **Very Similar: 0.75 - 1.00, Center: 0.875**

All assessments are converted to their corresponding range, and the center of each range will be used as the ground truth, i.e. the center of a range of a sample i will be used as its true similarity, $\hat{Y}_i$. Examples of how majority voting and averaging are applied to obtain the annotations are given below.

Suppose we have the following three ratings provided by three separate raters for some sample:

- **{Q1: Very Dissimilar, Q2: Similar, Q3: Dissimilar}**
- **{Q1: Dissimilar, Q2: Similar, Q3: Similar}**
- **{Q1: Similar, Q2: Similar, Q3: Similar}**

Majority Voting

Note that majority voting can only be applied to the ratings pertaining to Q2 and Q3. Three different ratings were provided for Q1, hence no majority vote exists. For Q2, all three raters assigned the ratings **Similar**, hence $\mathbf{0.50 - 0.75}$ will be the similarity interval assigned to Q2, and the true similarity will thus be $\mathbf{0.625}$. For Q3, we have two occurrences of **Similar** and only one occurrence of **Dissimilar**, hence the similarity interval will be $\mathbf{0.50 - 0.75}$ and the true similarity score will be set to $\mathbf{0.625}$.

Averaging

Averaging can be applied in any case. For each question, the average similarity interval corresponding to the three provided ratings will be used to represent the ground truth similarity interval. For Q1, we have **{Very Dissimilar, Dissimilar, Similar}** which corresponds to **{0.0-0.25, 0.25-0.50, 0.50-0.75}**. The lower bound of the resulting similarity interval will be the mean of the lower bounds, and the upper bound will be computed similarly. The resulting bounds and similarity interval will be:

$$\left.\begin{aligned} \text{Lower-Bound} &= \frac{0.0 + 0.25 + 0.50}{3} = 0.25, \\ \text{Upper-Bound} &= \frac{0.25 + 0.50 + 0.75}{3} = 0.50, \end{aligned}\right\} \rightarrow \text{0.25-0.50} \rightarrow \text{Mean } = 0.375$$

One can verify that similar computations for Q2 and Q3 will yield the following averages:

$$Q2_{average} = \text{0.50-0.75} \rightarrow \text{Mean: } 0.625$$
$$Q3_{average} = \text{0.417-0.667} \rightarrow \text{Mean: } 0.567$$

Selected (Hybrid) Approach

I have decided to apply majority voting whenever possible, and average the various ratings in the case that no majority vote exists. While the results of averaging and majority voting are not necessarily meaningful, this approach enables us to convert all provided assessments to annotations that can be used to train the neural network. Still, difficulties may arise when computing the accuracy, F_1, Recall and Precision. These difficulties, and how to deal with them, will be discussed next.

Averaging: Divergent Similarity Intervals

Suppose that we have the following set of ratings for a given sample and for one of the questions: **{Similar, Dissimilar, Similar, Dissimilar}**. We have two occurrences of both **Similar** and **Dissimilar**. Once could easily verify that this yields an average similarity interval of **0.375-0.625**, its center being at 0.5, which is right at the boundary between similarity classes Similar and Dissimilar. How then, does one decide to which similarity class this sample belongs? Picking one of the two classes to assign such boundary cases to seems quite arbitrary.

In other cases, the center of a similarity interval resulting from the application of averaging, might fall within the boundaries of one of the original bins. Consider the case where the following five ratings were provided for a given question and sample: **{Similar, Dissimilar, Dissimilar, Dissimilar, Dissimilar}**. One can verify that this yields the average interval **0.30-0.55**. Frequent occurrences of divergent similarity intervals, that is similarity intervals other than the original intervals, can give rise to issues when computing and evaluating the accuracy of the neural network.

Example: Divergent Similarity Intervals

Suppose that one is faced with a set of assessments provided by multiple raters, and obtains the following annotations by means of applying averaging:

- **0.0-0.25**, **center: 0.125, 45 occurrences**
- **0.25-0.50, center: 0.375, 57 occurrences**
- **0.417-0.667 center: 0.542, 120 occurrences**
- **0.50-0.75 center: 0.625, 65 occurrences**
- **0.625-0.875 center: 0.75, 240 occurrences**
- **0.75-1.00 center: 0.875, 84 occurrences**

One can see that the first two bins together with the last two bins, are the original bins. Since 240 samples (almost half of all samples) fall on the boundary between the third and fourth bin, the neural network that one obtains through training will likely misclassify many of the 240 boundary cases. Moreover, there are 120 samples that are associated with the similarity interval **0.417-0.667**.

While these samples are not boundary cases, if one views them as belonging to the similarity class **0.50-0.75**, then the margin to the boundary with the similarity class **0.25-0.50** will only be **0.042**. hence these are also much more likely to be classified incorrectly.

Handling Divergent Similarity Intervals

While there probably is no flawless way to handle such cases properly, one can think of some strategies to handle them. When outlining such strategies one can distinguish between two distinct categories, namely boundary cases and non-boundary cases. Different dealing strategies that are appropriate for dealing with one of these categories, as well as strategies that work for both, are outlined separately below.

- **Boundary Cases**
  - **Ignore Samples**: *An alternative is to ignore the boundary cases when computing the accuracy. Note that this doesn't affect the training phase of the neural network, nor does it influence the mean Ssuared error and mean absolute error of the resulting model.*
  - **Accept Both Adjacent Classes**: *Another alternative is to make an exception for boundary cases when computing the model's accuracy. Whenever a boundary case presents itself, one considers it correctly classified when the predicted similarity falls within the boundaries of either of the adjacent similarity intervals. Thus, if the true similarity is set to 0.75, predictions 0.77 and 0.73 will both result in a correct classification.*
- **Non-Boundary Cases**
  - **Do Nothing**: *The least costly approach to dealing with the occurrence of non-boundary cases is to leave them as be. This approach works since the center of such similarity intervals falls well within the bounds of one of the original bins.*
- **Boundary and Non-Boundary Cases**
  - **Intermediate Bin**: *One strategy is to introduce an intermediate bin. Consider the example from the previous subsection. There are 240 boundary cases between the adjacent similarity intervals* **0.50-0.75** *and* **0.75-1.00**. *One could simply move the upperbound and lowerbound of the former and latter down and up by 0.0625, respectively, to introduce the intermediate bin* **0.5625-0.6875**. *Similarly, for the 120 samples associated with similarity interval* **0.417-0.667** *with center 0.542, a similar policy would yield new similarity bins* **0.25-0.4585**, **0.4585-0.4585-0.5835** *and* **0.5835-0.75**. When one wants to introduce new similarity intervals to cover both cases, one could simply

consider the centers of all represented bins in increasing order, in this particular case 0.125, 0.375, 0.542, 0.625, 0.75 and 0.875 and define new boundaries in the middle between subsequent center values to obtain similarity intervals **0.0-0.25**, **0.25-0.4585**, **0.4585-0.5835**, **0.5835-0.6875**, **0.6875-0.8125** and **0.8125-1.000**

- **Convert to Original Bin**: *Another alternative for dealing with both boundary and non-boundary cases is to simply replace their similarity bins with one of the original bins, specifically the nearest bin, which implies that the true similarity of these samples will also shift. Of all the dealing strategies proposed thus far, this is the only one which affects the true similarity score of the sample as resulting from the application of averaging.*

Note: *Introducing narrower similarity intervals will cause a sharp drop in accuracy, since it significantly reduces the margin around the bin centers. However, the MSE and MAE will not be affected by the number of similarity classes.*

4.5 FEM Dataset Experiments

In this section I report on neural network Experiments performed on the FEM dataset. First, I discuss how the labels were obtained for this dataset. Secondly, I discuss how the different cracked facades were fed as input to the neural network, and then the three different experiments are discussed.

4.5.1 FEM Dataset Annotations

This section sheds light on the distribution of the labels. As discussed in the previous section, converting annotations from multiple raters to labels sometimes requires averaging of multiple annotations which may result in divergent similarity intervals. Therefore, after conversion of the annotations to labels, an additional post-processing step is often required to handle such cases. Table 4.1 shows the distribution of labels before and after handling divergent similarity intervals. The *Before* columns show the distribution of the labels of the samples right after applying the *hybrid approach* to obtain the labels (discussed in Section 4.4.1), while the *After* columns show the distribution of the labels after handling the divergent similarity intervals (after post-processing step). Note that there are samples that are not associated with any of the 4 original similarity intervals in the *Before* column, namely all those that are not associated with similarities 0.125, 0.375, 0.625 or 0.875. Out of a total of 499 samples, there are only 12 such samples for Q1, 15 for Q2, and 13 for Q3, a small fraction of the total number of samples. Hence, I have decided to convert each of these samples to the nearest original bin. That is, all similarity intervals were converted to the original similarity intervals, and thus, the resulting similarity intervals are {**0-0.25, 0.25-0.50, 0.50-0.75, 0.75-1.00**}.

Similarities	Q1		Q2		Q3	
	Before	After	Before	After	Before	After
0.125	206	206	154	155	224	226
0.25	0	0	1	0	2	0
0.375	84	84	149	155	104	106
0.458	0	0	4	0	1	0
0.5	0	0	2	0	1	0
0.542	11	0	6	0	7	0
0.625	94	106	94	102	58	67
0.75	1	0	2	0	2	0
0.875	103	103	87	87	100	100

Table 4.1: FEM Dataset: Distribution of the true similarity scores of all samples before and after handling of divergent similaritie intervals

In the following, The **Random-Split**, **Leave One Crack Out** and **Leave One Class Out** experiments will be discussed in order.

4.5.2 Preparing Cracked Facades for Training

In order to fit a neural network to a dataset which consists of pairs of cracked facades, one must feed these cracked facades to the network. To this end, we have decided to represent the cracked facades as numpy matrices. For the FEM dataset, these cracked facades were originally modelled in Diana by Krishna Ajithkumar Pillai [4]. An important consideration in this context is that the cracked facades modelled in Diana have different shapes and aspect ratio's. Hence, the resulting numpy matrices have different shapes as well. In order to still enable fitting the network to these different cracked facades, we have decided to resize the images to size (height = 81px, width = 161px). The code for this was provided by Wyke Pereboom and Krishna Ajithkumar Pillai. In order to minimize the amount of information lost in the process of resizing, nearest neighbour interpolation from the cv2 Python library was used. In the resulting numpy matrices, cracks are represented by values > 0, the facade itself by 0 and openings are represented by -1.

4.5.3 Random-Split

This section presents the results of the Random-Split Experiment on the FEM dataset.

Experimental Set-Up

In this experiment, 75% of the entire dataset was assigned to the training set, and 25% to the test set. To shuffle the data in order to ensure that samples

are assigned to either sets at random, I have used **np.random.shuffle**. The training set is then fed to the neural network for training. Upon termination of the training phase, the best model encountered thus far is returned and used for prediction. This entire process is repeated 10 times in order to obtain reliable estimates of the model's performance. In each of the 10 runs, the MAE, MSE, F_1, accuracy, precision and recall as well as the underlying confusion matrices, are computed. After all 10 runs have finished, these same metrics are averaged across the 10 runs, including the confusion matrices.

In order to present a detailed overview of the neural network's performance, I have computed several scores including F_1, accuracy, precision, recall, MAE and MSE and calculated confusion matrices. Furthermore, I have identified common characteristis of samples for which the neural network performed relatively poorly (see Section 4.5.6).

In the following, first the results for each of the three questions will be shown. Then, a discussion of all results follows.

Results Q1

Runs	MSE		MAE		Accuracy		F1		Precision		Recall	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Run 1	.0104	.0112	.0766	.0781	.794	.776	.804	.805	.794	.776	.837	.884
Run 2	.0091	.0096	.0708	.0748	.810	.816	.821	.825	.810	.816	.860	.843
Run 3	.0106	.0068	.0775	.0601	.799	.856	.812	.862	.799	.856	.848	.890
Run 4	.0108	.0109	.0762	.0813	.805	.744	.816	.760	.805	.744	.863	.817
Run 5	.0085	.0095	.0672	.0745	.829	.824	.838	.836	.829	.824	.866	.874
Run 6	.0081	.0103	.0647	.0763	.856	.752	.862	.771	.856	.752	.882	.817
Run 7	.0106	.0070	.0757	.0650	.791	.896	.806	.900	.791	.896	.851	.914
Run 8	.0118	.0089	.0809	.0714	.770	.832	.785	.842	.770	.832	.827	.879
Run 9	.0107	.0067	.0763	.0607	.786	.896	.802	.899	.786	.896	.848	.913
Run 10	.0104	.0079	.0753	.0659	.791	.848	.803	.859	.791	.848	.844	.898
Averages	.0101	.0089	.0741	.0708	.803	.824	.815	.836	.803	.824	.853	.873
St.devs	.0011	.0016	.0047	.0071	.0229	.0501	.021	.045	.023	.051	.015	.034

Table 4.2: FEM Dataset: Random-Split Performance by various metrics, Q1.

$$\begin{bmatrix} 113.7 \pm 3.95 & 40.1 \pm 5.43 & 0.9 \pm 0.54 & 0.0 \pm 0.0 \\ 4.3 \pm 0.9 & 52.2 \pm 3.76 & 6.9 \pm 1.51 & 0.8 \pm 0.4 \\ 0.0 \pm 0.0 & 10.8 \pm 2.09 & 65.6 \pm 3.23 & 1.9 \pm 0.83 \\ 0.0 \pm 0.0 & 0.0 \pm 0.0 & 7.9 \pm 2.12 & 68.9 \pm 5.01 \end{bmatrix}$$

Figure 4.2: Confusion Matrix, Train, Q1.

$$\begin{bmatrix} 38.6 \pm 4.08 & 12.6 \pm 4.1 & 0.1 \pm 0.3 & 0.0 \pm 0.0 \\ 1.2 \pm 1.33 & 16.2 \pm 4.14 & 2.4 \pm 1.56 & 0.0 \pm 0.0 \\ 0.0 \pm 0.0 & 3.0 \pm 1.67 & 24.3 \pm 3.23 & 0.4 \pm 0.49 \\ 0.0 \pm 0.0 & 0.0 \pm 0.0 & 2.3 \pm 1.19 & 23.9 \pm 4.3 \end{bmatrix}$$

Figure 4.3: Confusion Matrix, Test, Q1.

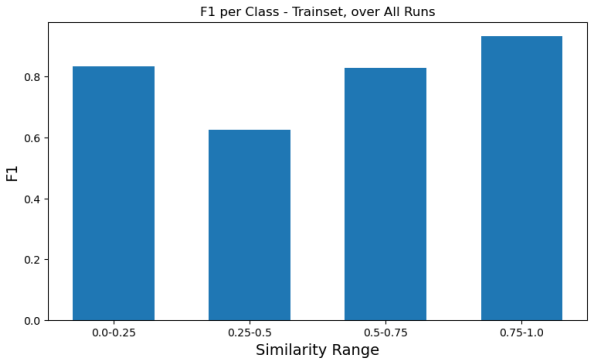

Figure 4.4: Mean F1 per class, Train, Q1.

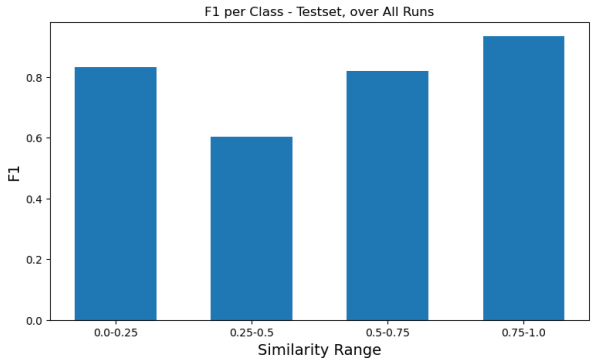

Figure 4.5: Mean F1 per class, Test, Q1.

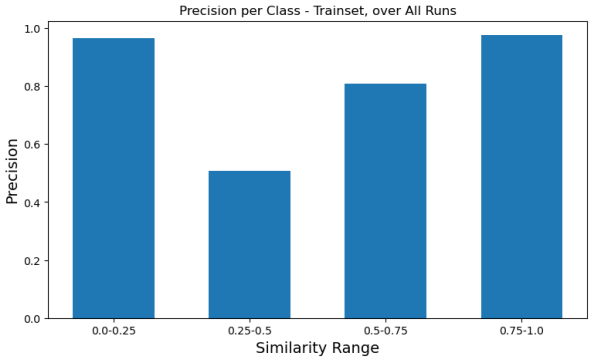

Figure 4.6: Mean precision per class, Train, Q1.

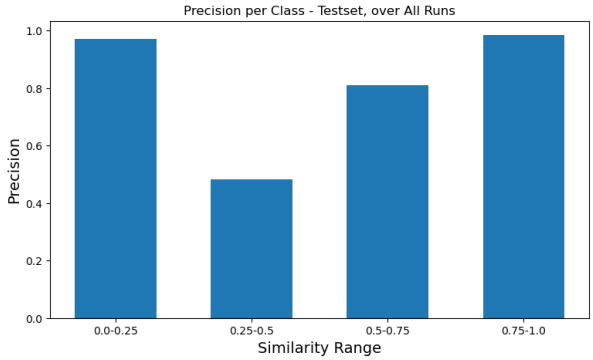

Figure 4.7: Mean precision per class, Test, Q1.

