GROUP 4 DEEP LEARNING PROJECT
ROBUST MODELS FOR DETECTING BRIDGE DEFECTS UNDER NATURAL DISTRIBUTION SHIFTS

Daniel Anthony
Graduate Data Science Student
University of Virginia
Charlottesville, VA 22904
dja8tx@virginia.edu

Eric Rodriguez
Graduate Data Science Student
University of Virginia
Charlottesville, VA 22904
vuh5mk@virginia.edu

Kanitta Srichan
Graduate Data Science Student
University of Virginia
Charlottesville, VA 22904
uay3yb@virginia.edu

Nicholas Miller
Graduate Data Science Student
University of Virginia
Charlottesville, VA 22904
dkr5ud@virginia.edu

December 12, 2024

ABSTRACT

Maintaining bridge safety is critical for public welfare, but traditional inspection methods are labor-intensive and prone to human error [5]. This project utilized machine learning, specifically convolutional neural networks (CNNs), to automate the detection of cracks in concrete via image classification. Leveraging datasets such as SDNET2018, CNN models were trained and evaluated to identify various defects [4]. Related work highlighted the potential of deep learning for structural health monitoring, demonstrating its advantages in accuracy and scalability [2, 6]. Our solution achieved notable improvements in defect detection accuracy and efficiency compared to traditional methods. Our work addressed challenges related to natural distribution shift and adversarial attacks [1, 3] and demonstrated that large and diverse training data improved robustness against natural distribution shifts, a critical factor for real-world infrastructure monitoring [9].

1 Motivation

Image classification with CNNs presents an opportunity to automate the identification of cracks in concrete, enhancing both accuracy and speed compared to manual methods [5, 6]. Despite these promising prospects, challenges remain in ensuring robustness across diverse real-world image conditions [2]. Our project evaluates CNN models' resilience to natural distribution shifts commonly observed in real-world images and explores methods to improve their performance under these conditions [1]. Additionally, we assess the models' robustness to adversarial attacks by implementing the gradient sign attack, also known as the Fast Gradient Sign Method (FGSM), using the Foolbox library [3].

2 Literature Review

Distribution shift is an area of image classification that requires continued research, particularly with natural distribution shift. According to Taori et al., [1] most research on robustness focuses on synthetic image perturbations. Although synthetic distribution shifts allow researchers to test models during development and understand which features they learn in accordance with the carefully controlled variables, the robustness of these models when deployed and tested on images with natural distribution shifts is an area of research that demands further study. One of the major conclusions

that drives the structure of our experiment is the notion that training on more diverse data improves robustness. Our experiment aims to confirm this finding by training a model on a more diverse dataset and a more homogeneous dataset, with respect to natural distribution shifts, and compare the accuracy on both a homogeneous and shifted test set.

The next major component for this experiment is determining how to evaluate robustness. Taori and Liu [1], [7] each propose methods for evaluating robustness. Taori defines the terms effective robustness and relative robustness. Effective robustness disentangles the shifted accuracy scores from the standard accuracy scores by evaluating the difference between the two test scores for each model. It is expected that as improvements in accuracy on the standard set increase, there is a consistent linear improvement on the distributed test set. An effectively robust model will go beyond what is expected from having higher accuracy on the standard test set. Relative robustness refers to maintaining an objective increase in the accuracy of the model with an intervention relative to the accuracy of the baseline model. Although our initial intent was to evaluate both measures of robustness, our project limited the scope to evaluate relative robustness for the targeted interventions. Additional work can be done to construct a beta function and incorporate the effective robustness into the results.

Liu [1] developed a benchmark called ARES-Bench that they used to evaluate the robustness of ImageNet models with diverse architectures (CNNs, Transformers) and learning algorithms (pre-training, adversarial training). One of the major findings is that there is an intrinsic tradeoff between adversarial and natural robustness for the same model architecture. Although this research does not have a framework for natural robustness beyond defining the classification accuracy, they have developed a tool to evaluate adversarial robustness and shared the code on GitHub. Our project ultimately chose the foolbox library as a way to implement adversarial attacks on the images with gradient sign attack, also known as fast gradient sign method, FGSM [3].

