VitCrackSeg:Crack Segmentation Using Vision Transformer

Oscar Poudel
New Jersey Institue of Tenchology
University Hts, Newark, NJ 07102
op72@njit.com

Abstract

This study explores the efficacy of a Vision Transformer (ViT) model for the task of real-time crack detection in various surfaces which is an essential component of structural health monitoring in civil engineering and developing a architecture VitCrackSeg. Utilizing a dataset designed for crack segmentation, the model was trained using Binary Cross-Entropy with Logits (BCELogit) loss further enhanced with jitter transformations to improve its generalization capabilities on unseen data. The model's performance was evaluated based on the Intersection over Union (IoU) and Dice coefficient metrics, achieving moderate scores of 0.4229 and 0.4991, respectively. Additionally, the model was trained using Mean Squared Error (MSE) loss for comparison, which yielded a lower accuracy of 45%, indicating the superior suitability of BCELogit.The findings demonstrate the potential of using advanced deep learning architectures like ViT for real-time, accurate crack detection. This approach not only aids in the rapid assessment of structural integrity but also enhances maintenance strategies through timely interventions. A prediction pipeline was developed for further application of the model in real-world scenario. This study establishes a foundation for leveraging cutting-edge Deep learning techniques in critical infrastructure monitoring, contributing to safer and more resilient civil structures.The codes are availabe at https://github.com/oscarpoudel/DL_VitCrackSeg

1 Introduction

Deep learning (DL) is transforming various fields of engineering, including civil engineering, by providing powerful techniques to enhance data analysis, automation, and accuracy in inspection and maintenance tasks. In particular, Vision Transformers (ViTs) have emerged as a groundbreaking architecture with significant potential in visual inspection tasks due to their scalability, self-attention mechanisms, and ability to process complex structural images.

The classification of building superstructures, such as aprons and ceilings, is crucial for assessing structural integrity. Traditional manual inspections are subjective and inefficient, while convolutional neural networks (CNNs) like VGG16 and advanced models like ViT and Token-to-Token Vision Transformer (T2T-ViT) deliver a more objective, scalable approach. ViT uses self-attention heads to efficiently analyze smaller image patches while maintaining accuracy and reducing computational costs, making it suitable for classifying structural components and detecting critical damage[1]. A novel ViT-based approach has also been proposed for achieving high-resolution segmentation in civil engineering inspections. Eltouny et al. (2022) highlight a hybrid framework combining ViT with learnable downsampler-upsampler modules for managing global and local semantics trade-offs in high-resolution inspection images, while retaining both fine local details and global contextual information. This combination ensures high fidelity in structural analysis, enabling accurate detection of cracks, spalling, and other structural defects [2]. Another key challenge is defect classification in various environments. Chow et al. (2020) proposed an artificial intelligence pipeline using

deep learning for anomaly detection and defect classification in concrete structures. This pipeline combines deep classifiers with anomaly extraction methods to identify and classify defects under varying conditions like lighting, camera angles, and distances, achieving an average testing accuracy of 95.6 Furthermore, Hoskere et al. (2022) emphasized the importance of ViT in 3D synthetic

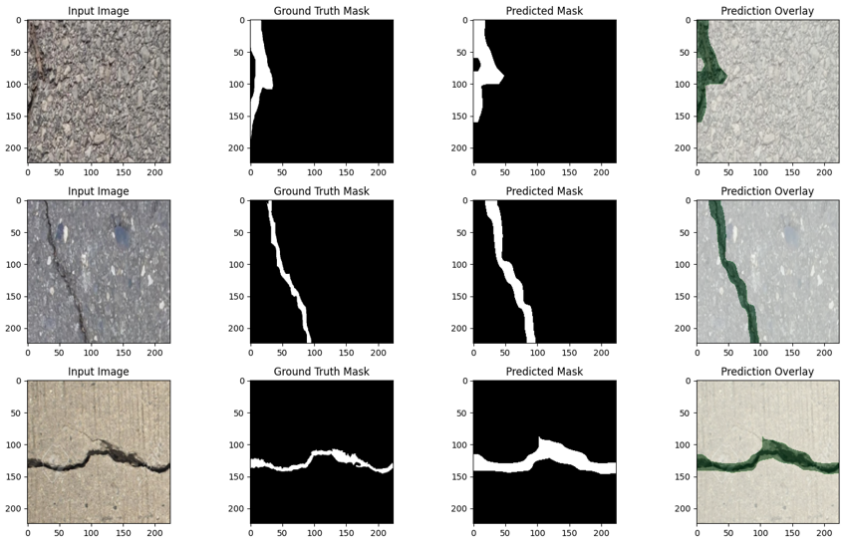

Figure 1: Cracksegmentation results on testing set after feeding them through VitCrackSeg models (first column being input, second column groundtruth, third column predicted mask and fourth column being predicted mask over input image