Figure 4.8: Mean recall per class, Train, Q1.

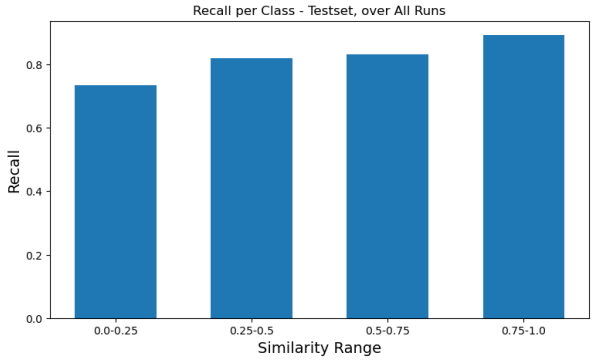

Figure 4.9: Mean recall per class, Test, Q1.

Results Q2

Runs	MSE		MAE		Accuracy		F1		Precision		Recall	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Run 1	.0175	.0165	.0974	.1005	.703	.736	.704	.736	.703	.736	.729	.770
Run 2	.0160	.0148	.0950	.0921	.722	.760	.721	.761	.722	.760	.749	.767
Run 3	.0175	.0159	.0996	.0980	.690	.712	.689	.714	.690	.712	.720	.731
Run 4	.0176	.0128	.0994	.0864	.690	.784	.691	.786	.690	.784	.708	.813
Run 5	.0163	.0159	.0965	.0944	.717	.744	.716	.741	.717	.744	.736	.756
Run 6	.0138	.0242	.0920	.1072	.746	.696	.746	.690	.746	.696	.777	.705
Run 7	.0162	.0124	.0946	.0872	.730	.792	.729	.796	.730	.792	.741	.807
Run 8	.0185	.0127	.0996	.0899	.719	.768	.715	.768	.719	.768	.764	.776
Run 9	.0167	.0188	.0982	.1019	.714	.712	.716	.712	.714	.712	.742	.723
Run 10	.0184	.0144	.1016	.0917	.701	.776	.702	.776	.701	.776	.718	.798
Averages	.0168	.0158	.0974	.0949	.713	.748	.713	.748	.713	.748	.738	.765
St.devs	.0013	.0034	.0027	.0065	.0168	0.0317	0.017	0.033	0.017	0.032	0.020	0.035

Table 4.3: FEM Dataset: Random-Split Performance by various metrics, Q2.

Figure 4.10: Mean F1 per class, Train, Q2.

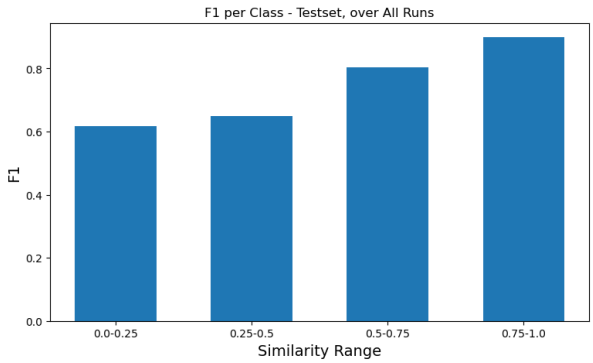

Figure 4.11: Mean F1 per class, Test, Q2.

$$\begin{bmatrix} 62.6 \pm 5.12 & 54.7 \pm 3.8 & 0.5 \pm 0.5 & 1.6 \pm 0.66 \\ 19.0 \pm 4.9 & 88.8 \pm 6.05 & 7.8 \pm 1.78 & 0.0 \pm 0.0 \\ 0.8 \pm 0.6 & 12.6 \pm 2.15 & 60.2 \pm 3.06 & 1.8 \pm 1.25 \\ 0.0 \pm 0.0 & 1.0 \pm 0.77 & 7.5 \pm 3.01 & 55.1 \pm 3.86 \end{bmatrix}$$

Figure 4.12: Confusion Matrix, Train, Q2.

$$\begin{bmatrix} 20.6 \pm 4.27 & 14.6 \pm 3.14 & 0.1 \pm 0.3 & 0.3 \pm 0.64 \\ 5.8 \pm 2.04 & 30.7 \pm 4.65 & 2.9 \pm 1.7 & 0.0 \pm 0.0 \\ 0.2 \pm 0.4 & 4.0 \pm 1.0 & 21.6 \pm 1.43 & 0.8 \pm 1.08 \\ 0.0 \pm 0.0 & 0.4 \pm 0.49 & 2.4 \pm 1.11 & 20.6 \pm 3.64 \end{bmatrix}$$

Figure 4.13: Confusion Matrix, Test, Q2.

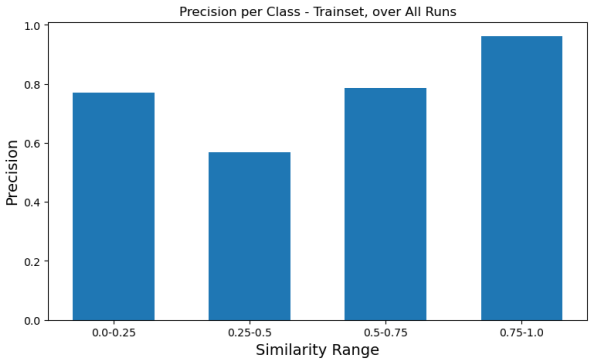

Figure 4.14: Mean precision per class, Train, Q2.

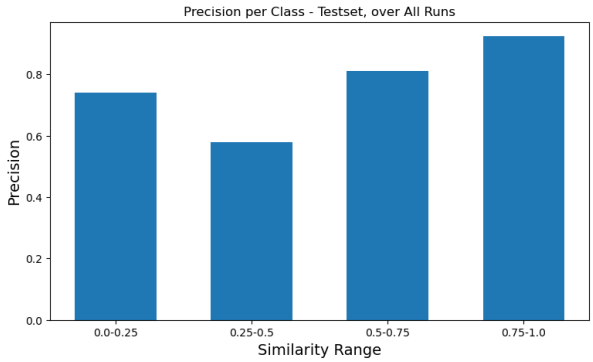

Figure 4.15: Mean precision per class, Test, Q2.

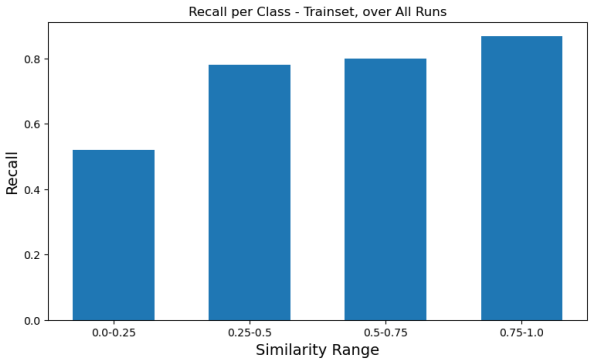

Figure 4.16: Mean recall per class, Train, Q2.

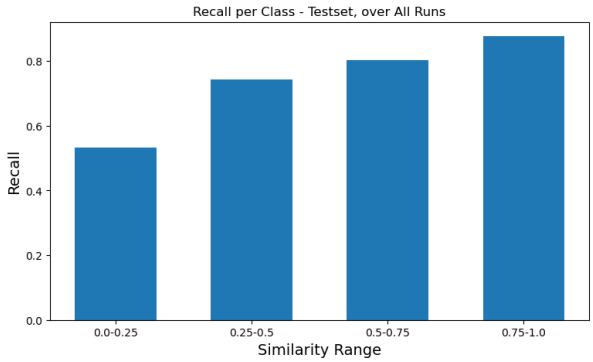

Figure 4.17: Mean recall per class, Test, Q2.

$$\begin{bmatrix} 131.1 \pm 3.21 & 38.3 \pm 3.82 & 0.8 \pm 0.75 & 0.2 \pm 0.4 \\ 4.8 \pm 1.89 & 66.9 \pm 3.83 & 6.0 \pm 1.1 & 0.0 \pm 0.0 \\ 0.0 \pm 0.0 & 3.2 \pm 1.47 & 48.1 \pm 2.84 & 0.3 \pm 0.46 \\ 0.0 \pm 0.0 & 1.0 \pm 0.63 & 6.5 \pm 1.28 & 66.8 \pm 2.68 \end{bmatrix}$$

Figure 4.18: Confusion Matrix, Train, Q3.

Results Q3

Runs	MSE		MAE		Accuracy		F1		Precision		Recall	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Run 1	.0108	.0100	.0729	.0719	.834	.816	.841	.825	.834	.816	.873	.864
Run 2	.0096	.0097	.0677	.0745	.853	.832	.858	.840	.853	.832	.879	.872
Run 3	.0109	.0085	.0732	.0678	.821	.856	.830	.859	.821	.856	.870	.884
Run 4	.0104	.0113	.0713	.0781	.829	.792	.837	.802	.829	.792	.871	.850
Run 5	.0106	.0081	.0710	.0667	.824	.848	.833	.850	.824	.848	.863	.859
Run 6	.0086	.0145	.0675	.0859	.837	.784	.843	.796	.837	.784	.865	.840
Run 7	.0093	.0138	.0703	.0829	.834	.792	.842	.799	.834	.792	.880	.839
Run 8	.0095	.0082	.0661	.0693	.861	.872	.866	.876	.861	.872	.888	.895
Run 9	.0087	.0111	.0693	.0698	.832	.848	.838	.854	.832	.848	.858	.880
Run 10	.0104	.0100	.0726	.0724	.842	.824	.848	.833	.842	.824	.878	.858
Averages	.0099	.0105	.0702	.0739	.837	.826	.843	.833	.837	.826	.872	.864
Stdevs	.0008	.0021	.0023	.0061	.0119	.0286	0.011	0.026	0.012	0.029	0.009	0.018

Table 4.4: FEM Dataset: Random-Split Performance by various metrics, Q3.

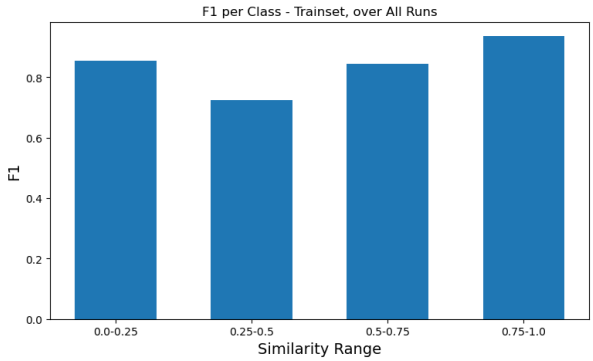

Figure 4.19: Mean F1 per class, Train, Q3.

Figure 4.20: Mean F1 per class, Test, Q3.

$$\begin{bmatrix} 42.3 \pm 4.8 & 12.8 \pm 1.66 & 0.5 \pm 0.81 & 0.0 \pm 0.0 \\ 1.8 \pm 0.98 & 24.1 \pm 4.09 & 2.4 \pm 0.8 & 0.0 \pm 0.0 \\ 0.0 \pm 0.0 & 1.3 \pm 1.35 & 14.0 \pm 2.61 & 0.1 \pm 0.3 \\ 0.0 \pm 0.0 & 0.2 \pm 0.4 & 2.6 \pm 1.36 & 22.9 \pm 2.3 \end{bmatrix}$$

Figure 4.21: Confusion Matrix, Test, Q3.

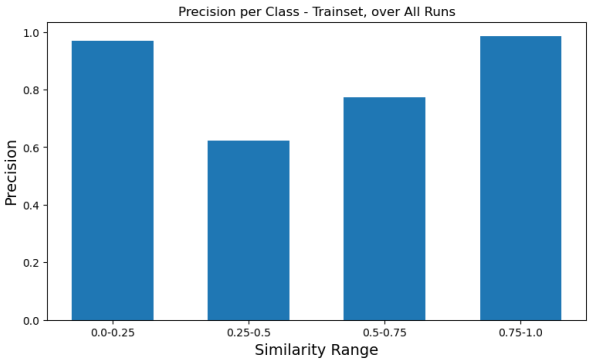

Figure 4.22: Mean precision per class, Train, Q3.

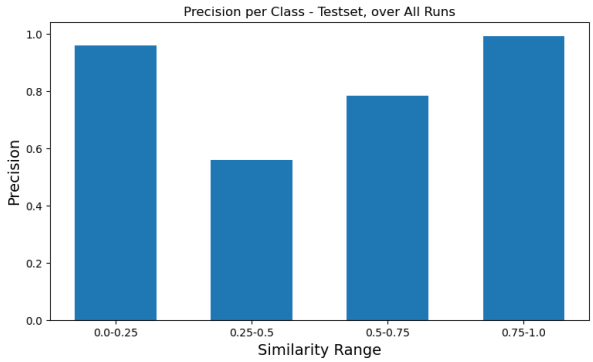

Figure 4.23: Mean precision per class, Test, Q3.

Figure 4.24: Mean recall per class, Train, Q3

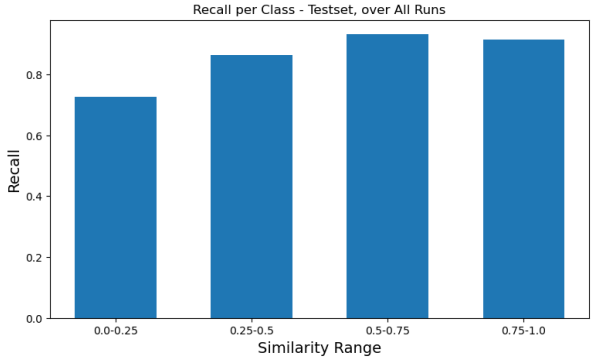

Figure 4.25: Mean recall per class, Test, Q3

Discussion of Results Q1

Figures 4.2 through 4.9 and Table 4.2 reflect the performance of the network concerning Q1.

One can observe that the accuracy is roughly 80% on both training and test sets. Earlier experiments in which the neural network was fitted to annotations provided by a single rater resulted in significantly higher accuracy scores in excess of 90%. I suspect that this drop is due to the low overall agreement among the raters. As pointed out in chapter 3, there is substantial disagreement among all subsets of raters, including the raters who are considered to be experts. One can imagine that raters who hold different views may provide different ratings for similar samples, which implies that the neural network is faced with conflicting logic, resulting in more incorrect classifications. Another interesting observation is the low average precision achieved for the second class.(values in the interval [0.25,0.5]). This is further supported by the confusion matrices for both training and test sets, which show that many samples which belong to class 1 were wrongly assigned to class 2, 40.7 samples for the training set, and

13.5 for the test set.

Discussion of Results Q2

Figures 4.10 through 4.17 and Table 4.3 reflect the performance of the network for the second question.

Compared to the results obtained for Q1, one finds that the average accuracy on both training and test sets are significantly lower, just over 70%. The average MAE scores for both the train and test sets are in accordance with this, the average MAE was about 0.074 for Q1 and 0.96 for Q2, which is quite a difference, especially when one considers that the similarity classes have a margin around the bin centers of 0.125. This is further evidence that the rate of agreement among the raters for a given dataset and the performance of a neural network fitted to that dataset are strongly correlated. When one compares the confusion matrices for Q1 and Q2, one finds that the recall for class 1 is still low combined with a low precision for the second class, but moreover, contrary to Q1, for Q2 one finds that the precision and f1 scores of class 1 are also quite low. This is not simply due to the fact that the relative number of samples belonging to class 1 that were classified as belonging to class 2 is higher (entry (0,1): 61.4 correct vs 54.7 incorrect for train set), but also 20% - 30% of the assignments to the first class actually belong to class 2 (entry (1,0) 17.7 samples for train set, 6.9 for test set).

Discussion of Results Q3

Figures 4.18 through 4.25 and Table 4.4 reflect the performance of the network for the third question.

The neural network performed best for Q3, and this is supported by almost all scores. However, the difference in performance between Q1 and Q3 is small, the accuracy is slightly higher for Q3 and the MAE slightly lower. Once more, one can easily verify that the distinction between class 1 and 2 is most problematic. Furthermore, one can verify that the distinction between class 3 and classes 2 and 4 is somewhat problematic for the neural network, although less than the distinction between classes 1 and 2. All in all, these results support the suspicion that the rate of agreement and the performance of the neural network are strongly correlated, and show that the distinction between class 1 and 2 are most problematic. In fact, the performance of the neural network across the different questions follows the same trend as the overall agreement, the performance is best for Q3, followed by Q1, and significantly worse for Q2. In general, one can argue that the network is somewhat biased towards the middle of the similarity spectrum, that is, it is more likely to wrongly classify very similar samples as being less similar and very dissimilar samples as more similar, than it is to classify similar as very similar or dissimilar as very dissimilar. This could be due to there being more disagreement between these adjacent classes

(1 and 2, and 3 and 4). This is something that could be investigated in future work.