Although data for structural health monitoring is collected using sensors and nondestructive techniques such as acoustic emission and guided waves [2], our focus is on how artificial neural networks are using image data to classify concrete cracks. With regards to this subject, convolutional neural networks (CNNs) are the primary technique researched. Abubakr et al. [4] compared the performance of the Xception architecture with vanilla CNNs on classifying multiple defect types of reinforced concrete bridges and determined that Xception has a higher percentage of accuracy (95%) versus the vanilla CNNs (85%). For our experiments, we will plan to use transfer learning where possible and utilize the performance of pre-trained models.

Arafin [6] proposes three categories for damage detection using CNNs: image patch method, boundary box regression, and semantic segmentation. Compared to the first two, semantic segmentation provides a more quantitative assessment of damage characteristics by outlining the shape of the crack instead of using a box or patches to show where the crack is located generally in the image. Since our proposal is to simply classify images with or without cracks, the team will not use this technique. Once a model that is robust to natural distribution shift is developed, there is potential for future research here to relay more information to the engineering team regarding the size of the cracks. Arafin [6] proposes U-Net and PSP-Net for the semantic segmentation and uses stochastic gradient descent (SGD) and adaptive moment estimation (ADAM) as optimization functions with pre-trained weights from ImageNet. Although our team may not utilize the three damage detection methods, the authors' testing of optimizers, learning rates, and transfer learning will prove useful in our experiments.

Taori et al. [1] bring attention to a gap in robustness research, specifically the over-reliance on synthetic perturbations to evaluate image classification models. While synthetic shifts such as adding noise or blurring are useful for controlled experiments, the study critically exposes how these models often fail to generalize to natural distribution shifts encountered in real-world environments. This highlights a significant shortcoming in current research, where the deployment environment is not well understood, giving an inflated sense of robustness. The failure to test on real-world shifts, such as variations in lighting, weather, or sensor quality, demonstrates a lack of practical applicability. Taori's study calls for the development of more comprehensive robustness metrics beyond standard accuracy, such as accuracy under distribution shift and generalization error across natural shifts. There is an opportunity for future research to explore proactive solutions to create training data that mirrors real-world variability, such as transfer learning from naturally shifted domains or more advanced data augmentation techniques.

3 Dataset

Dataset Description: For this study, we utilized the SDNET2018 dataset. This dataset contains over 56,000 images of cracked and non-cracked concrete bridge decks, walls, and pavements. The SDNET2018 dataset captures real-world challenges with natural variations in images, such as lighting conditions, shadows, varied scales of cracks, focus, and differences in background colors and shading [9].

Dataset Summary: Table 1 provides an overview of the SDNET2018 dataset and its characteristics.

Table 1: SDNET2018 Dataset Summary

Category	Details
Dataset Name	SDNET2018
Total Images	56,000
Content	Cracked and non-cracked concrete bridge decks, walls, and pavements
Real-World Challenges	Lighting, shadows, varied scales, focus, and background colors/shading
Utilized Folders	D (Homogeneous images) and P (Heterogeneous images)
Subfolder Prefixes	U (Uncracked) and C (Cracked)
Folder Characteristics	- D Folder: Homogeneous images with consistent colors and scales - P Folder: Heterogeneous images with significant feature variations
Data Preprocessing	- Removed mislabeled images from CD and UD subfolders - Moved mislabeled images to "not used" folder - Improved baseline performance

Dataset Structure: The data used in this study was sourced from the 'D' and 'P' folders within the dataset. Each folder contains two subfolders, with the prefixes "U" and "C" indicating uncracked and cracked images, respectively. The images in the 'D' folder are considered the standard or homogeneous set due to similar colors and scales throughout the folder. In contrast, the 'P' folder images exhibit significant feature variations and are regarded as the shifted or heterogeneous set.

Data Cleaning: During preprocessing, it was observed that the "CD" and "UD" folders contained mislabeled images. These images were manually removed and relocated to a "not used" subfolder. This adjustment led to noticeable improvements in baseline model performance.

Preprocessing Steps:

- **Labeling:** Images were assigned labels of 1 if located in folders starting with 'C' (cracked), and 0 if in folders starting with 'U' (uncracked).
- **Resizing:** All images were resized to 64×64 pixels to ensure uniformity.
- **Color Channels:** Images were converted to three-channel RGB format using the `convert('RGB')` method.
- **Normalization:** Pixel values were normalized to the range $[0, 1]$ to improve model convergence and performance.

Data Splitting: After preprocessing, the dataset was split into training and testing sets stratified by label. Stratified splitting ensures that the proportion of labels in the overall dataset is preserved in both subsets. This approach allows the model parameters to be optimized using the training set while performance evaluation on the test set measures the models' ability to generalize to unseen data.