environments. They developed an autonomous inspection testbed using Vision Transformers to classify damage and structural conditions in virtual earthquake-damaged buildings, demonstrating the improvement of deep learning models in real-world damage inference [4]. Another innovative application involves active learning strategies to tackle imbalanced datasets. Frick et al. (2022) proposed a method using a binary discriminator to prioritize data annotation tasks for training ViT models in civil engineering inspections. Their method improved 38% over traditional active learning methods[5].

This research project particularly emphasizes the identification of concrete cracks, corrosion, spalling, and other damage types to ensure the structural health of buildings and infrastructure. By leveraging the powerful features of deep learning and Vision Transformer (ViT) models, engineers can improve the precision of defect detection and classification. Additionally, the incorporation of real-time monitoring through camera systems enables immediate detection and response, ultimately leading to safer and more resilient infrastructure. The segmentation result from the model are shown in Figure 1

2 Method

2.1 Dataset

The Crack Segmentation Dataset available on Kaggle represents a significant resource for researchers and engineers dedicated to the advancement of structural health monitoring and the development of cutting-edge image processing technologies. This dataset combines together approximately 11,200 high-resolution images sourced from 12 distinct crack segmentation datasets.

Comprehensive Data Collection: Each image within this dataset has been standardized to a uniform resolution of 448x448 pixels which ensured consistency across data inputs. This uniformity allows for streamlined processing and analysis, eliminating the need for additional preprocessing steps to normalize image sizes from various sources.

Detailed Annotations and Masking Accompanying each image are corresponding masks that outline the crack patterns. These masks provide labels that guide the algorithm in distinguishing between

cracked and uncracked regions, a critical factor in developing reliable and accurate crack detection systems.

Data Structure and Accessibility The dataset is organized into four main folders: Images: Contains all the high-resolution images. Masks: Includes the corresponding masks for each image, highlighting the crack patterns. Train: A subset of images and masks designated for training purposes. The selection of images in this folder ensures a representative mix of various conditions and types of cracks. Test: Contains images and masks for testing and validating the performance of developed models. This split is stratified, maintaining a proportional representation of each source dataset to ensure comprehensive testing conditions.

Non-Crack Data Inclusion Moreover, the dataset includes images labeled with the prefix "non-crack*", indicating the absence of crack features within these images. This inclusion is important for training models capable of correctly identifying negative instances where no cracks are present, thereby enhancing the model's accuracy and generalability in real-world applications.

Use Cases and Applications The Crack Segmentation Dataset is designed not only for academic research but also for practical applications in civil engineering, architecture, infrastructure maintenance, and disaster response. By facilitating the development of algorithms that can automatically detect and analyze structural cracks, this dataset plays a crucial role in preventative maintenance strategies and safety inspections of buildings, bridges, and other critical structures.

2.2 Preprocessing

To ensure reproducibility in experiments that involve stochastic processes such as data shuffling and weight initialization, seeds for the random number generators are initialized at the beginning of the script. Specific seed values are set for both NumPy and PyTorch, with a chosen seed of 1010. This step is crucial as it makes the model's behavior deterministic, ensuring that results can be replicated exactly across different runs and on different machines. The configuration of the model's training parameters is established early in the script. The batch size is set to 32 which is the number of samples processed before the model updates its internal parameters. The model is defined to recognize one class as a segmentation is a binary classification task. Training is set to run for 60 epochs, allowing the model sufficient time to learn from the data. A DataFrame is constructed to associate each image with its corresponding mask, simplifying the process of accessing the data during model training. The dataset is split into training, validation, and test sets, with 30% of the data allocated to the test set and 15% of the remaining training data allocated to validation. This stratified split ensures the model is evaluated on a representative sample of data, enhancing its ability to generalize well to new, unseen data.