4.5.4 Leave-One-Crack-Out

This section presents the results of the leave one crack out experiments for the FEM dataset. This dataset contains eight different crack archetypes, 18, 20, 21, 23, 24, 101, 102 and 103. First, the set-up of the experiments is described. Secondly, the results are presented for all three questions, followed by an interpretation of the results.

Experimental Set-Up

This experiment consists of eight sub-experiments, one for each crack archetype present in the dataset, namely: 18, 20, 21, 23, 24, 101, 102 and 103.. For each of these sub-experiments, the samples are assigned to the training and test sets as follows. If a sample contains one or two cracks that belongs to the crack archetype to be left out, assign it to the testset, otherwise, assign it to the training set. Each sub-experiment is repeated three times, and the results are averaged in order to obtain reliable estimates of the performance. Furthermore, the entire experiment is conducted three times, once for each of the three questions. In the following, the results are shown for each question, separately. For each of the questions, the average accuracies obtained on train and test sets are shown for each crack archetype, as well as the average MAE and MSE. These averages are taken over the three runs. This yields one bar chart and one table for each question.

Results Q1, Q2 and Q3

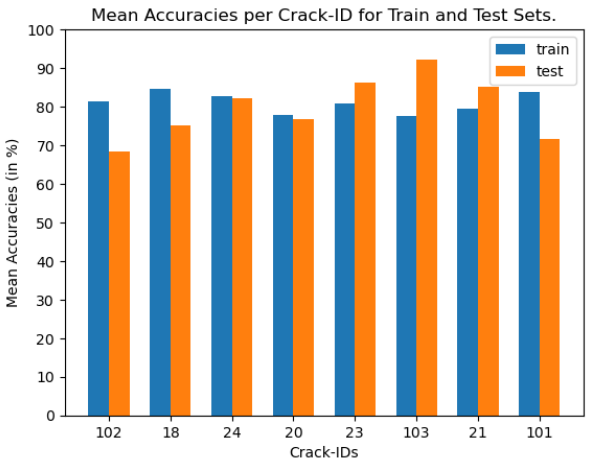

Figure 4.26: Mean Train/Test Accuracies per Crack

Crack-ID	MSE		MAE	
	Train	Test	Train	Test
102	.0097	.0144	.0727	.0966
18	.0088	.0109	.0684	.0801
24	.0088	.0098	.0690	.0686
20	.0111	.0121	.0805	.0819
23	.0094	.0083	.0713	.0634
103	.0110	.0052	.0787	.0531
21	.0107	.0064	.0765	.0615
101	.0088	.0117	.0675	.0866

Table 4.5: MAE/MSE - Leave out Crack, Q1, FEM

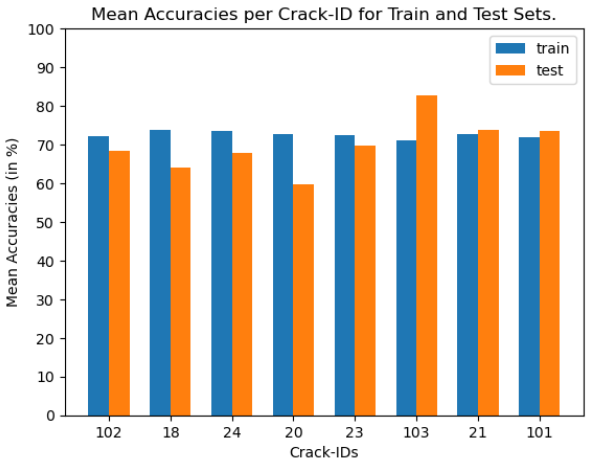

Figure 4.27: Mean Train/Test Accuracies per Crack

Crack-ID	MSE		MAE	
	Train	Test	Train	Test
102	.0173	.0139	.0970	.0945
18	.0142	.0243	.0912	.1158
24	.0163	.0157	.0945	.0996
20	.0138	.0300	.0919	.1277
23	.0171	.0161	.0959	.1006
103	.0175	.0085	.0995	.0703
21	.0170	.0137	.0965	.0910
101	.0173	.0141	.0980	.0941

Table 4.6: MAE/MSE - Leave out Crack, Q2, FEM

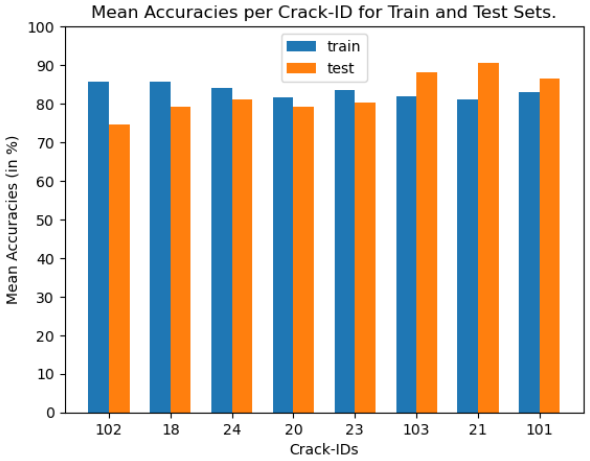

Figure 4.28: Mean Train/Test Accuracies per Crack

Crack-ID	MSE		MAE	
	Train	Test	Train	Test
102	.0097	.0114	.0674	.0864
18	.0088	.0123	.0675	.0772
24	.0097	.0106	.0714	.0734
20	.0093	.0158	.0723	.0848
23	.0102	.0106	.0713	.0761
103	.0110	.0058	.0755	.0530
21	.0108	.0062	.0744	.0554
101	.0100	.0089	.0689	.0773

Table 4.7: MAE/MSE - Leave out Crack, Q3, FEM

Discussion of Results

Figures 4.26 through 4.28 and tables 4.5 through 4.7 are shown side-by-side, and reflect the performance of the network for the leave one crack out experiments for the FEM dataset on all three questions, separately.

The main aim of these experiments is to determine whether there is a universal logic that underlies the assessment of the similarity for different pairs of crack archetypes. For example, one may wonder whether the process of assessing the

similarities between cracked facades of archetypes 18 and 101 is similar to assessing the degree of similarity between cracked facades belonging to archetypes 23 and 24.

The results clearly show that this is the case to some extent, because the test accuracies are generally not far lower than the train accuracies. In fact, in quite a few cases, the test accuracy even exceeds the train accuracy, which is very surprising. Especially for crack archetype 103, the test accuracy exceeds the train accuracy by around 15%, which is remarkable.

Interestingly, the test accuracy exceeds the train accuracy for crack archetypes 103 and 21 across all three questions, while for crack archetype 23 this is only the case for the first question and for crack archetype 101 it holds true for the second and third question. It would be interesting to further investigate what underlies the phenomenon that the test accuracy sometimes exceeds the train accuracy.

4.5.5 Leave One Class Out

This section presents the results of the leave one class out experiments on the FEM dataset.

Experimental Set-Up

This experiment consists of four sub-experiments, one for each similarity class present in the dataset, namely: {**0-0.25, 0.25-0.50, 0.50-0.75, 0.75-1.00**}. For each of these sub-experiments, the samples are assigned to the training and test sets as follows. If a sample contains belongs to the crack similarity class to be left out, assign it to the testset, otherwise, assign it to the training set. Each sub-experiment is run three times from scratch, and the results are averaged in order to obtain reliable estimates of the performance. This entire experiment is, of course, conducted three times, once for each of the three questions.

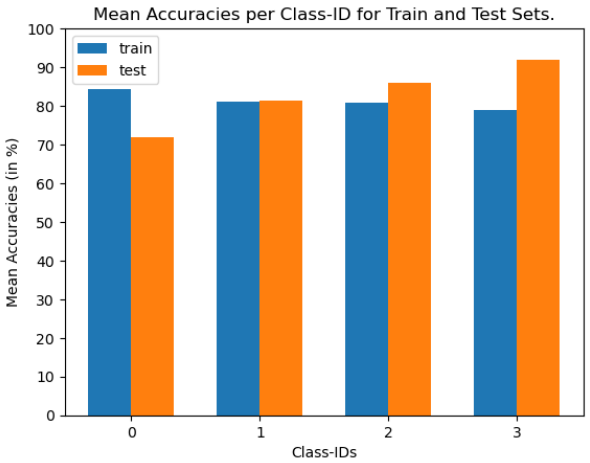

Figure 4.29: Mean Train/Test Accuracies per Crack

Class-ID	MSE		MAE	
	Train	Test	Train	Test
0	.0080	.0135	.0644	.0893
1	.0099	.0093	.0727	.0738
2	.0096	.0067	.0728	.0620
3	.0099	.0064	.0748	.0546

Table 4.8: MAE/MSE - Leave out Class, Q1, FEM

Results Q1, Q2 and Q3

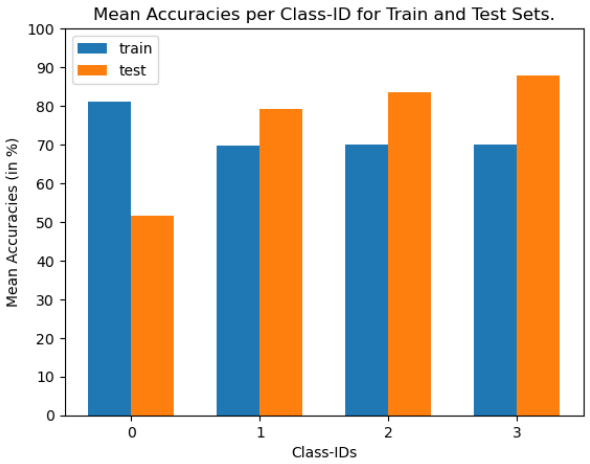

Figure 4.30: Mean Train/Test Accuracies per Crack

Class-ID	MSE		MAE	
	Train	Test	Train	Test
0	.0115	.0270	.0785	.1344
1	.0181	.0100	.1003	.0812
2	.0170	.0102	.1006	.0713
3	.0165	.0112	.0995	.0688

Table 4.9: MAE/MSE - Leave out Class, Q2, FEM

Discussion of Results

Figures 4.29 through 4.31 and tables 4.8 through 4.10 are shown side-by-side, and reflect the performance of the network for the leave one class out experiments for the FEM dataset on all three questions, separately.

The same phenomenon that was observed in the leave one crack out experiments, can also be observed in the results for the leave one class out experiment. Which

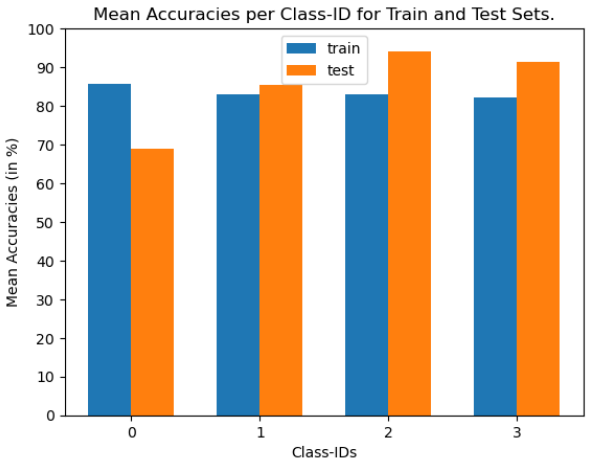

Figure 4.31: Mean Train/Test Accuracies per Crack

Class-ID	MSE		MAE	
	Train	Test	Train	Test
0	.0087	.0172	.0667	.1006
1	.0105	.0071	.0723	.0630
2	.0100	.0038	.0720	.0425
3	.0096	.0077	.0704	.0575

Table 4.10: MAE/MSE - Leave out Class, Q3, FEM

is that test accuracies sometimes exceed train accuracies. In fact, it occurs even more frequently here.

One can verify that the network generalizes excellently to different similarity classes, the test accuracy exceeds the train accuracy across all three questions when the second, third or fourth similarity class is left out (in order of increasing similarity).

However, the lowest similarity class is a big exception to this rule. One can see that the train accuracy is considerably higher than the test accuracy when the lowest similarity class is left out, across all three questions.

All in all, the results that the process of assessing the similarity between cracked facades is similar across different degrees of similarity. That is, assessing the similarity of a pair of cracked facades that lean towards the similar end of the spectrum, is comparable to assessing a pair of cracked facades that are less similar. However, the poor results that one obtains when leaving out samples belonging to the lowest similarity class points oot caution. Further research is needed to identify the cause of this strange result.

4.5.6 FEM: Characteristics of Misclassified Samples

This section provides an analysis of the samples for which the network performed poorly, that is the samples for which the NN's predictions were farthest off from the true similarites, in general. Separate analyses are shown for each of the three questions. Tables 4.11 through 4.13 show which pairs of crack archetypes were most difficult to accurately predict for the network.

Archetypes	Occurrences	Total Occurrences
(18, 102)	9	14
(18, 21)	9	23
(23, 24)	8	23
(18, 20)	8	23
(20, 101)	7	14
(24, 102)	6	10
(20, 102)	6	14
(20, 20)	4	13
(18, 101)	4	14
(21, 101)	4	11
(18, 24)	4	14
(103, 103)	3	14
(101, 102)	3	15
(24, 101)	3	11
(23, 102)	3	12
(23, 103)	2	11
(20, 24)	2	14
(101, 101)	2	15
(18, 23)	2	14
(20, 21)	2	23
(20, 23)	1	14
(23, 101)	1	10
(24, 103)	1	23
(101, 103)	1	11
(24, 24)	1	12
(23, 23)	1	13
(21, 102)	1	11
(18, 18)	1	15
(18, 103)	1	14

Table 4.11: FEM Dataset, Q1: All pairs of crack archetypes that occur (and how often) in the 100 samples most poorly predicted by the network, and how often these crack pairs occur in the entire dataset (500 samples)

Archetypes	Occurrences	Total Occurrences
(18, 20)	12	23
(20, 24)	6	14
(18, 21)	6	23
(18, 23)	5	14
(24, 102)	5	10
(18, 24)	5	14
(20, 20)	4	13
(21, 24)	4	10
(23, 24)	4	23
(20, 23)	4	14
(20, 101)	4	14
(23, 101)	3	10
(101, 103)	3	11
(18, 102)	3	14
(23, 103)	3	11
(20, 21)	3	23
(20, 102)	3	14
(24, 101)	3	11
(21, 102)	3	11
(23, 102)	2	12
(20, 103)	2	15
(18, 101)	2	14
(24, 103)	2	23
(21, 103)	2	10
(18, 103)	2	14
(101, 102)	1	15
(21, 23)	1	11
(21, 101)	1	11
(101, 101)	1	15
(102, 103)	1	4

Table 4.12: FEM Dataset, Q2: All pairs of crack archetypes that occur (and how often) in the 100 samples most poorly predicted by the network, and how often these crack pairs occur in the entire dataset (500 samples)

Archetypes	Occurrences	Total Occurrences
(18, 20)	8	23
(23, 24)	8	23
(18, 102)	7	14
(24, 102)	6	10
(20, 102)	6	14
(101, 103)	6	11
(18, 21)	5	23
(23, 102)	5	12
(20, 20)	4	13
(18, 101)	4	14
(20, 101)	4	14
(18, 24)	3	14
(23, 103)	3	11
(24, 103)	3	23
(21, 24)	3	10
(21, 101)	3	11
(103, 103)	2	14
(23, 101)	2	10
(20, 24)	2	14
(18, 18)	2	15
(20, 23)	2	14
(18, 23)	2	14
(101, 102)	2	15
(20, 21)	1	23
(24, 101)	1	11
(20, 103)	1	15
(21, 23)	1	11
(102, 102)	1	12
(23, 23)	1	13
(21, 102)	1	11
(101, 101)	1	15

Table 4.13: FEM Dataset, Q3: All pairs of crack archetypes that occur (and how often) in the 100 samples most poorly predicted by the network, and how often these crack pairs occur in the entire dataset (500 samples)

When comparing Tables 4.11 through 4.13, one notices that the results of the three different questions are somewhat similar, but there are some differences. While pair (18, 102) can be viewed as a problematic pair for the first question, 9 out of the 14 are among the 100 most poorly classified, for question 2 only 3 out of the 14 are poorly classified. On the other hand, pair (18, 20) seems to be a problematic pair for all three questions. Pairs (18, 20), (24, 102), (18, 24), (20, 20), (20, 102) and (18, 102) occur more frequently among the 100 most problematic samples than is to be expected, based purely on how often these pairs occur in total throughout the dataset.

It is instructive to compare the results for each question to Table 3.7, since such a comparison may reveal a correlation between the pairs that are hardest

to predict for the network, and the pairs for which the agreement is generally lowest. We find that some pairs that are in the top portion of Table 3.7, such as (23,24), also occur in the top segment of at least one of Tables 4.11 through 4.13, while this is not true for other pairs (e.g. (24, 103)). In relation to this discrepancy between the problematic pairs for the three questions compared to the pairs featured in Table 3.7, it is instructive to note that Table 3.7 the per sample agreement is calculated over all three questions, rather than per question.