Statistical Summary: Table 2 provides a summary of the dataset statistics after preprocessing.

Table 2: Dataset statistics after preprocessing.

Category	Number of Images	Proportion (%)
Cracked (C)	47,608	86.04
Uncracked (U)	7,719	13.96
Total	55,327	100.0

4 Method

The primary techniques used to train the CNN for both the baseline models and experimental models were transfer learning, regularization, and learning rate reduction. Transfer learning leveraged the pre-trained VGG16 model, allowing our fine-tuning process to focus on other techniques rather than designing an architecture from scratch. This avoided the computational cost of training the base model and accelerated learning by utilizing VGG16's ability to interpret image data.

The regularization techniques employed included data augmentation, early stopping, and dropout, all aimed at reducing overfitting and generalization error. Data augmentation techniques such as flipping, rotation, and zooming were applied

Figure 1: Sample images from the SDNET2018 'D' Folder showing standard examples.

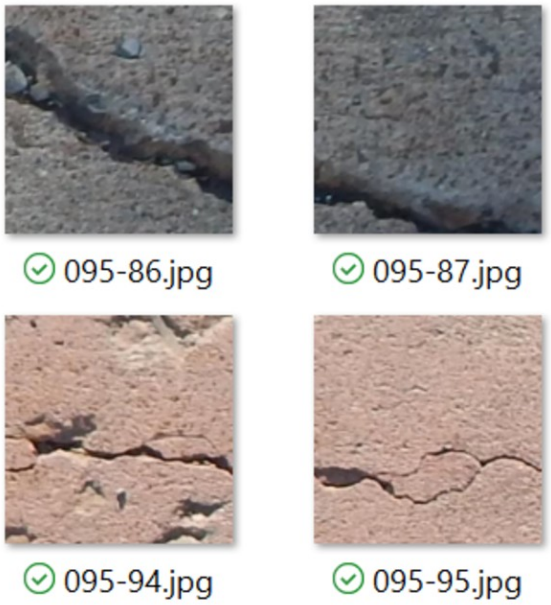

Figure 2: Sample images from the SDNET2018 'P' Folder showing shifted examples.

to simulate a more diverse dataset during training. Early stopping monitored validation accuracy and halted training after three epochs without improvement, ensuring the model did not overfit the training data. Dropout was applied to the fully connected layers to improve robustness and prevent reliance on specific pathways through the network.

The learning rate of the Adam optimizer was dynamically adjusted, reducing after two epochs if validation accuracy plateaued. This ensured better convergence and stability during training.

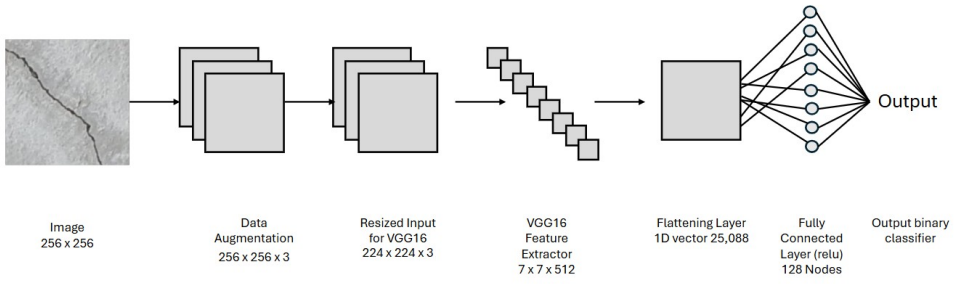

Figure 3: Baseline Model Architecture

For the experimental models, additional techniques included handling class imbalance using class weights, fine-tuning the last 10 layers of the VGG16 base model, and optimizing the classification threshold. Class weights prioritized underrepresented classes in the dataset, addressing class imbalance. Fine-tuning the last 10 layers allowed the model to adapt domain-specific features, improving its accuracy for defect detection. Optimizing the classification threshold improved the balance between sensitivity and specificity, tailoring the model for practical applications.

These enhancements built on the baseline model's strong foundation, improving performance, robustness, and real-world applicability for defect detection. The model architecture diagrams in Figure 3 and Figure 4 showcase the transition from the baseline to the experimental model design.