The images and masks undergo several preprocessing steps to prepare them for use in training the neural network. All images and masks are resized to 224x224 pixels to ensure consistency in input data size. A series of transformations is applied to both the training and validation datasets, which includes converting the images to the PIL format, resizing, and normalizing. The normalization uses mean values of [0.485, 0.456, 0.406] and standard deviations of [0.229, 0.224, 0.225], which are used for models pre-trained on the ImageNet dataset. Custom dataset classes and DataLoaders are set up to manage data loading and batching efficiently. These components are essential for handling different aspects of data management during the training process, such as shuffling the training data and batching it for each iteration. This setup not only streamlines the training process but also ensures that the model is fed with correctly formatted and preprocessed data vital for the successful training of deep learning models, especially in complex tasks such as crack segmentation in images.

2.3 Model Architecture

The core of our deep learning approach involves the Vision Transformer (ViT), specifically leveraging the pre-trained ImageNet-21k ViT model available through Hugging Face's Transformers library, configured for image sizes of 224x224. Vision Transformers are particularly suitable for this task due to their ability to process images as sequences of patches, applying self-attention mechanisms that consider the global context of the image, which is vital for accurately detecting and segmenting cracks. The architecture of the model is shown in Figure 2.

Removing the Classification Head

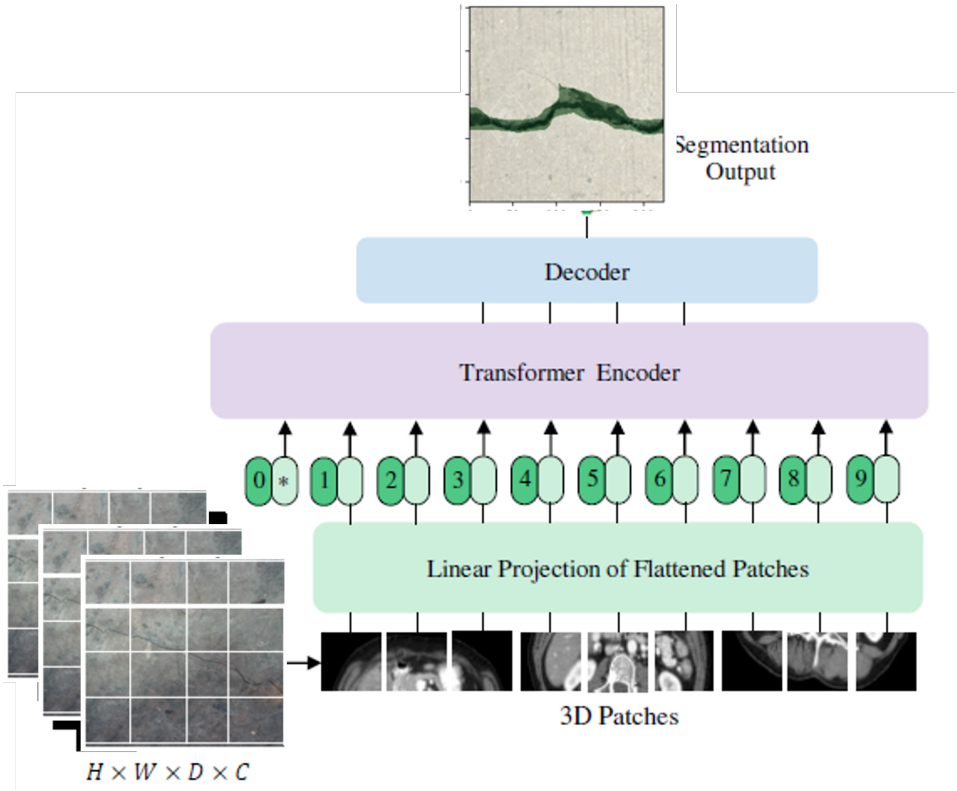

Figure 2: Representative architecture of the model

The standard ViT model is typically used for classification tasks. However, for the purpose of segmentation, the classification head of the pre-trained model is removed to adapt the model for generating image features rather than class outputs. This modification allows the transformer to output a sequence of embeddings corresponding to different patches of the image, which are then used for the segmentation task.