4.6 Markov-Walk Dataset Experiments

In this section I report on neural network experiments that were performed on the Markov-Walk dataset. All relevant subtopics are presented in the same order as was done for the FEM dataset.

4.6.1 Experimental Set-Up

The set-up of the random-split experiments is the same as for the FEM-dataset. The only differences are that this dataset is considerably bigger, consisting of 2466 samples instead of 499, and that it involves 7 similarity classes instead of 4, namely: {**0-0.1825, 0.1825-0.3175, 0.3175-0.4325, 0.4325-0.5625, 0.5625-0.6875, 0.6875-0.8125, 0.8125 - 1.0**}.

4.6.2 Markov-Walk Dataset Annotations

This section reports on the annotations obtained for the samples of the Markov-Walk dataset. Table 4.8 shows the distribution of the true similarity scores for all three questions, before and after conversion with the hybrid approach.

One can observe that there are far more similarity intervals after converting the assessments into single annotations per sample than is the case for the FEM dataset. This is not surprising, given that this dataset contains significantly more samples, 2466 rather than 499, and considerably more raters are involved here, 28 instead of 7.

I have postprocessed these annotations by narrowing the original similarity intervals and introducing new similarity intervals to account for boundary cases. This resulted in the following similarity intervals: {**0-0.1875, 0.1875-0.3125, 0.3125-0.4375, 0.4375-0.5625, 0.5625-0.6875, 0.6875-0.8125, 0.8125-1.0**}. Thus, all true similarities were converted to the nearest value from the following, {**.125, .25, .375, .5, .625, .75, .875**}. The resulting set of similarity intervals along with the number of corresponding samples for each interval, are shown below:

Similarities	Q1		Q2		Q3	
	Before	After	Before	After	Before	After
0.125	333	333	229	229	262	262
0.25	50	50	29	29	49	49
0.325	2	0	2	0	4	0
0.375	615	618	868	875	795	804
0.425	1	0	5	0	5	0
0.458	9	0	13	0	3	0
0.5	23	45	45	75	39	52
0.542	13	0	17	0	10	0
0.575	3	0	6	0	4	0
0.625	747	756	735	745	805	813
0.675	6	0	4	0	4	0
0.696	0	0	1	0	1	0
0.75	84	84	51	52	56	57
0.875	580	580	461	461	429	429

Table 4.14: Markov-Walk Dataset: True similarities before and after handling of divergent similarity intervals.

- **0.0 - 0.1875**: Center: 0.125
- **0.1875 - 0.3125**: Center: 0.25
- **0.3125 - 0.4375**: Center: 0.375
- **0.4375 - 0.5625**: Center: 0.5
- **0.5625 - 0.6875**: Center: 0.625
- **0.6875 - 0.8125**: Center: 0.75
- **0.8125 - 1.0**: Center: 0.875

Note that the centers (true similarities) of the lowest and highest similarity intervals were not modified. All in all, the true similarities of only 36 of the 2466 samples have been affected by this procedure. Next, each of the three experiments will be discussed, one by one.

4.6.3 Preparing Cracked Facades for Training

By contrast to the FEM dataset, all cracked facades associated with this dataset have the same size, 100 pixels high and 333 pixels wide. The same procedure used for the FEM cracked facades was applied here to convert the cracked facades to numpy matrices of size 81 by 161. There are two motivations for this. First of all, propagating large numpy matrices through the network is both memory and cpu intensive. Secondly, converting all cracked facades to the same size allows one to train and test the network on cracked facades from both datasets.

$$\begin{bmatrix} 55.6 \pm 8.6 & 104.8 \pm 14.18 & 60.6 \pm 4.43 & 21.2 \pm 2.93 & 5.0 \pm 2.68 & 1.6 \pm 1.11 & 1.8 \pm 0.87 \\ 2.6 \pm 1.28 & 24.5 \pm 2.69 & 7.6 \pm 2.42 & 1.2 \pm 0.87 & 0.2 \pm 0.4 & 0.5 \pm 0.5 & 0.0 \pm 0.0 \\ 61.8 \pm 16.39 & 164.2 \pm 24.96 & 131.0 \pm 12.55 & 73.2 \pm 7.57 & 29.4 \pm 4.13 & 9.2 \pm 2.18 & 1.2 \pm 0.75 \\ 0.0 \pm 0.0 & 0.6 \pm 0.49 & 2.6 \pm 0.92 & 23.8 \pm 2.14 & 5.2 \pm 1.33 & 0.1 \pm 0.3 & 0.0 \pm 0.0 \\ 2.6 \pm 1.2 & 8.7 \pm 2.37 & 38.3 \pm 6.36 & 123.3 \pm 10.39 & 207.8 \pm 12.0 & 141.5 \pm 8.42 & 45.6 \pm 7.77 \\ 0.0 \pm 0.0 & 0.0 \pm 0.0 & 0.0 \pm 0.0 & 1.5 \pm 0.81 & 8.8 \pm 2.14 & 40.0 \pm 2.19 & 9.8 \pm 2.64 \\ 0.0 \pm 0.0 & 0.1 \pm 0.3 & 6.3 \pm 2.0 & 25.7 \pm 3.29 & 63.1 \pm 7.35 & 150.3 \pm 10.79 & 186.1 \pm 10.38 \end{bmatrix}$$

Figure 4.32: Confusion Matrix, Train, Q1.

$$\begin{bmatrix} 19.4 \pm 3.29 & 31.0 \pm 6.13 & 22.1 \pm 3.67 & 8.0 \pm 2.57 & 1.0 \pm 0.77 & 0.3 \pm 0.64 & 0.6 \pm 0.66 \\ 0.8 \pm 0.6 & 9.5 \pm 2.33 & 2.3 \pm 1.68 & 0.5 \pm 0.5 & 0.2 \pm 0.4 & 0.1 \pm 0.3 & 0.0 \pm 0.0 \\ 17.2 \pm 5.33 & 53.2 \pm 8.21 & 39.3 \pm 5.25 & 23.0 \pm 5.71 & 10.8 \pm 1.94 & 4.0 \pm 2.68 & 0.5 \pm 0.5 \\ 0.0 \pm 0.0 & 0.3 \pm 0.46 & 1.4 \pm 1.11 & 8.7 \pm 1.73 & 2.2 \pm 1.4 & 0.1 \pm 0.3 & 0.0 \pm 0.0 \\ 1.1 \pm 0.94 & 2.3 \pm 1.9 & 14.4 \pm 3.69 & 41.9 \pm 7.2 & 65.1 \pm 6.53 & 48.0 \pm 4.56 & 15.4 \pm 4.18 \\ 0.0 \pm 0.0 & 0.0 \pm 0.0 & 0.0 \pm 0.0 & 0.6 \pm 0.49 & 2.8 \pm 1.66 & 16.4 \pm 3.77 & 4.1 \pm 1.81 \\ 0.0 \pm 0.0 & 0.1 \pm 0.3 & 2.7 \pm 1.79 & 7.7 \pm 3.38 & 19.0 \pm 5.0 & 53.1 \pm 6.58 & 65.8 \pm 7.51 \end{bmatrix}$$

Figure 4.33: Confusion Matrix, Test, Q1.

4.6.4 Random-Split

The results obtained for the random-split experiments for all questions are discussed in this section. The structure of this section is similar to that of the random-split section that covers the FEM dataset.

Results Q1

Runs	MSE		MAE		Accuracy		F1		Precision		Recall	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Run 1	.0272	.0316	.1273	.1394	.357	.324	.404	.365	.357	.324	.539	.502
Run 2	.0237	.0241	.1197	.1202	.357	.379	.416	.420	.357	.379	.588	.563
Run 3	.0266	.0218	.1265	.1144	.353	.371	.406	.423	.353	.371	.559	.592
Run 4	.0238	.0210	.1192	.1156	.377	.350	.438	.402	.377	.350	.614	.558
Run 5	.0249	.0230	.1224	.1191	.360	.381	.418	.430	.360	.381	.590	.587
Run 6	.0231	.0244	.1184	.1216	.365	.355	.421	.406	.365	.355	.592	.584
Run 7	.0236	.0241	.1197	.1184	.365	.361	.424	.408	.365	.361	.590	.560
Run 8	.0244	.0244	.1221	.1172	.354	.384	.409	.441	.354	.384	.571	.604
Run 9	.0226	.0225	.1171	.1165	.363	.365	.416	.416	.363	.365	.582	.594
Run 10	.0222	.0245	.1167	.1194	.367	.363	.423	.415	.367	.363	.596	.585
Averages	.0242	.0241	.1209	.1202	.362	.363	.417	.413	.362	.363	.582	.573
Stdevs	.0015	.0027	.0035	.0067	.0068	.0169	0.010	0.019	0.007	0.017	0.020	0.028

Table 4.15: Markov-Walk Dataset: Random-Split Performance by various metrics, Q1.

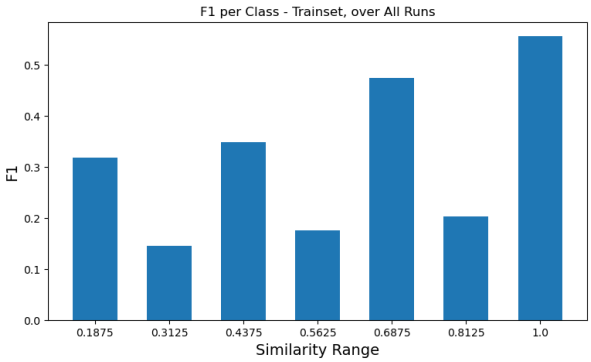

Figure 4.34: Mean F1 per class, Train, Q1.

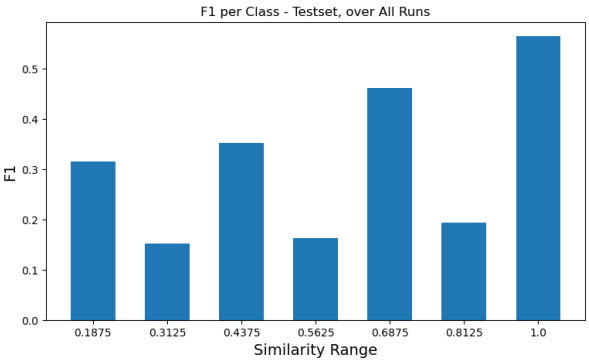

Figure 4.35: Mean F1 per class, Test, Q1.

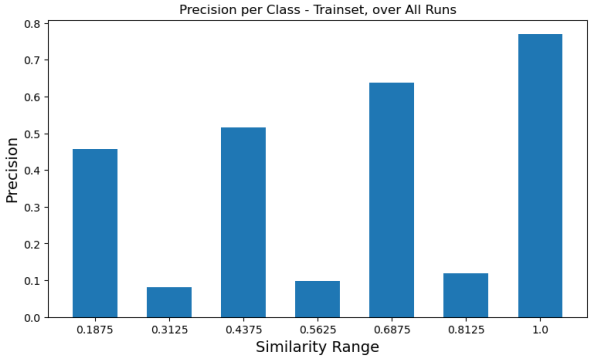

Figure 4.36: Mean precision per class, Train, Q1.

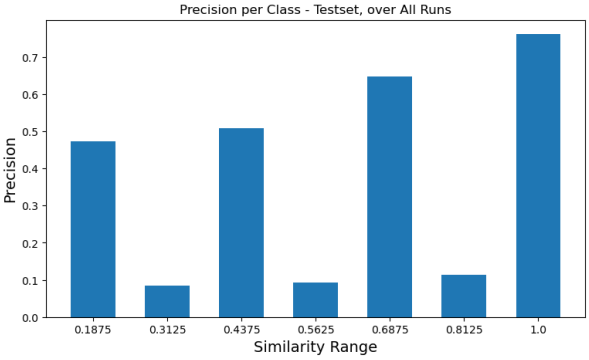

Figure 4.37: Mean precision per class, Test, Q1.

Figure 4.38: Mean recall per class, Train, Q1

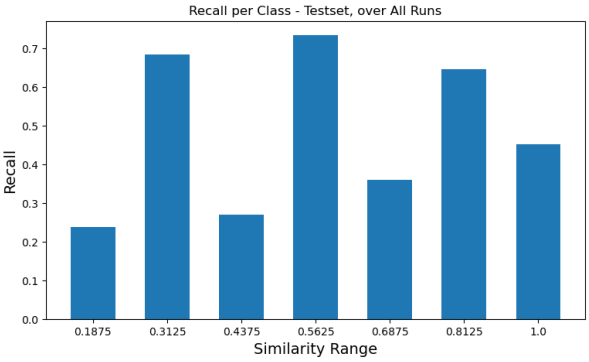

Figure 4.39: Mean recall per class, Test, Q1

Results Q2

Runs	MSE		MAE		Accuracy		F1		Precision		Recall	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Run 1	.0365	.0342	.1483	.1444	.294	.316	.338	.368	.294	.316	.475	.519
Run 2	.0297	.0308	.1353	.1375	.309	.326	.363	.368	.309	.326	.520	.531
Run 3	.0353	.0336	.1463	.1465	.301	.290	.351	.331	.301	.290	.500	.471
Run 4	.0321	.0289	.1399	.1347	.320	.319	.371	.361	.320	.319	.523	.509
Run 5	.0349	.0335	.1474	.1421	.293	.314	.334	.352	.293	.314	.464	.477
Run 6	.0319	.0283	.1389	.1298	.312	.345	.368	.394	.312	.345	.532	.551
Run 7	.0355	.0348	.1470	.1442	.299	.337	.342	.392	.299	.337	.486	.542
Run 8	.0335	.0321	.1426	.1426	.319	.284	.367	.323	.319	.284	.515	.459
Run 9	.0327	.0317	.1413	.1374	.319	.311	.369	.359	.319	.311	.526	.504
Run 10	.0334	.0355	.1427	.1477	.308	.292	.359	.336	.308	.292	.514	.462
Averages	.0336	.0323	.1430	.1407	.307	.313	.356	.358	.307	.313	.505	.502
Stdevs	.0019	.0023	.0040	.0054	.0097	.0192	.013	.023	.010	.019	.022	.032

Table 4.16: Markov-Walk Dataset: Random-Split Performance by various metrics, Q2.

$$\begin{bmatrix} 31.4 \pm 6.26 & 78.8 \pm 5.79 & 36.8 \pm 5.53 & 16.8 \pm 1.33 & 5.5 \pm 2.73 & 1.4 \pm 0.49 & 1.4 \pm 0.49 \\ 1.1 \pm 1.14 & 15.2 \pm 2.09 & 3.3 \pm 1.19 & 1.8 \pm 0.75 & 0.2 \pm 0.4 & 0.0 \pm 0.0 & 0.0 \pm 0.0 \\ 35.2 \pm 12.14 & 131.2 \pm 12.32 & 163.7 \pm 20.05 & 182.6 \pm 9.7 & 99.9 \pm 7.34 & 36.8 \pm 6.69 & 12.2 \pm 2.82 \\ 0.3 \pm 0.46 & 1.7 \pm 1.42 & 16.0 \pm 2.28 & 28.0 \pm 3.9 & 7.1 \pm 1.76 & 2.5 \pm 1.02 & 0.0 \pm 0.0 \\ 10.1 \pm 4.25 & 39.0 \pm 5.22 & 69.8 \pm 6.18 & 123.5 \pm 6.48 & 150.7 \pm 8.22 & 108.7 \pm 7.13 & 59.1 \pm 9.27 \\ 0.0 \pm 0.0 & 0.1 \pm 0.3 & 0.4 \pm 0.66 & 1.7 \pm 0.9 & 4.7 \pm 1.55 & 25.9 \pm 3.91 & 3.7 \pm 2.05 \\ 0.8 \pm 0.98 & 2.8 \pm 1.08 & 8.2 \pm 2.82 & 21.4 \pm 4.13 & 51.7 \pm 5.2 & 102.4 \pm 6.3 & 153.4 \pm 6.14 \end{bmatrix}$$

Figure 4.40: Confusion Matrix, Train, Q2.

$$\begin{bmatrix} 12.5 \pm 3.23 & 24.0 \pm 3.38 & 12.1 \pm 2.34 & 6.3 \pm 1.9 & 1.3 \pm 1.0 & 0.3 \pm 0.46 & 0.4 \pm 0.49 \\ 0.3 \pm 0.46 & 5.3 \pm 1.95 & 0.9 \pm 0.7 & 0.7 \pm 0.78 & 0.2 \pm 0.6 & 0.0 \pm 0.0 & 0.0 \pm 0.0 \\ 12.8 \pm 5.19 & 43.0 \pm 4.29 & 54.2 \pm 8.34 & 57.0 \pm 4.12 & 31.5 \pm 5.64 & 11.3 \pm 1.49 & 3.6 \pm 1.2 \\ 0.0 \pm 0.0 & 0.6 \pm 0.66 & 5.9 \pm 2.21 & 10.3 \pm 3.32 & 2.2 \pm 1.83 & 0.3 \pm 0.46 & 0.1 \pm 0.3 \\ 2.3 \pm 1.55 & 11.7 \pm 2.45 & 22.8 \pm 5.23 & 39.6 \pm 3.41 & 49.7 \pm 4.29 & 35.8 \pm 6.19 & 22.2 \pm 4.71 \\ 0.0 \pm 0.0 & 0.0 \pm 0.0 & 0.1 \pm 0.3 & 0.1 \pm 0.3 & 2.3 \pm 1.42 & 11.0 \pm 3.26 & 2.0 \pm 1.73 \\ 0.2 \pm 0.4 & 1.2 \pm 0.98 & 2.0 \pm 1.55 & 7.3 \pm 2.37 & 20.9 \pm 3.62 & 38.3 \pm 3.9 & 50.4 \pm 6.12 \end{bmatrix}$$

Figure 4.41: Confusion Matrix, Test, Q2.

