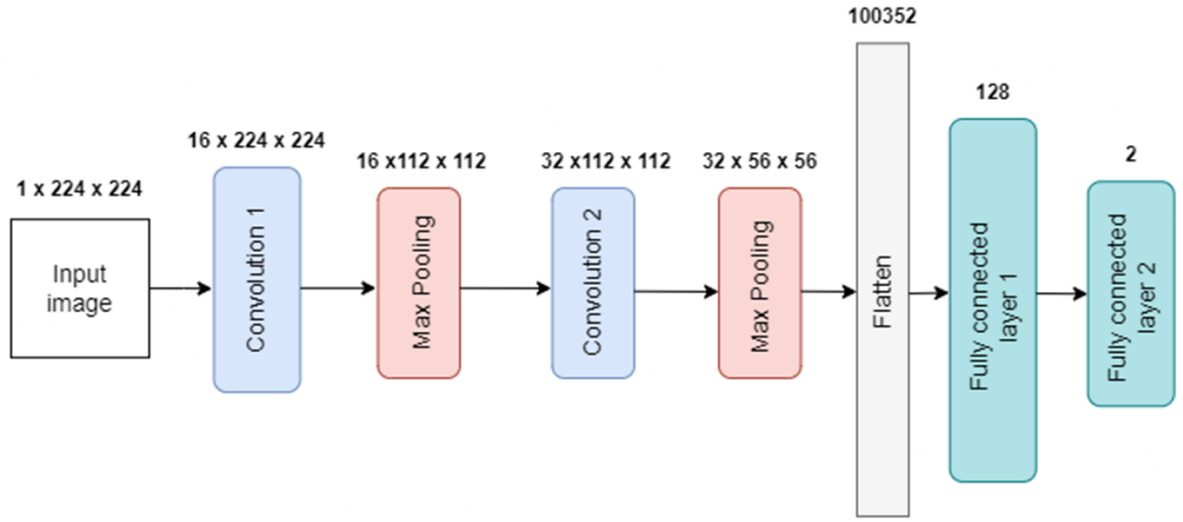
Deliverable 3: Concrete Crack Detection

The objectives of this project are defined as follows:

- How does the performance of a pretrained model (ResNet18) compare with that of a custom-made simple CNN trained from scratch?
- What is the effect of the dataset on which the custom CNN is trained on the performance of the model?

Custom CNN

In deliverable 2, it was demonstrated that modifying and training ResNet18 on dataset 1 led to an increase in performance metric such as accuracy and precision. To tackle objective 1 mentioned above, a custom CNN was built. The architecture is shown below.



Two convolution layers were used for feature extraction. Two max pooling layers were added to reduce dimension, avoid overfitting and increase efficiency. The flatten layer converts the 2D feature maps from the convolution and pooling layers into a 1D vector, which is then fed into the fully connected layers for final classification. The architecture was chosen to be simple to allow for comparison with the performance of the model with the more complex ResNet18.

It must be noted that during the preprocessing stage of the dataset used for training the custom CNN, the images were converted into grayscale. However, this did not affect the comparison with the modified ResNet18 since concrete is mostly grayscale in appearance, and the conversion to grayscale did not result in a significant loss of feature information relevant for classification.

Training the model and Results

This custom CNN was trained in three different ways. First, it was trained on dataset 1, like the modified ResNet18. The training loop was also kept constant: k-fold cross-validation was used with the cross-entropy loss function. As expected, this simple model trained much

faster than the ResNet18. To compare with the performance of the ResNet18, this trained custom CNN was tested on dataset 2. The table below illustrates the performance of both models.

Comparison of Performance on Dataset 2

	Modified ResNet18	Custom CNN
Accuracy	0.8439	0.791
Precision	0.8754	0.8274
Recall	0.8439	0.791
F1-Score	0.8406	0.7851

The performance comparison between the Modified ResNet18 and the Custom CNN indicates that the Modified ResNet18 outperformed the Custom CNN across all evaluation metrics, suggesting it is a more effective model for the given task. This can be attributed to its deeper architecture, residual connections, and higher capacity to extract complex features. The Custom CNN, while simpler, may have struggled to capture as much information due to fewer layers and a possibly less optimized design.

To address objective 2 described above, the Custom CNN went through two additional rounds of training from scratch: training on dataset 2 and training on dataset 1 and 2 combined. The training loop and hyperparameters were kept constant as previously.

Final Results

We now have four different CNN models. They were all tested on a new dataset, dataset 4, to allow for comparison across the board. The performance metrics are shown in the table below.

Comparison of Performance on Dataset 4

Metric	Model 0	Model 1	Model 2	Model 3
	Modified ResNet18 trained on DS 1	Custom CNN trained on DS 1	Custom CNN trained on DS 2	Custom CNN trained on DS 1+2
Accuracy	0.9968	0.9962	0.9138	0.8790
Precision	0.9968	0.9962	0.6241	0.9005
Recall	0.9968	0.9962	0.9138	0.8790
F1-Score	0.9968	0.9962	0.9132	0.8774

It can be observed that Model 0 and Model 1 show high performance metrics on dataset 4, with the modified ResNet18 still surpassing the custom CNN by a small margin. Model 2 gives slightly lower performance. However, when trained on dataset 1 and 2 combined, model 3 shows a noticeable drop in performance metrics.

Concerning our defined objectives, the following conclusions can be drawn:

- Pretrained Model Superiority:

The modified ResNet18 consistently outperforms the Custom CNN in feature extraction and classification tasks. This is evident from the higher performance metrics achieved by the pretrained model on both Dataset 2 and Dataset 4. This underscores the advantage of using pretrained models for complex feature extraction in applications like this project.

- Custom CNN Sensitivity to Training Data:

The Custom CNN's performance is more sensitive to the quality and characteristics of the training data. The results suggest that the model's ability to generalize is closely tied to the dataset it was trained on, leading to variations in performance depending on the training data used. The quality, consistency, and domain alignment of training datasets play a crucial role in achieving optimal results.

Pretrained models like ResNet18 have been trained on large and diverse datasets (e.g., ImageNet), enabling them to learn more robust, transferable feature representations. These representations often generalize better to new domains, even if there are shifts in data distribution. On the other hand, the Custom CNN, being trained from scratch, lacks the diverse feature representations of a pretrained model. As a result, its performance is more susceptible to variations in training data, limiting its ability to adapt to domain shifts in datasets with different distributions or characteristics. The results reinforce the importance of domain adaptation in achieving robust performance across datasets.