Segmentation Head

After processing the image through the transformer encoder and removing the classification head, the sequence of patch embeddings generated by the ViT is reshaped back into a two-dimensional spatial grid that mirrors the arrangement of patches in the original input image. This restructuring is crucial as it maps the learned deep features back to the spatial dimensions of the image, facilitating a more intuitive application of convolutional operations.

A convolutional layer with a 1x1 kernel size is subsequently applied to these reshaped features. This convolutional 'segmentation head' serves to map the deep feature representations of the patches to the desired number of output channels. For crack detection and segmentation, the output typically includes channels for different types of damage or different damage states (e.g., cracked, not cracked).

Upsampling

Due to the division of the original image into patches and the downscaling involved in matching them to the transformer's input resolution, the output from the segmentation head usually has a lower resolution than the original image. To address this, an upsampling step is incorporated, where the output from the segmentation head is upscaled back to the dimensions of the original image. This upsampling is typically performed using bilinear interpolation, which helps in smoothing the output segmentation map and aligning it closely with the ground truth masks. This configuration not only preserves the high resolution of the input images but also ensures that the segmentation maps generated by the model are precise and detailed, enabling accurate pixel-wise classification of the structural integrity of the materials being analyzed.

```python
class ViTForSegmentation(nn.Module):
    def __init__(self, num_classes, output_dim=(224, 224)):
        super(ViTForSegmentation, self).__init__()
        self.vit = ViTForImageClassification.from_pretrained('google/vit-base-patch16-224-in21k')
        self.segmentation_head = nn.Conv2d(768, num_classes, kernel_size=1)
        self.upsample = nn.Upsample(size=output_dim, mode='bilinear', align_corners=False)

    def forward(self, x):
        features = self.vit.vit(x).last_hidden_state

        batch_size, sequence_length, num_channels = features.size()
        # first token is the classification token and should be discarded for segmentation
        features = features[:, 1:, :]  # Skip the classification token

        side_length = int((sequence_length - 1)**0.5)  # Adjust for the removed classification token

        # Now the number of patches should form a perfect square
        assert side_length * side_length == (sequence_length - 1), "Number of patches does not form a perfect square"

        features = features.permute(0, 2, 1).reshape(batch_size, num_channels, side_length, side_length)

        # Apply the segmentation head and then upsample
        output = self.segmentation_head(features)
        output = self.upsample(output)
        return output
```

Figure 3: Code for the Model

By integrating the Vision Transformer with a tailored segmentation head and upsampling strategy, this methodology leverages the strengths of transformers for feature extraction while adapting them to the specific needs of high-resolution image segmentation tasks in civil engineering. This approach ensures that the spatial relationships and detailed textural information crucial for crack detection are effectively captured and utilized, resulting in highly accurate segmentation outputs that are critical for structural health monitoring and assessment.

Code

As shown in the Figure 3, the ViTForSegmentation class, implemented in Python using the PyTorch framework, is central to our model's architecture. It adapts a pre-trained Vision Transformer for image segmentation by attaching a convolutional segmentation head that classifies each image patch based on its features.

In the forward pass, after processing the input through the Vision Transformer to obtain the last hidden states, the classification token is discarded. The patch embeddings are then reshaped into a 2D spatial grid. This structured arrangement of features is crucial for maintaining spatial coherence, which allows the subsequent convolutional layer to effectively perform segmentation based on localized image features. The convolutional segmentation head then maps these features to the number of desired output channels for each class in the segmentation task. This process effectively transforms the deep feature representation into class-specific segmentation predictions for each patch.

The output from the segmentation head usually has a lower resolution due to the division into patches. To address this, the output is upsampled back to the original image dimensions using bilinear interpolation. This upsampling step is vital as it ensures that the final segmentation map is both precise and aligned with the high-resolution input images, facilitating accurate pixel-wise segmentation. This part of the methodology leverages the detailed textural information captured by the Vision Transformer, applying it effectively for the precise localization required in the segmentation tasks. This approach not only retains the integrity of the structural analysis but also enhances the model's ability to detect minute details crucial for assessing the condition of infrastructure materials.

2.4 Training

In the training process, the study utilized the Binary Cross-Entropy with Logits (BCEWithLogitsLoss) as the loss function. The optimizer of choice was the Adam optimizer with a set learning rate of 0.0001, facilitating fine-grained adjustments in the model weights. Data was processed in manageable batches of 32, which balanced the computational load and allowed for efficient gradient updates.