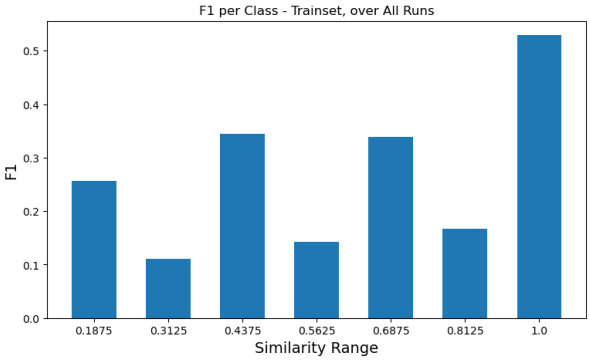

Figure 4.42: Mean F1 per class, Train, Q2.

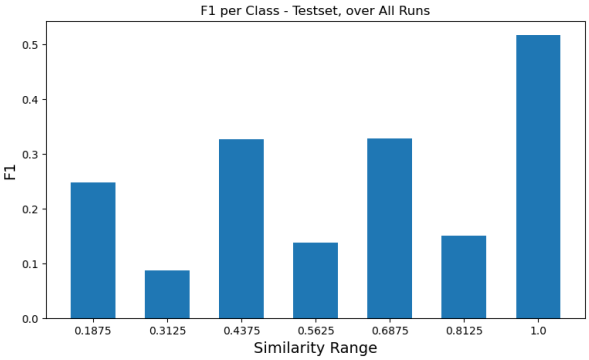

Figure 4.43: Mean F1 per class, Test, Q2.

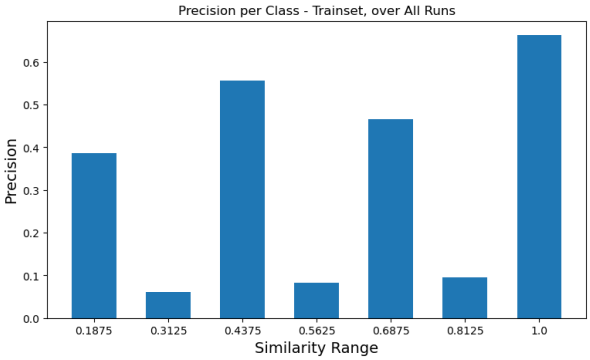

Figure 4.44: Mean precision per class, Train, Q2.

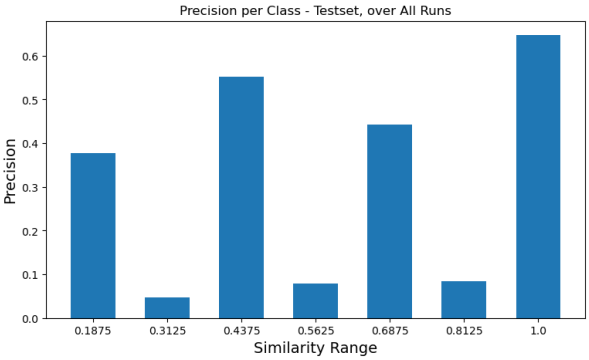

Figure 4.45: Mean precision per class, Test, Q2.

Figure 4.46: Mean recall per class, Train, Q2

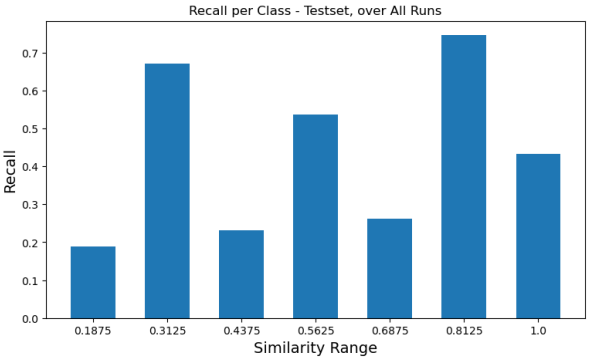

Figure 4.47: Mean recall per class, Test, Q2

Results Q3

Runs	MSE		MAE		Accuracy		F1		Precision		Recall	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Run 1	.0271	.0274	.1292	.1327	.343	.326	.391	.373	.343	.326	.552	.527
Run 2	.0260	.0276	.1266	.1300	.348	.345	.400	.397	.348	.345	.570	.557
Run 3	.0260	.0275	.1281	.1328	.333	.298	.379	.345	.333	.298	.544	.522
Run 4	.0262	.0232	.1280	.1211	.342	.358	.394	.408	.342	.358	.557	.580
Run 5	.0263	.0277	.1281	.1303	.339	.373	.386	.421	.339	.373	.544	.575
Run 6	.0261	.0266	.1270	.1292	.350	.319	.405	.361	.350	.319	.575	.526
Run 7	.0256	.0240	.1249	.1230	.368	.342	.424	.391	.368	.342	.591	.564
Run 8	.0252	.0281	.1247	.1319	.345	.342	.400	.391	.345	.342	.567	.525
Run 9	.0273	.0279	.1301	.1304	.347	.334	.392	.384	.347	.334	.542	.553
Run 10	.0267	.0258	.1288	.1262	.345	.348	.390	.403	.345	.348	.542	.569
Averages	.0263	.0266	.1275	.1288	.346	.339	.396	.387	.346	.339	.558	.550
St.devs	.0006	.0016	.0017	.0038	.0087	.0198	.012	.021	.009	.020	.016	.022

Table 4.17: Markov-Walk Dataset: Random-Split Performance by various metrics, Q3.

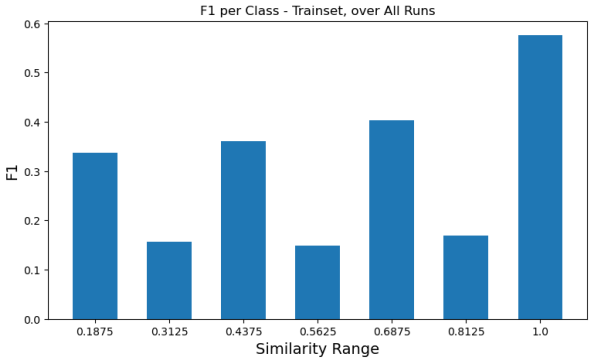

Figure 4.48: Mean F1 per class, Train, Q3.

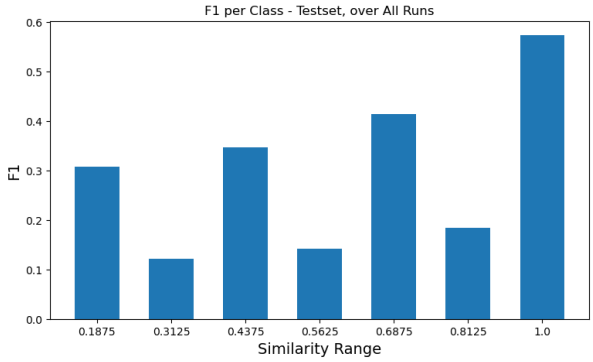

Figure 4.49: Mean F1 per class, Test, Q3.

$$\begin{bmatrix} 68.9 \pm 5.52 & 67.4 \pm 5.57 & 43.2 \pm 4.66 & 17.8 \pm 3.52 & 2.6 \pm 0.66 & 1.1 \pm 0.83 & 0.4 \pm 0.49 \\ 5.5 \pm 1.91 & 21.7 \pm 2.72 & 8.3 \pm 1.49 & 1.4 \pm 0.66 & 0.0 \pm 0.0 & 0.7 \pm 0.46 & 0.1 \pm 0.3 \\ 125.3 \pm 6.02 & 140.9 \pm 11.42 & 155.5 \pm 11.88 & 115.3 \pm 9.8 & 47.5 \pm 6.04 & 13.8 \pm 2.68 & 6.7 \pm 1.35 \\ 0.0 \pm 0.0 & 0.5 \pm 0.5 & 3.9 \pm 1.7 & 23.9 \pm 2.74 & 8.8 \pm 2.09 & 0.3 \pm 0.46 & 0.5 \pm 0.5 \\ 1.5 \pm 0.81 & 12.9 \pm 3.56 & 56.5 \pm 7.42 & 138.3 \pm 7.52 & 183.0 \pm 7.21 & 156.6 \pm 6.99 & 57.5 \pm 6.0 \\ 0.0 \pm 0.0 & 0.0 \pm 0.0 & 0.0 \pm 0.0 & 1.1 \pm 0.54 & 4.4 \pm 1.74 & 30.8 \pm 3.03 & 6.4 \pm 1.11 \\ 0.0 \pm 0.0 & 1.0 \pm 0.63 & 4.3 \pm 1.35 & 12.6 \pm 3.17 & 39.5 \pm 4.13 & 104.7 \pm 8.52 & 155.9 \pm 8.13 \end{bmatrix}$$

Figure 4.50: Confusion Matrix, Train, Q3.

$$\begin{bmatrix} 21.4 \pm 5.24 & 19.4 \pm 2.06 & 13.4 \pm 2.46 & 4.9 \pm 1.87 & 0.8 \pm 0.87 & 0.5 \pm 0.67 & 0.2 \pm 0.4 \\ 2.6 \pm 1.5 & 5.9 \pm 1.14 & 2.3 \pm 1.42 & 0.3 \pm 0.46 & 0.0 \pm 0.0 & 0.2 \pm 0.4 & 0.0 \pm 0.0 \\ 39.9 \pm 4.59 & 46.0 \pm 3.22 & 50.2 \pm 5.56 & 39.1 \pm 5.56 & 16.5 \pm 3.38 & 5.4 \pm 2.01 & 1.9 \pm 0.94 \\ 0.0 \pm 0.0 & 0.2 \pm 0.4 & 1.8 \pm 1.25 & 8.8 \pm 2.86 & 2.9 \pm 1.51 & 0.2 \pm 0.4 & 0.2 \pm 0.4 \\ 0.8 \pm 0.75 & 4.3 \pm 2.15 & 20.3 \pm 6.1 & 44.3 \pm 8.44 & 58.9 \pm 7.73 & 57.9 \pm 6.59 & 20.2 \pm 2.75 \\ 0.0 \pm 0.0 & 0.0 \pm 0.0 & 0.0 \pm 0.0 & 0.4 \pm 0.49 & 1.3 \pm 0.9 & 10.6 \pm 2.42 & 2.0 \pm 1.1 \\ 0.0 \pm 0.0 & 0.4 \pm 0.49 & 1.2 \pm 1.08 & 5.4 \pm 1.5 & 15.3 \pm 2.28 & 35.6 \pm 3.69 & 53.1 \pm 6.25 \end{bmatrix}$$

Figure 4.51: Confusion Matrix, Test, Q3.

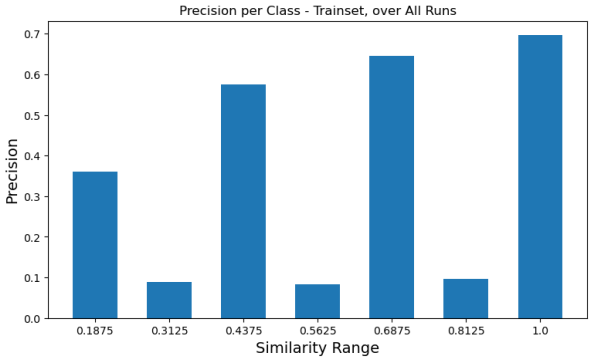

Figure 4.52: Mean precision per class, Train, Q3.

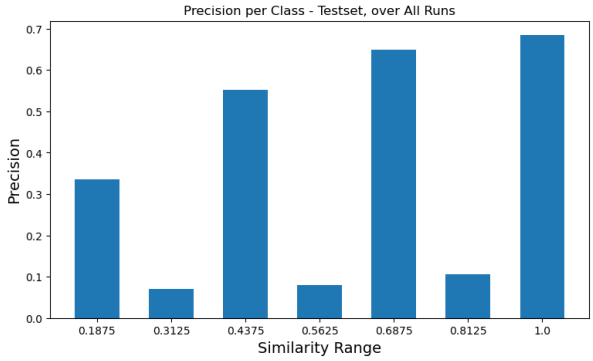

Figure 4.53: Mean precision per class, Test, Q3.

Discussion of Results

Figures 4.32 through 4.39 and Table 4.12 show the performance of the network for the random-split experiments for Q1, Figures 4.40 through 4.47 and Table 4.13 for Q2, and Figures 4.48 through 4.55 and Table 4.14 for Q3.

When comparing the obtained results to the results obtained for the FEM dataset, one quickly notices that the accuracy is considerably lower across all

Figure 4.54: Mean recall per class, Train, Q3

Figure 4.55: Mean recall per class, Test, Q3

three questions. Note that this was to be expected, since there are seven similarity categories instead of four, and thus the margins around the centers of the similarity intervals, which serve as ground truth similarities of the samples, are smaller.

Nonetheless, the huge difference between the accuracies obtained for the FEM and Markov-Walk datasets is not entirely due to the fact that there are more similarity categories. One can make two additional observations which may explain why the accuracies are much lower than for the FEM dataset. Firstly, the mean absolute error and mean squared error are higher. The difference is substantial with the mean absolute error being 50% to 75% higher for all questions and for both training and test sets for the FEM dataset than for the Markov-Walk dataset.

Secondly, one can verify that the overall agreements are lower for all three questions in the case that all raters are involved (See Table 3.4). However, the latter difference is quite small, as measured by Krippendorff's Alpha, the overall agreement for the FEM dataset amounts to 0.674, 0.562, and 0.733, for questions one through three, respectively, whereas the overall agreement for the Markow-Walk dataset amounts to 0.634, 0.519, and 0.618. While the overall agreement is lower for the Markov-Walk dataset across all three questions, the overall agreement on the second question for the FEM dataset is much lower than the overall agreement on the first question for the Markov-Walk dataset.

Yet, averaged across 10 runs, the mean absolute error on the annotations for the second question for the FEM dataset is around 0.095, while the mean absolute error on the annotations for the first question for the Markov-Walk dataset is around 0.125. Hence, the rate overall agreement, alone, is not an accurate predictor of the performance of the neural network for a given dataset. One must be mindful of the fact that an existence of a correlation between the rate of agreement and the performance of the network does not imply that a higher

rate of agreement always results in better performance. There are other factors and variables which we do not fully understand that influence the outcome, which motivates future work to gain a better understanding of these factors.

However, I suspect that the higher number of raters involved in labelling the Markov-Walk samples might explain the difference. It is evident that the agreement between pairs of raters is typically quite poor. Hence, the involvement of numerous raters leads to numerous different views of how similar certain pairs of cracked facades are, which in turn leads to more inconsistent data which ultimately results in poorer performance. To determine whether this is truly the case, additional experimentation is needed.

Furthermore, the results for the Markov-Walk dataset further support that the overall agreement and the performance of the network are correlated, the performance is best for Q1, for which the agreement is highest, followed closely by Q3, for which the agreement is only slightly lower, and it is worst for Q2, for which the agreement is clearly the lowest.

Regarding the average per class precision, recall, and F1 scores, one finds that these are far worse for the intermediate intervals than for the original intervals. One can also tell this from analyzing the confusion matrices which clearly show that many samples were wrongly classified by a single class. This is likely due to there being far less samples belonging to these intermediate classes.

In the next section, an analysis of the relation between the network's performance on a per sample level is given. After that, the leave one out experiments, also referred to as the "generalizability" experiments, will be discussed, starting with the leave-one-crack out experiments, followed by the leave-one-class out experiments.

4.6.5 Markov-Walk: Characteristics of Misclassified Samples

This section provides an analysis of the samples for which the network performed poorly. that is, the samples for which the NN's predictions were farthest off from the true similarites, in general. Separate analyses are shown for each of the three questions.

Tables 4.18 through 4.20 show how often each pair of crack archetypes that occur among the n samples ($n = 250$) for which the network's predictions were poorest, how often these pairs of crack archetypes occurred throughout those n samples (column "Occurrences") as well as how often these pairs occur throughout the entire Markov-Walk dataset, separately for the three similarity questions. One could compute how often a pair of crack archetypes occurs throughout the n samples relatively to how often it occurs throughout the whole dataset (column

"Total Occurrences") to gain a better understanding of how much trouble a pair of crack archetypes poses to the network in terms of its predictability.

When interpreting the results shown in Tables 4.18 through 4.20, a few considerations must be kept in mind. Firstly, some pairs of crack archetypes may occur less frequently than others, and yet be more difficult to accurately predict as indicated by a higher relative occurrence frequency (touched on in previous paragraph). On the other hand, if a pair of crack archetypes is scarecely represented throughout the entire dataset, a high relative occurrence frequency does not provide strong evidence that a pair of crack archetypes poses problems to the network. In such cases, it may be wise to generate and annotate more samples that contain said crack archetypes in order to determine whether it is a problematic pair.

Furthermore, one observes that there is a correlation between which pairs of crack archetypes pose difficulty to the network in terms of predictability, and which pairs of crack archetypes have the lowest rate of agreement. One can verify this by comparing Tables 4.18 through 4.20 to Table 3.6. One can verify that pairs (101, 102), (103, 103), (103, 201), (201, 201), (102, 102) are show up in the top portion of both Tables 3.6, 4.18, and 4.20 which provides evidence that the rate of agreement for a given pair of crack archetypes is correlated with how well the network is able to perform on said pair of crack archetypes.

Regarding the relation between the rate of agreement for a given pair of crack archetypes and its predictability from a neural network perspective, it is important to note that the two concepts are somewhat correlated, but not strongly correlated. This is reflected by the fact that the relative occurrence frequency of some pairs of crack archetypes greatly varies between Table 3.6 and Tables 4.18 through 4.20. An example of this is pair (101, 102). While it occurs 54 times among the 250 samples with the lowest agreement, it occurs only 19 times throughout the 250 samples for which the network's predictions are poorest.

A possible explanation for the difference between the occurrence frequencies for pair (101, 102) in Table 3.6 compared to Tables 4.18 to 4.20, only 12 times for question 2 (Table 4.19), is that there is likely much variation in how disagreement among a set of samples featuring the same crack archetypes affects the true similarities for those samples. For example, suppose that 10 samples consisting of the same pair of crack archetypes, all rated by two raters, are given similarity assessments Very Dissimilar and Very Similar. While this implies that the agreement is shockingly low, averaging the ratings will yield a stable 0.5, which will make it easier for the network to predict the similarity score for these samples. In other cases, low agreement among a small subset of all samples consisting of a given pair of crack archetypes, might cause the network to poorly predict many of those samples. Lastly, it is important to keep in mind that Tables 4.18 through 4.20 reflect results per question, whereas the agreement per sample takes all questions into consideration at once.

Archetypes	Occurrences	Total Occurrences
(103, 201)	34	156
(103, 103)	25	100
(101, 102)	19	154
(201, 201)	16	89
(31, 32)	14	181
(23, 30)	14	165
(20, 21)	11	155
(24, 32)	11	181
(24, 24)	11	107
(102, 102)	10	71
(18, 18)	9	78
(20, 20)	5	88
(23, 23)	5	103
(18, 20)	5	131
(30, 30)	5	81
(24, 31)	4	6
(32, 32)	4	96
(101, 101)	4	114
(21, 21)	4	103
(23, 31)	3	4
(31, 31)	3	106
(21, 23)	3	8
(18, 23)	2	4
(21, 30)	2	3
(18, 24)	2	6
(23, 24)	2	4
(21, 24)	2	4
(24, 103)	2	3
(20, 23)	2	6
(20, 32)	2	3
(18, 30)	2	2
(30, 31)	1	1
(31, 101)	1	6
(21, 102)	1	5
(30, 102)	1	9
(30, 101)	1	2
(30, 32)	1	3
(20, 30)	1	2
(102, 201)	1	1
(101, 103)	1	5
(21, 32)	1	2
(23, 201)	1	2
(18, 32)	1	2
(24, 201)	1	3

Table 4.18: Markov-Walk Dataset, Q1: All pairs of crack archetypes that occur (and how often) among the 250 samples most poorly predicted by the network, and how often these crack pairs occur in the entire dataset (2466 samples)

Archetypes	Occurrences	Total Occurrences
(23, 30)	26	165
(103, 201)	24	156
(24, 32)	23	181
(31, 32)	21	181
(24, 24)	16	107
(103, 103)	15	100
(20, 21)	14	155
(18, 20)	13	131
(101, 102)	12	154
(23, 23)	11	103
(31, 31)	11	106
(18, 18)	9	78
(20, 20)	8	88
(30, 30)	6	81
(21, 21)	6	103
(102, 102)	4	71
(201, 201)	4	89
(32, 32)	4	96
(18, 24)	3	6
(23, 31)	2	4
(18, 23)	2	4
(23, 24)	2	4
(101, 101)	2	114
(24, 101)	1	5
(21, 24)	1	4
(18, 32)	1	2
(31, 101)	1	6
(101, 201)	1	4
(102, 201)	1	1
(24, 201)	1	3
(21, 30)	1	3
(102, 103)	1	4
(20, 30)	1	2
(18, 31)	1	4
(24, 31)	1	6

Table 4.19: Markov-Walk Dataset, Q2: All pairs of crack archetypes that occur (and how often) among the 250 samples most poorly predicted by the network, and how often these crack pairs occur in the entire dataset (2466 samples)

Archetypes	Occurrences	Total Occurrences
(103, 201)	27	156
(103, 103)	23	100
(31, 32)	19	181
(23, 30)	19	165
(101, 102)	18	154
(24, 32)	16	181
(102, 102)	12	71
(24, 24)	11	107
(23, 23)	9	103
(20, 20)	8	88
(20, 21)	8	155
(18, 18)	8	78
(18, 20)	8	131
(201, 201)	7	89
(32, 32)	6	96
(31, 31)	5	106
(30, 30)	5	81
(101, 101)	4	114
(31, 101)	4	6
(21, 21)	4	103
(23, 31)	3	4
(21, 30)	3	3
(18, 24)	2	6
(21, 23)	2	8
(102, 103)	2	4
(24, 101)	2	5
(23, 24)	1	4
(18, 23)	1	4
(21, 24)	1	4
(24, 103)	1	3
(30, 103)	1	3
(20, 32)	1	3
(102, 201)	1	1
(23, 102)	1	5
(24, 31)	1	6
(30, 32)	1	3
(24, 201)	1	3
(20, 30)	1	2
(23, 201)	1	2
(20, 101)	1	5
(21, 101)	1	7

Table 4.20: Markov-Walk Dataset, Q3: All pairs of crack archetypes that occur (and how often) among the 250 samples most poorly predicted by the network, and how often these crack pairs occur in the entire dataset (2466 samples)

4.6.6 Leave One Crack Out

This section presents the results of the leave one crack out experiments on the Markov-Walk dataset.

Results Q1, Q2, and Q3

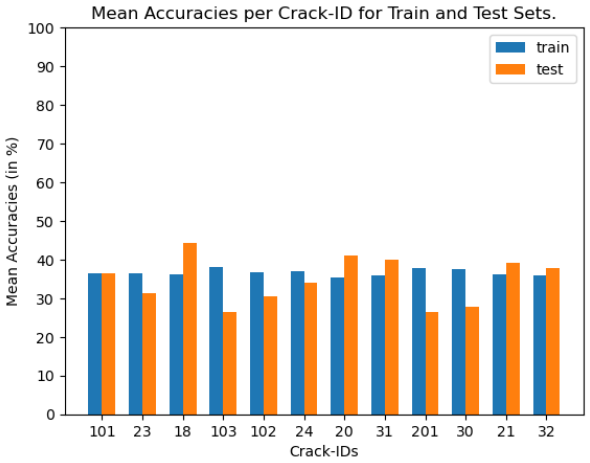

Figure 4.56: Mean Train/Test Accuracies per Crack

Crack-ID	MSE		MAE	
	Train	Test	Train	Test
101	.0246	.0224	.1217	.1158
23	.0239	.0315	.1200	.1393
18	.0242	.0246	.1217	.1111
103	.0219	.0331	.1146	.1481
102	.0249	.0276	.1221	.1316
24	.0247	.0262	.1216	.1266
20	.0257	.0188	.1253	.1033
31	.0247	.0217	.1219	.1134
201	.0229	.0324	.1172	.1451
30	.0246	.0282	.1210	.1380
21	.0233	.0228	.1193	.1138
32	.0247	.0180	.1218	.1067

Table 4.21: MAE/MSE - Leave out Crack, Q1, FEM

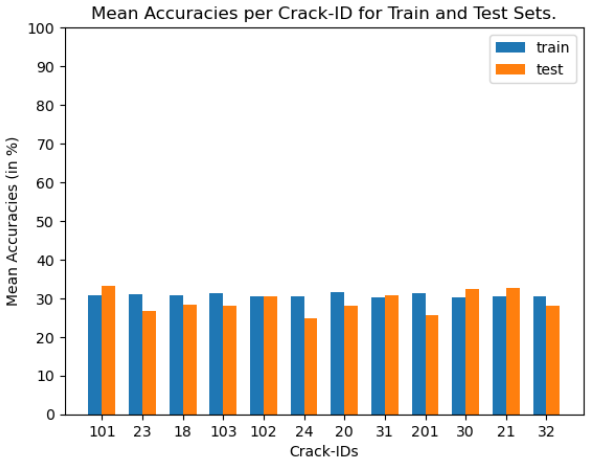

Figure 4.57: Mean Train/Test Accuracies per Crack

Crack-ID	MSE		MAE	
	Train	Test	Train	Test
101	.0348	.0264	.1460	.1267
23	.0318	.0413	.1392	.1615
18	.0332	.0350	.1425	.1475
103	.0319	.0369	.1396	.1535
102	.0331	.0271	.1431	.1307
24	.0323	.0453	.1411	.1671
20	.0324	.0350	.1411	.1449
31	.0341	.0345	.1446	.1451
201	.0336	.0318	.1430	.1438
30	.0335	.0331	.1427	.1424
21	.0319	.0285	.1407	.1305
32	.0347	.0328	.1456	.1434

Table 4.22: MAE/MSE - Leave out Crack, Q2, FEM

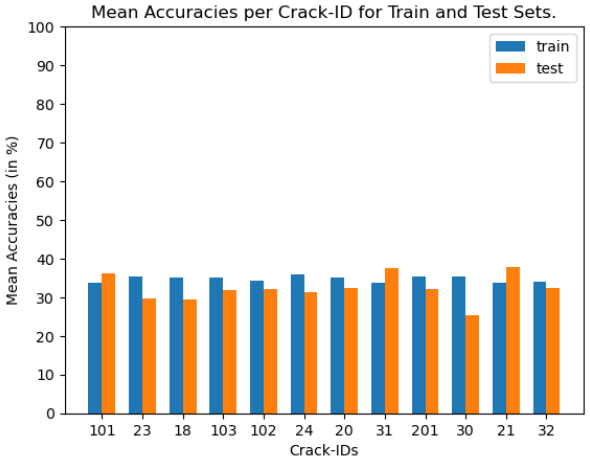

Figure 4.58: Mean Train/Test Accuracies per Crack

Crack-ID	MSE		MAE	
	Train	Test	Train	Test
101	.0273	.0237	.1303	.1210
23	.0251	.0320	.1245	.1459
18	.0245	.0267	.1234	.1294
103	.0244	.0354	.1235	.1462
102	.0267	.0260	.1287	.1288
24	.0252	.0283	.1247	.1321
20	.0248	.0216	.1244	.1143
31	.0269	.0251	.1290	.1252
201	.0248	.0285	.1240	.1313
30	.0268	.0342	.1281	.1550
21	.0268	.0202	.1296	.1094
32	.0264	.0230	.1277	.1228

Table 4.23: MAE/MSE - Leave out Crack, Q3, FEM

4.6.7 Experimental Set-Up

The set-up of the leave one crack out experiments is the same as for the FEM-dataset. However, this dataset contains 12 crack archetypes, namely: 18, 20, 21, 23, 24, 30, 31, 32, 101, 102, 103 and 201.

Discussion of Results

Figures 4.56 through 4.58 and tables 4.15 through 4.17 are shown side-by-side, and reflect the performance of the network for the leave one class out experiments for the Markov-Walk dataset on all three questions, separately.

When one compares these results to those obtained for the same experiment on the FEM dataset, one quickly notices that the network performs much worse. This is not surprising since the network also performs far worse on the Markov-Walk dataset for the Random-Split than it does on the FEM dataset.

Furthermore, one observes the same surprising phenomenon that occurred in the results for the FEM dataset, that is test accuracies exceeding train accuracies. This holds true for crack archetypes 21, 32 and 101 across all three questions.

For crack archetypes 18 and 31 it is true for some of the questions. For crack archetype 18, the test accuracy far exceeds the train accuracy for the first question, while for the second and third questions, the reverse holds true. This strongly suggests that this phenomenon is not archetype specific, but that it also depends on the question.

4.6.8 Leave One Class Out

This section presents the results of the leave one class out experiments on the Markov-Walk dataset.

Results Q1, Q2 and Q3

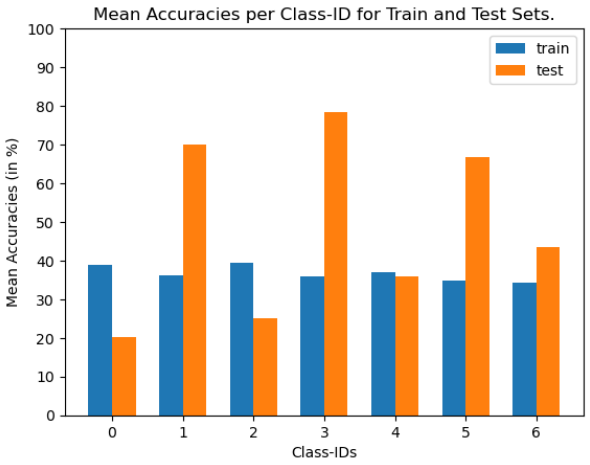

Figure 4.59: Mean Train/Test Accuracies per Crack

Class-ID	MSE		MAE	
	Train	Test	Train	Test
0	.0200	.0423	.1108	.1655
1	.0241	.0092	.1206	.0595
2	.0255	.0259	.1210	.1345
3	.0247	.0042	.1222	.0432
4	.0267	.0186	.1263	.1087
5	.0242	.0062	.1214	.0554
6	.0239	.0271	.1198	.1289

Table 4.24: MAE/MSE - Leave out Class, Q1, Markov-Walk

Figure 4.60: Mean Train/Test Accuracies per Crack

Class-ID	MSE		MAE	
	Train	Test	Train	Test
0	.0318	.0517	.1397	.1803
1	.0329	.0090	.1419	.0649
2	.0336	.0331	.1416	.1477
3	.0356	.0120	.1478	.0829
4	.0328	.0342	.1409	.1478
5	.0346	.0071	.1467	.0550
6	.0331	.0352	.1431	.1437

Table 4.25: MAE/MSE - Leave out Class, Q2, Markov-Walk

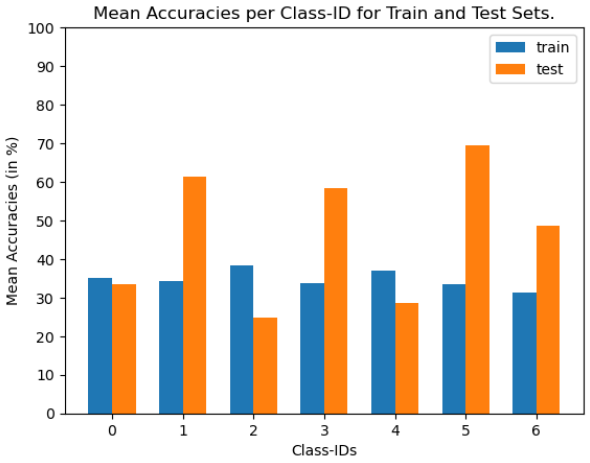

Figure 4.61: Mean Train/Test Accuracies per Crack

Class-ID	MSE		MAE	
	Train	Test	Train	Test
0	.0237	.0352	.1221	.1440
1	.0256	.0123	.1267	.0721
2	.0252	.0324	.1224	.1480
3	.0270	.0082	.1300	.0667
4	.0281	.0232	.1302	.1241
5	.0278	.0055	.1316	.0529
6	.0255	.0252	.1266	.1208

Table 4.26: MAE/MSE - Leave out Class, Q3, Markov-Walk

4.6.9 Experimental Set-Up

The set-up of the leave one class out experiments is the same as for the FEM-dataset. However, as mentioned before this dataset contains 7 similarity classes instead of 4, namely: {**0-0.1875, 0.1875-0.3125, 0.3125-0.4375, 0.4375-0.5625, 0.5625-0.6875, 0.6875-0.8125, 0.8125-1.0**}. Thus, there are 7 sub-experiments.

Discussion of Results

Figures 4.59 through 4.61 and tables 4.18 through 4.20 are shown side-by-side, and reflect the performance of the network for the leave one class out experiments for the Markov-Walk dataset on all three questions, separately.