To ensure robustness and prevent overfitting, several regularization techniques were employed including gradient clipping and early stopping. Gradient clipping was used to prevent the exploding gradient problem by capping the gradients during backpropagation to not exceed a defined threshold.

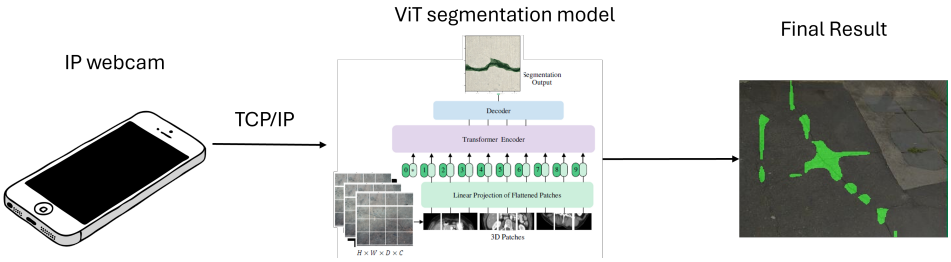

Figure 4: Prediction pipeline for the crack segmentation application

Early stopping was triggered after 10 epochs indicating that continuing training would likely lead to overfitting.

The training was logged and monitored using TensorBoard, which provided a visual interface to track and analyze metrics like loss and accuracy over time, helping to fine-tune the training phases. The model's training was conducted on a high-performance hardware setup consisting of 32 GB of RAM and an NVIDIA RTX 3090 16 GB GPU. The final model comprised a significant number of trainable parameters, totaling 85,800,963, which underscores the complexity and depth of the network architecture used. Overall, the training was completed in approximately 2 hours.

2.5 Prediction Piepline

The primary objective of the study is to implement a real-time crack detection system leveraging an IP webcam setup on a smartphone and a deep learning model running on a laptop. The prediction pipeline is shown in Figure 4. Data acquisition involves the smartphone configured as an IP webcam capturing live video footage, streamed over a TCP/IP network to a laptop for processing. Each video frame is analyzed using a Vision Transformer (ViT) segmentation model resulting precise segmentation masks overlayed on the original video frames. This segmented output is immediately displayed on the laptop, offering real-time visualization crucial for applications like in-field structural assessments or continuous infrastructure monitoring.This methodology combines advanced deep learning techniques with accessible technology to offer a robust tool for real-time safety and maintenance assessments across diverse industries.

The evaluation was conducted using two different metrics: Intersection over Union (IoU) and Dice coefficient, alongside traditional accuracy measures from model training with Mean Squared Error (MSE) loss.

Training and Validation Loss and Accuracy The training process was documented through a series of plots showing both training and validation loss and accuracy under two conditions: with and without jitter transformation. These conditions are crucial for understanding the model's robustness to input variations.

3 Results

Figure 5plot (a) and (b) illustrate the training and validation loss and accuracy when trained with Binary Cross-Entropy with Logits (BCELogit) loss without jitter. It is observed that the training loss decreases steadily, indicating good model convergence. Correspondingly, the training accuracy increases which reaches a plateau, which suggests that the model effectively learns from the training data without overfitting, as shown by the parallel trend in validation accuracy. Figure 5plot (c) and (d) display the results for training with BCELogit loss with jitter transformations. The inclusion of jitter, which introduces variations in the training images (such as changes in brightness or contrast), shows a slightly more erratic progression in both loss and accuracy curves. However, the model still improves over time, demonstrating adaptability to augmented data. Notably, the validation accuracy here does not plateau as distinctly as in the non-jitter case, possibly indicating a model that generalizes better on unseen data due to the diversity in training examples.