It is useful to compare the results of these experiments to the results that were acquired for the same experiments on the FEM dataset. One notices that the model is not able to accurately assess the degree of similarity for samples that belong to the lowest similarity class when it is fitted to samples belonging to the other classes.

Furthermore, it is quite remarkable that the model performs best when leaving out one of the intermediate similarity classes: 1, 2, or 3. In those cases, the test accuracies far exceed the train accuracies. Note that these intermediate classes contain few samples, therefore the network has more samples to train with if one of those classes is left out. However, the fact that the network has more samples to train on, alone, does not explain the huge discrepancy between the train and test accuracies.

One can observe that the MAE and MSE roughly provide similar views of the networks performance, i.e. if the test accuracy is higher than the train accuracy, then the MAE on the testset is generally lower than the MAE on the trainset, and vice versa.

All in all, these results further highlight the importance of further research in order to better understand under which circumstances the network is able to generalize well and under which circumstances it is not.

Chapter 5

Future Work and Conclusions

This section provides various suggestions for future work. First, I discuss how one could proceed with the Inter-rater Reliability in order to gain a better understanding of the differences between the ratings provided by different masonry experts, and what can be done about these. One must be mindfull of the fact that the suggested experiments involve people, and their willingness to cooperate and invest their precious time to help advance this project is greatly appreciated. Secondly, I provide suggestions on how to improve the performance of the Neural Network as well as how to more effectively determine the network's ability to generalize to inputs from different parts of the distribution.

5.0.1 Inter-rater Reliability

In the following, several ideas are presented on how to extend the Inter-rater reliability analysis. One must be mindful of the fact that the suggested experiments involve people, and while their willingness to cooperate and invest their precious time to help advance this project is greatly appreciated, for the sake of the goal of partially automating the assessment of cracked facades, it is essential to provide the Neural Network with consistent ratings that reflect an objective ground truth, which highly motivates the suggested experiments:

- One can imagine that there are external factors that affect the labelling behaviour of raters. An obvious factor that could play a role when raters are asked to label many samples in a single labelling session is fatigue. As the raters proceed, they can become tired which results in a decrease in focus. In order to determine at what point raters get tired, i.e. how many samples raters can typically rate before the quality of their judgement takes a dip, the following experimental design could be useful.

Invite n masonry experts to label N samples (with N large, $N \geq 200$), and have each of the n raters label all of the N samples in the same order. Compute the Inter-rater agreement over the whole set of samples, as well as over samples 1 through 50, samples 51 through 100, samples 101 through 150, etcetera. If the Inter-rater agreement gradually decreases after c samples have been rated ($c < n$), this suggests that c can be viewed as a sort of cut-off point, and raters should not be asked to label more than c samples in a single session. Furthermore, one could sprinkle some duplicates and samples with identical images through the dataset. Of course, samples that feature the same cracked facade image twice should be rated as Very Similar across all questions, while the duplicated samples can be used to determine the Intra-rater Reliability of the different raters, and how it fluctuates as a function of how many samples have already been labelled by the rater in question.

- In order to to filter out unreliable raters, one can sprinkle some duplicate samples through the dataset, as well as pairs of samples that are one another's reverse, that is both samples show the same cracked facades but in the opposite order. Note that reversed samples as well as duplicates should be rated identically, because the degree of similarity between image A and image B is the same as the degree of similarity between image B and image A (images reversed). Duplicate samples should receive the same ratings for obvious reasons. By adding such pairs of samples throughout the dataset, one can measure the Intra-rater Reliability of a given rater over different samples that should have received the same ratings. Note that the Inter-rater Reliability can be measured within a single labelling session as well as across labelling sessions.

- Finally, one could consider a different labelling set-up in which raters are asked to motivate their judgements. To this end, one could add a small text box that allows the masonry experts to motivate their decisions. Since this will be much more time-consuming for the raters, it would be wise to include fewer samples per labelling session.

5.0.2 Neural Networks and Generalizability

Several suggestions to improve the network's performance as well as to better evaluate its ability to generalize to different parts of the input distribution are summarized below:

- Across all experiments reported in chapter 4 of this writing, one can observe that the epoch in which the best model is produced, i.e. the model that minimizes the mean squared error on the training set, varies considerably between the different experiments. In order to explain the neural network's search for a configuration of the weights that minimizes the loss on the training set, the landscape metaphor is oftentimes used. In this metaphor, the weights of the network can be viewed as points in the (x, y)

plane, whereas z denotes that objective function, the loss which is often represented by the mean squared error. In order to navigate the space to lower the value of z, gradient descent is used which is based on the partial derivatives with respect to the individual weights. These determine in which direction the search space is navigated in each epoch. The learning rate, l, determines the step size, i.e. the size of the step that is taken in the given direction. If the step size is too large, the network may overshoot the target, and the loss might increase. To avoid this, one can decrease the learning rate throughout the training phase of the network. To this end, several policies can be implemented. One can think of linearly and exponentially decreasing learning rates, to name but a few.

- As mentioned previously, the different pairs of crack archetypes are not equally represented in any of the three datasets. While it may be justified from the point of view of a structural engineer to include more samples that feature a specific pair of crack archetypes, and fewer samples that feature another pair of crack archetypes, in order to improve the network's performance, it may be wise to balance the degree of representation of the different pairs of crack archetypes, or at least ensure that each pair of crack archetypes is sufficiently represented in each dataset. One could, for instance, include a minimum number of samples for each pair of crack archetypes.

- In order to better understand why the network is able to generalize well to certain crack archetypes but not to others, it would be instructive to determine whether a correlation exists between how well the model generalizes to samples that contain a given crack archetype and to what extent the masonry experts tend to (dis)agree on samples that contain the given crack archetype.

- In order to better evaluate the network's performance on the leave one out experiments, in the case that the test accuracy is far lower than the train accuracy, it may be instructive to check whether samples that contain the given crack or class to be left out of the training set are also more often misclassified in the random split experiments. If this is indeed the case, this suggests that the poorer performance of the network on samples that contain the specific crack archetype can not be explained by a difference in how the similarity is to be determined between these problematic samples and other samples, but that it is rather due to a sample somehow posing problems to the network in terms of its predictability.

5.0.3 Conclusions

In this concluding section, all of the research questions are addressed individually. First, all research question will be listed along with the conclusions for each:

RQ 1: *The raters clearly agree much more than one would expect ratings to agree based purely on chance. However, for the purpose of automating the assessment of crack similarity, a higher rate of agreement is probably needed.*

RQ 2: *Seperately, for the FEM and Markov-Walk dataset, I have identified characteristics of samples that experts tend to disagree on. For the Markov-Walk dataset, pairs (101, 102) and (103, 103) were identified as samples for which the agreement is low, and for the FEM dataset, pairs (23, 24), (24, 103) and (18, 102) appear to be most problematic in terms of agreement.*

RQ 3: *Whether a neural network is able to learn to assess the similarity of pairs of cracked facades as assessed by experts, is dependent on the rate of agreement among the raters. If the agreement is high or the data was judged by a single rater, the neural network can do this quite well, but if the agreement is low and many raters are involved, the network typically performs poorly.*

RQ 4: *My results show that the network can generalize reasonably well across different degrees of similarity as well as different crack archetypes. That is it is somewhat able to assess the similarity of a pair of cracked facades, even in the case that one of the cracked facades is not present in the training set. This suggests that there may exist a notion of similarity in the context of pairs of cracked masonry facades that underlies all pairs of cracked facades, regardless of the patterns of the involved cracks. However, the discrepancies between the performances yielded for different leave out cracks suggests that more extensive research is needed to understand the limitations of the network when it is expected to generalize.*

Bibliography

[1] A. Rozsas, "Reliability analysis of RC structures: accomplishments and aspirations," https://www.youtube.com/watch?v=Ail012-Dh3M, 2018, [Online; accessed 19-November-2021].

[2] I. A. E. De Vent, "Structural damage in masonry: Developing diagnostic decision support," 2011.

[3] Á. Rózsás, A. Slobbe, W. Huizinga, M. Kruithof, and G. Giardina, "A neural network embedding for quantifying crack pattern similarity in masonry structures - a statistical, proof of concept study (accepted for publication). In 12th international conference on structural analysis of historical constructions, Barcelona, Spain." 2020, [Online; accessed 17-July-2021].

[4] K. Ajithkumar Pillai, "Integrating machine learning and computational physics to assess crack pattern similarity in masonry buildings," 2022.

[5] Á. Rózsás, A. Slobbe, W. Huizinga, M. Kruithof, G. Giardina, C. Intrigila, N. Nodargi, P. Bisegna, E. Coïsson, D. Ferretti *et al.*, "Development of a neural network embedding for quantifying crack pattern similarity in masonry structures," in *12th International Conference on Structural Analysis of Historical Constructions*. International Centre for Numerical Methods in Engineering, CIMNE, 2021, pp. 1905–1916.

[6] J. Twomey and A. Smith, "Performance measures, consistency, and power for artificial neural network models," *Mathematical and computer modelling*, vol. 21, no. 1-2, pp. 243–258, 1995.

[7] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3519–3529.

[8] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring generalization in deep learning," *Advances in neural information processing systems*, vol. 30, 2017.

[9] G. Koch, R. Zemel, R. Salakhutdinov *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2. Lille, 2015, p. 0.

[10] R. J. Hunt, "Percent agreement, pearson's correlation, and kappa as measures of inter-examiner reliability," *Journal of Dental Research*, vol. 65, no. 2, pp. 128–130, 1986.

[11] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[12] I. Lawrence and K. Lin, "A concordance correlation coefficient to evaluate reproducibility," *Biometrics*, pp. 255–268, 1989.

[13] J. L. Fleiss, "Measuring nominal scale agreement among many raters." *Psychological bulletin*, vol. 76, no. 5, p. 378, 1971.

[14] R. A. Fisher, "Statistical methods for research workers," in *Breakthroughs in statistics*. Springer, 1992, pp. 66–70.

[15] J. J. Bartko, "The intraclass correlation coefficient as a measure of reliability," *Psychological reports*, vol. 19, no. 1, pp. 3–11, 1966.

[16] H. X. Barnhart, M. Haber, and J. Song, "Overall concordance correlation coefficient for evaluating agreement among multiple observers," *Biometrics*, vol. 58, no. 4, pp. 1020–1027, 2002.

[17] K. Krippendorff, "Estimating the reliability, systematic error and random error of interval data," *Educational and Psychological Measurement*, vol. 30, no. 1, pp. 61–70, 1970.

[18] Y. Chan, "Biostatistics 104: correlational analysis," *Singapore Med J*, vol. 44, no. 12, pp. 614–619, 2003.

[19] C. P. Dancey and J. Reidy, *Statistics without maths for psychology*. Pearson education, 2007.

[20] D. V. Cicchetti, "Guidelines, criteria, and rules of thumb for evaluating normed and standardized assessment instruments in psychology." *Psychological assessment*, vol. 6, no. 4, p. 284, 1994.

[21] J. L. Fleiss, B. Levin, M. C. Paik *et al.*, "The measurement of interrater agreement," *Statistical methods for rates and proportions*, vol. 2, no. 212-236, pp. 22–23, 1981.

[22] M. L. McHugh, "Interrater reliability: the kappa statistic," *Biochemia medica*, vol. 22, no. 3, pp. 276–282, 2012.

[23] T. K. Koo and M. Y. Li, "A guideline of selecting and reporting intraclass correlation coefficients for reliability research," *Journal of chiropractic medicine*, vol. 15, no. 2, pp. 155–163, 2016.

[24] G. McBride, "A proposal for strength-of-agreement criteria for lin's concordance correlation coefficient. niwa client report: Ham2005-062; national institute of water & atmospheric research: Hamilton, new zeeland, may 2005," 2019.

[25] P. Watson and A. Petrie, "Method agreement analysis: a review of correct methodology," *Theriogenology*, vol. 73, no. 9, pp. 1167–1179, 2010.

[26] D. t. Hove, T. D. Jorgensen, and L. Ark, "On the usefulness of interrater reliability coefficients," in *The Annual Meeting of the Psychometric Society*. Springer, 2017, pp. 67–75.

[27] K. Pearson, "Notes on regression and inheritance in the case of two parents proceedings of the royal society of london, 58, 240-242," 1895.

[28] Anonymous, "Krippendorffs Alpha," https://en.wikipedia.org/wiki/Klaus_Krippendorff, 2021, [Online; accessed 10-December-2021].

[29] G. Giardina, J. Rots, and M. Hendriks, "Modelling of settlement induced building damage," *TU Delft, Delft*, 2013.

[30] J. Serhal, O. Deck, M. Al Heib, F. H. Chehade, and D. Y. A. Massih, "Damage of masonry structures relative to their properties: Development of ground movement fragility curves," *Engineering Structures*, vol. 113, pp. 206–219, 2016.

Chapter 6

Appendix

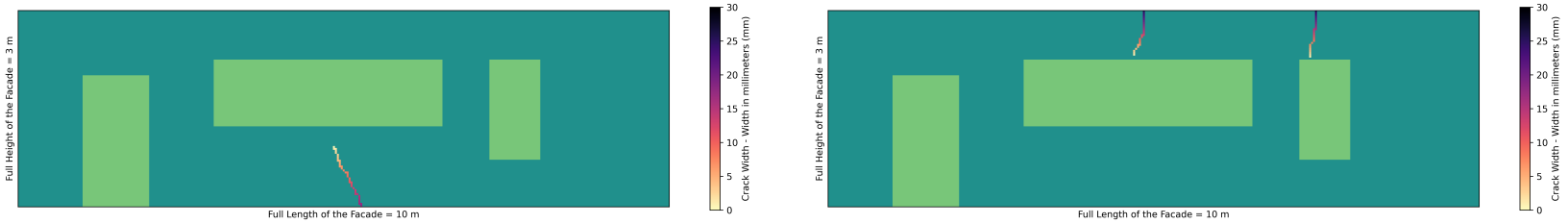

Figure 6.1: Markov-Walk: Pair of cracked facades

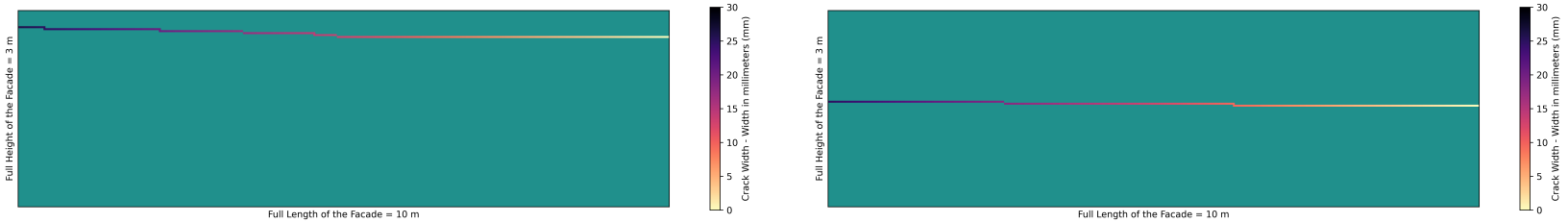

Figure 6.2: Markov-Walk: Pair of cracked facades

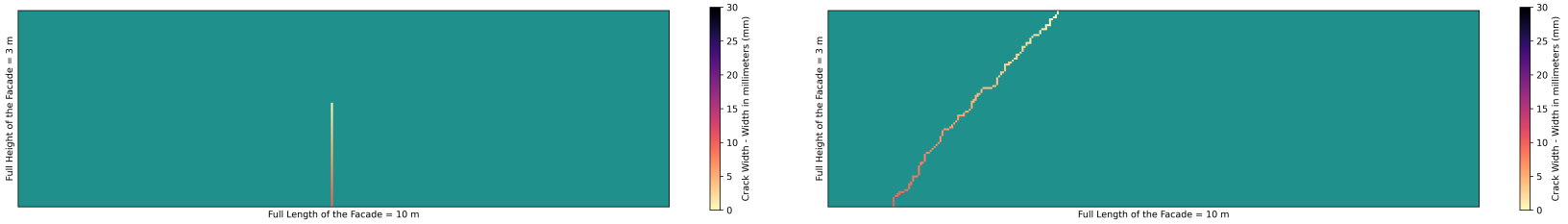

Figure 6.3: Markov-Walk: Pair of cracked facades

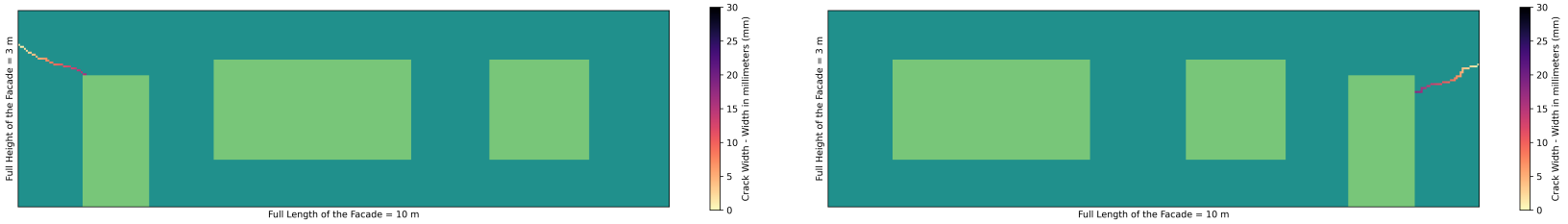

Figure 6.4: Problematic samples - Archetypes 101 (Left) and 102 (Right)