Performance Metrics

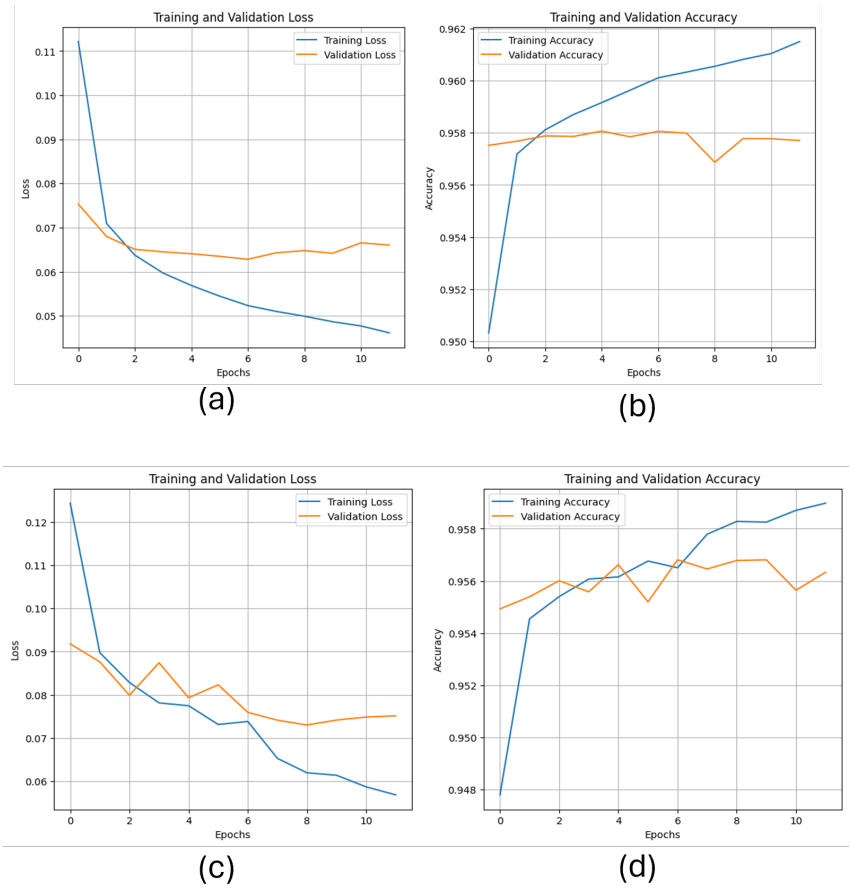

Figure 5: Training and validation (a)loss (b)accuracy for training with BCELOGIT loss without jitter. Training and validation (a)loss (b)accuracy for training with BCELOGIT loss with jitter transforms

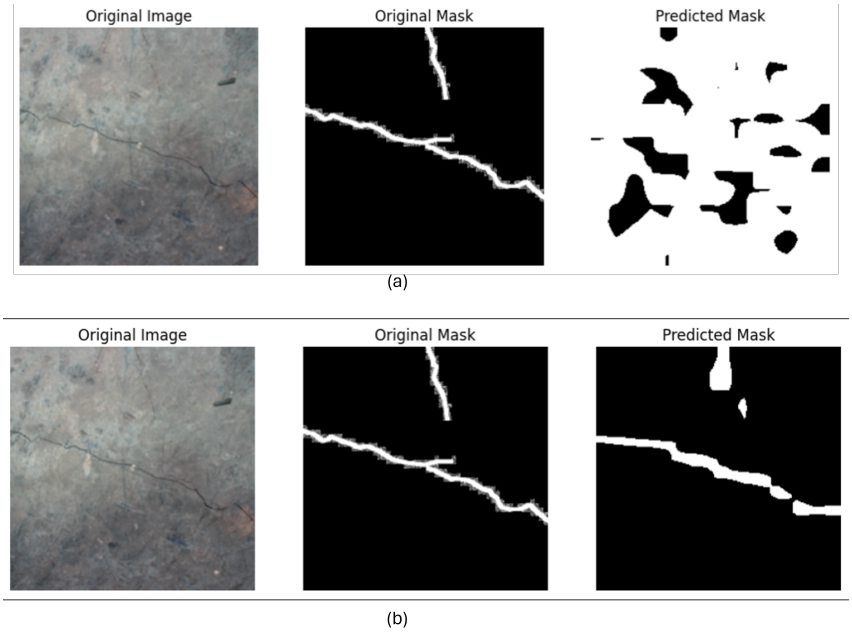

Figure 6: (a)Prediction from final model with MSE loss (b) Prediction from final model with BCElogitloss