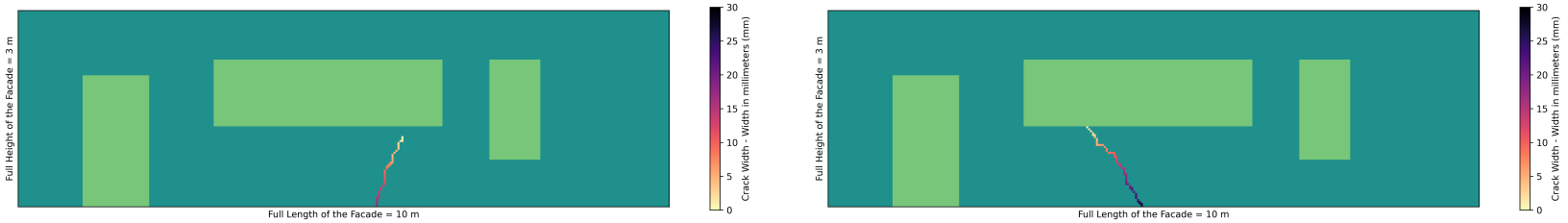

Figure 6.5: Problematic samples - Archetypes 103 (Left) and 103 (Right)

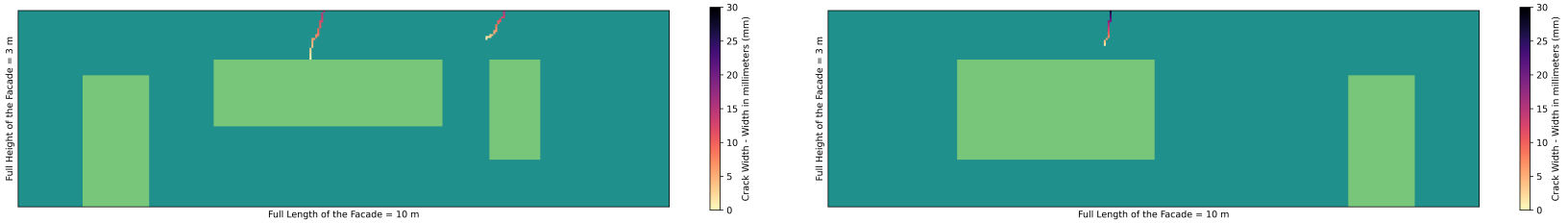

Figure 6.6: Problematic samples - Archetypes 201 (Left) and 201 (Right)

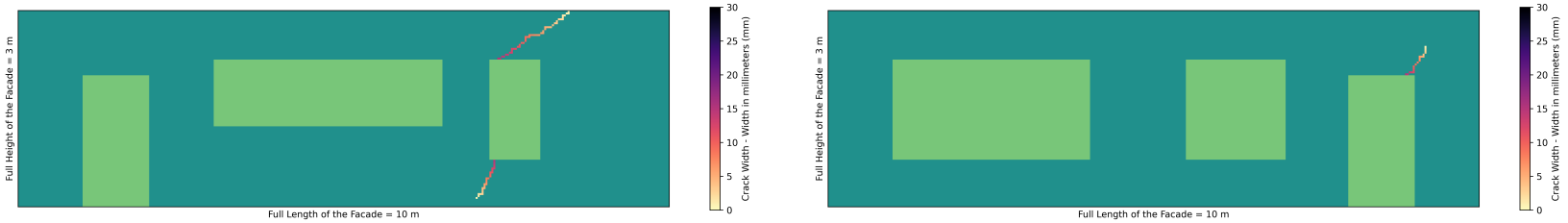

Figure 6.7: Problematic samples - Archetypes 102 (Left) and 102 (Right)

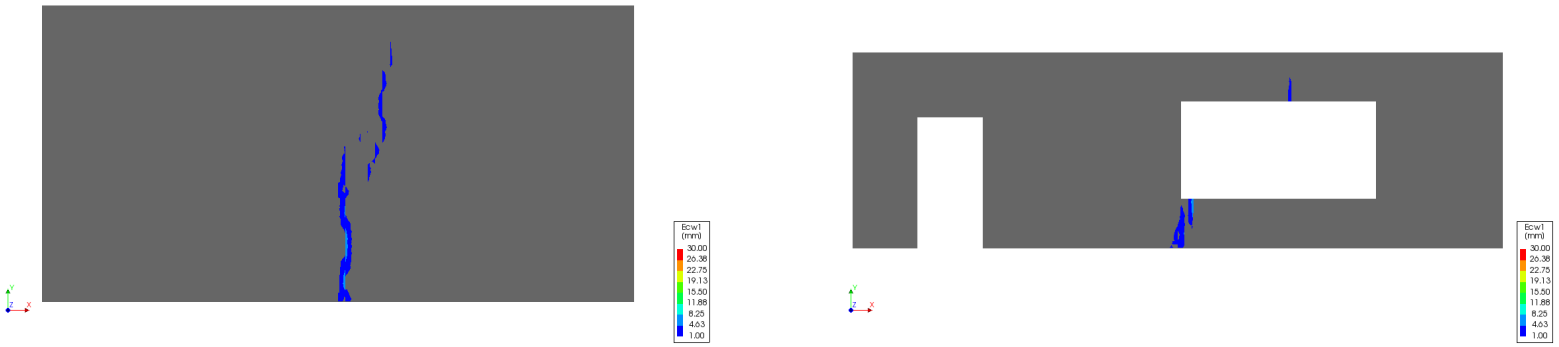

Figure 6.8: FEM: Problematic samples - Archetypes 24 (Left) and 103 (Right)

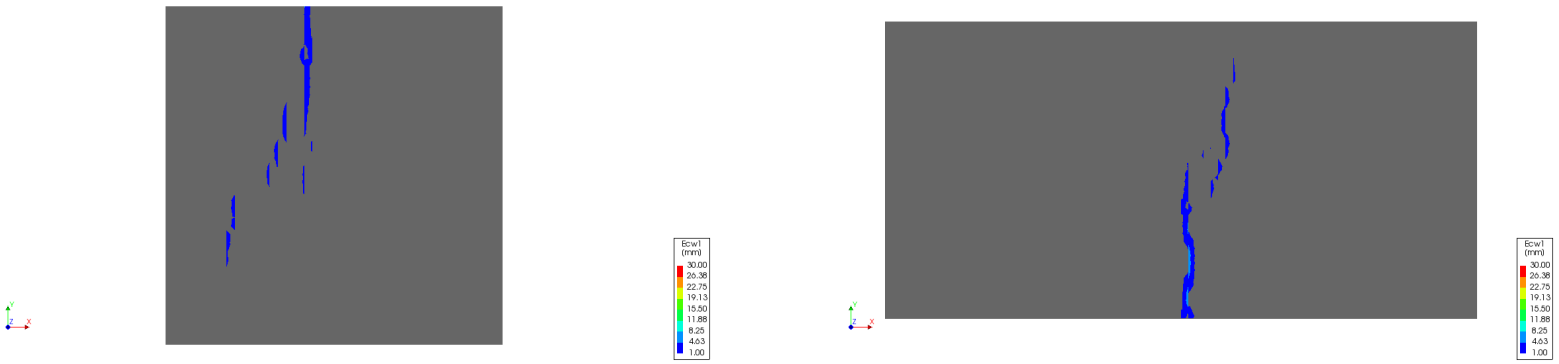

Figure 6.9: FEM: Problematic samples - Archetypes 23 (Left) and 24 (Right)

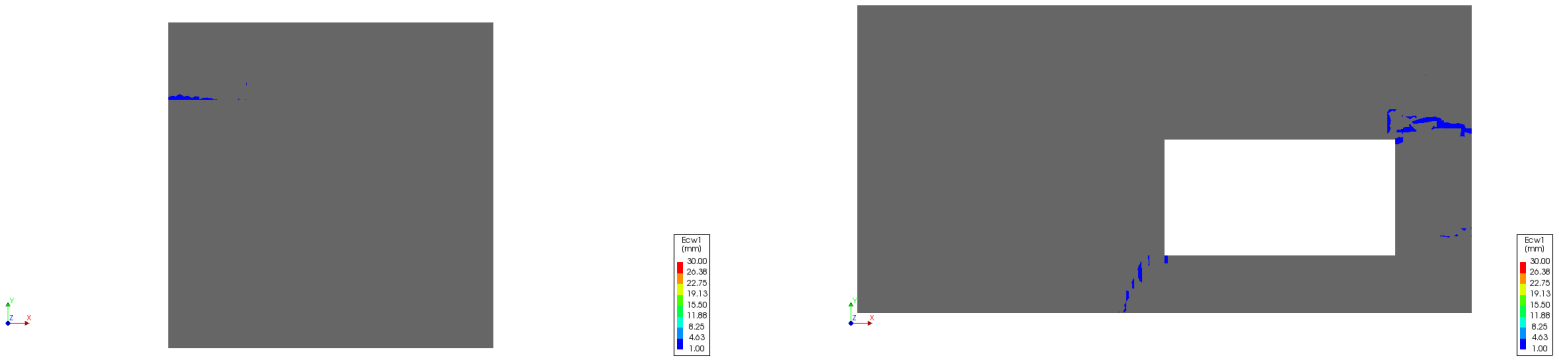

Figure 6.10: FEM: Problematic samples - Archetypes 18 (Left) and 102 (Right)

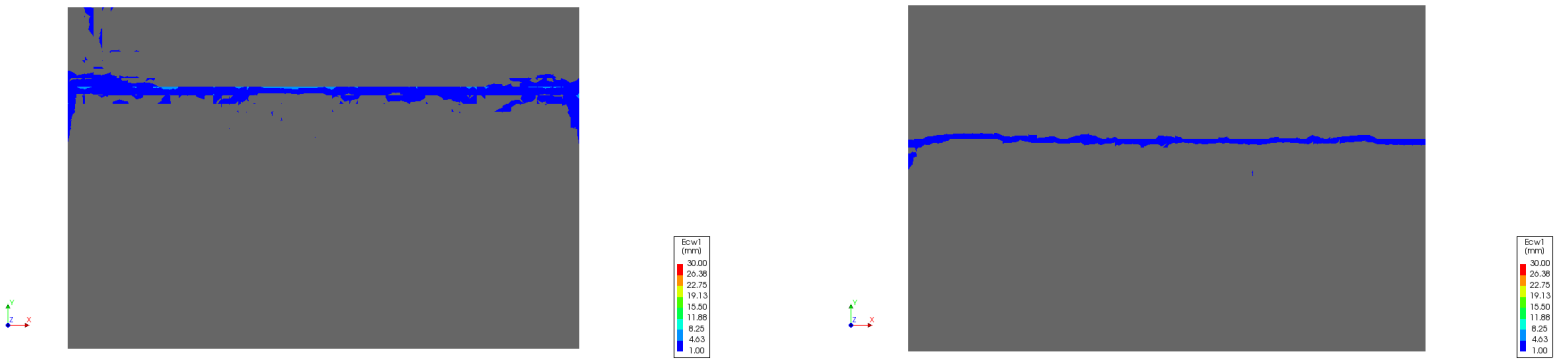

Figure 6.11: FEM: Problematic samples - Archetypes 20 (Left) and 21 (Right)

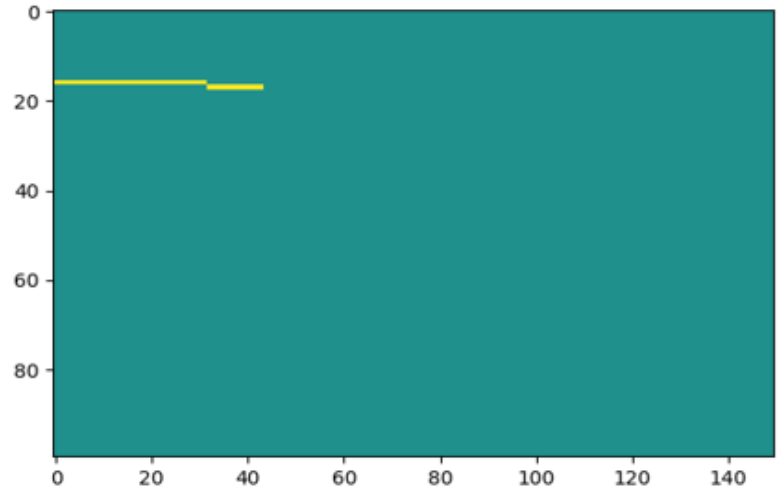

Figure 6.12: Crack Archetype 24

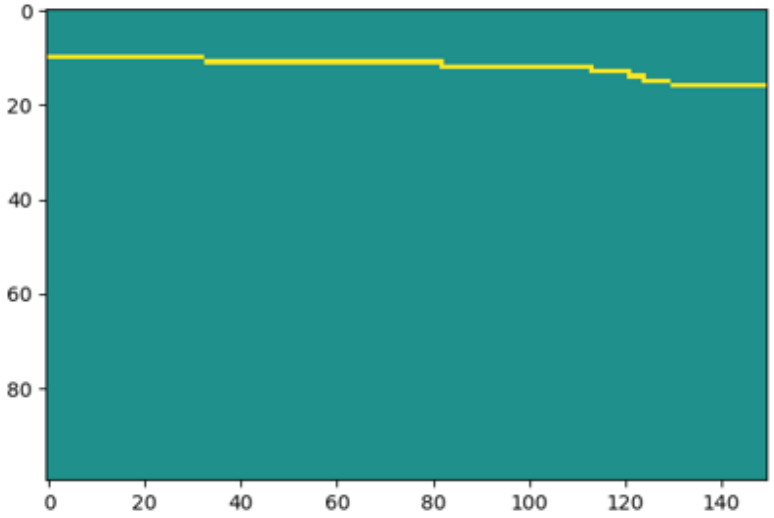

Figure 6.13: Crack Archetype 30

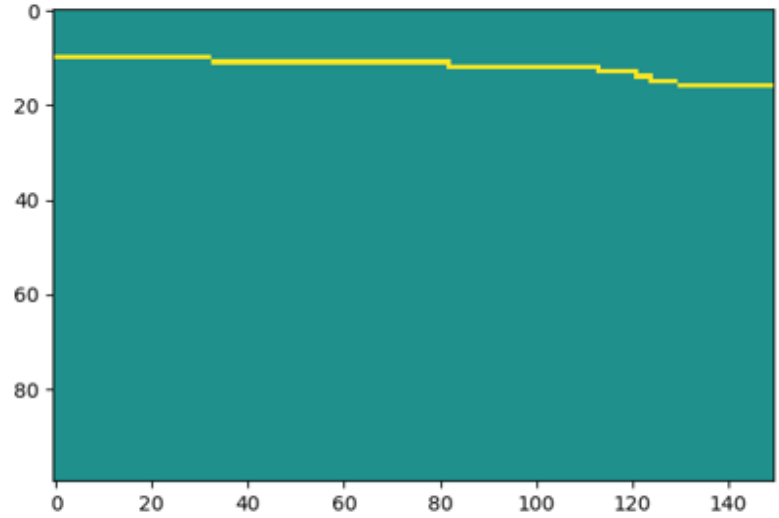

Figure 6.14: Crack Archetype 24

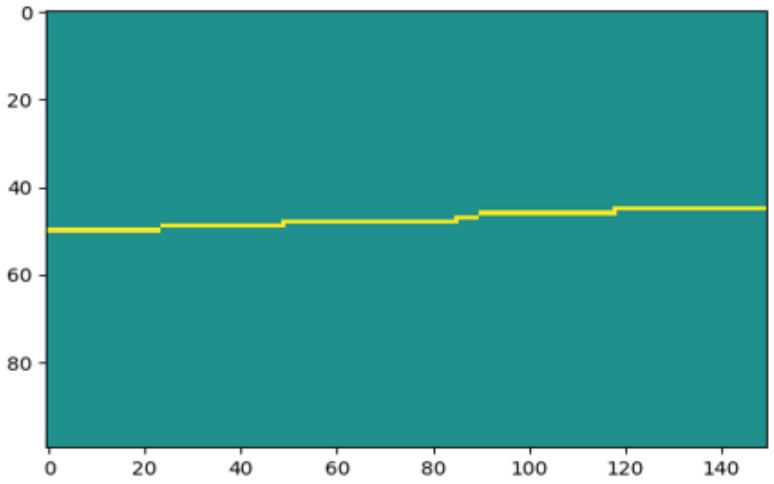

Figure 6.15: Crack Archetype 30