Intersection over Union (IoU): The average IoU achieved by the model is 0.4229. This metrics indicates a moderate level of accuracy in the model's segmentation capability.
Dice Coefficient: Similarly, the Dice coefficient, which is another method of measuring the efficacy of the segmentation by measuring the overlap twice the intersection over the sum of the sizes of two labels, is recorded at 0.4991. This value further supports the findings of the IoU, showcasing a nearly 50% effectiveness in identifying the correct regions of interest in the images. Model

Training with MSE Loss: When the model was trained using MSE loss, a commonly used loss function for regression tasks, it achieved an accuracy of 45%. This result is notably lower compared to the training with BCELogit loss, suggesting that MSE might not be as effective for the segmentation tasks handled by this model, potentially due to its emphasis on averaging the errors in pixel values rather than focusing on the categorical outcomes. The prediction results for the model with MSE loss and BCElogit loss are shown in Figure 6

4 Discussion and Conclusion

The outcomes of the project hence shows a nuanced understanding of the performance dynamics with a Vision Transformer (ViT) model under a changed operational framework for image segmentation task focusing in the crack detection. Applying BCELogit with augmentation of jitter transformations, the model learned to generalize well enough to present an average IoU of 0.4229 and Dice coefficient of 0.4991 over unseen data. These metrics are of medium value; however, it can be inferred that the model might help in successful crack pattern recognition performance, which is critical for real-time structural health monitoring applications to successfully identify and segment out crack patterns. Training the model with Mean Squared Error (MSE) loss gives an accuracy level that is drastically less effective; it gives a reduced 45% accuracy level. This further shows that there is a need for the right loss function, not only meant for the minimization of simple error but also one that is suited for the specific given segmentation task.

In conclusion, the study has successfully demonstrated the application of a deep learning model, leveraging a Vision Transformer architecture, for the segmentation of cracks in images—a task with significant implications in the fields of civil engineering and infrastructure maintenance and developed a model VitCrackSeg. The training process was optimized through strategic choices in loss functions and regularization techniques, ensuring that the model was not only accurate but also robust against overfitting and efficient in terms of computational resources. The use of advanced hardware, along with systematic training approaches, resulted in a model that was trained in a relatively short time frame, showcasing the feasibility of deploying such models in real-world scenarios where quick, reliable crack detection is essential. The early stopping of training at 10 epochs due to non-improving validation loss metrics is indicative of a well-tuned training process that prioritizes quality of learning over mere completion of arbitrary training cycles. The prediction pipeline gives a tangible method for the application of the deep learning model in the real world. Future work should focus on further enhancing the model's accuracy and generalization capabilities, possibly by exploring more diverse datasets and implementing more complex transformations and augmentation strategies. Additionally, experimenting with other loss functions like dice loss and optimization techniques could provide deeper insights into the model's performance and lead to higher precision in crack segmentation tasks. The insights gained from this project underscore the potential of using advanced machine learning techniques for critical applications in infrastructure health monitoring, with a view towards improving safety and longevity of civil structures.

References

[1] J. T. S. Phang, K. H. Lim, and C. H. Lim, "Deep Learning Neural Networks in Building Superstructure Classification," in 2023 International Conference on Digital Applications, Transformation Economy (ICDATE), Miri, Sarawak, Malaysia: IEEE, Jul. 2023, pp. 1–5. doi: 10.1109/ICDATE58146.2023.10248658.

[2] K. Eltouny, S. Sajedi, and X. Liang, "High-Fidelity Visual Structural Inspections through Transformers and Learnable Resizers." arXiv, Oct. 21, 2022. doi: 10.48550/arXiv.2210.12175.

[3] J. K. Chow et al., "Artificial intelligence-empowered pipeline for image-based inspection of concrete structures," Autom. Constr., vol. 120, p. 103372, Dec. 2020, doi: 10.1016/j.autcon.2020.103372.

[4] V. Hoskere, Y. Narazaki, and B. F. Spencer, "Physics-Based Graphics Models in 3D Synthetic Environments as Autonomous Vision-Based Inspection Testbeds," Sensors, vol. 22, no. 2, p. 532, Jan. 2022, doi: 10.3390/s22020532.

[5] T. Frick, D. Antognini, M. Rigotti, I. Giurgiu, B. Grewe, and C. Malossi, "Active Learning for Imbalanced Civil Infrastructure Data." arXiv, Oct. 19, 2022. doi: 10.48550/arXiv.2210.10586.