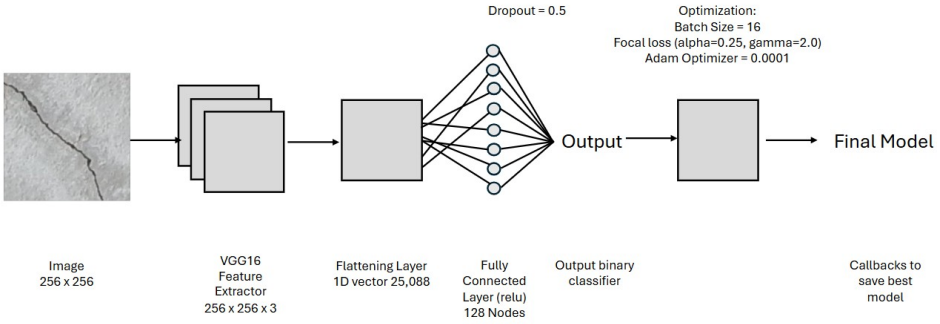

Figure 4: Experimental Model Architecture

5 Experiments

Our primary goals were to evaluate the models' robustness to natural distribution shifts using relative robustness, as proposed by Taori et al.[1] and to verify whether the size and diversity of the training set influence this metric. Additionally, the baseline architecture was tuned to determine if robustness could be improved. The robustness of the baseline and second iteration models to adversarial attacks was also evaluated using the Foolbox FGSM attack Rauber et al. [3]

5.1 Experimental Setup

The experimental setup included two test sets and three training sets:

- **Test Sets:**
  - **'P' Test Set:** A diverse and naturally shifted dataset comprised of images from the 'P' folder.
  - **'D' Test Set:** A more homogeneous and standard dataset comprised of images from the 'D' folder.

  The performance of each model on these test sets was used to calculate relative robustness to natural distribution shifts.
- **Training Sets:**
  - **'A' Large and Diverse Training Set:** Includes images from both the 'P' and 'D' folders.
  - **'B' Small Diverse Training Set:** Comprises images from both the 'P' and 'D' folders but with fewer samples than the large training set.
  - **'C' Small Homogeneous Training Set:** Includes only images from the 'D' folder.

Table 3: Training Dataset statistics after preprocessing.

Training Set	Number of D Images	Number of P Images
A	5,141	9,733
B	514	973
C	5,141	0

5.2 Experimental Procedure

The results for the experiments are found in Table 6. The accuracies used to evaluate relative robustness were all taken from performance on the shifted test set, P. The experiments were conducted as follows:

1. **Impact of Dataset Size:** The relative robustness of the baseline models trained on the large and small diverse training sets, 'A' & 'B', were compared to assess whether larger datasets lead to improved robustness. The experimental models trained on 'A' & 'B' were also compared.

2. **Impact of Dataset Diversity:** The relative robustness of the baseline models trained on the two small training sets, 'B' & 'C', were compared to determine if diversity in training data influences robustness.
3. **Impact of Architectural Changes:** The models trained on training set A with the experimental architecture and baseline architecture were compared to evaluate whether the architectural changes improved robustness.
4. **Adversarial Robustness:** The baseline experimental models were subjected to adversarial attacks using the Foolbox FGSM attack [3]. The robustness metric measures the models' ability to maintain accuracy with perturbations in the images. The metric used is between 0.0 and 1.0 and reflects the accuracy of the classification after the attacks.

5.3 Evaluation Metrics

The primary metric for evaluating robustness was **relative robustness to natural distribution shifts**, as defined by Taori et al. [1] Additionally, adversarial robustness was quantified based on the models' performance under FGSM attacks.

6 Results

The performance of the baseline models' were evaluated using accuracy and robustness metrics derived with the Foolbox library. Table 4 presents the results for all six train-test combinations. The models achieved strong accuracy across all combinations, with test set accuracies ranging from a minimum of 87% to a maximum of 97%.

We noticed that models tested on the shifted dataset, dataset 'P', overall had higher robustness than those tested on the standard dataset, dataset 'D'. The best model from the baseline results in terms of accuracy was `best_model_transfer_C_to_D`. This is likely because train set 'C' consisted only of 'D' images and was tested on only 'D' images. The model was being tested on images similar to the images it was trained on, thus performed the best overall. However, we noticed that the robustness metric for this model was the lowest.

Table 4: Baseline Architecture Performance: Accuracy and Robustness Metrics

Model	Train-Test Combination	Accuracy (%)	Adversarial Robustness
A	Best_Model_Transfer_A_to_D	96.0	0.31
A	Best_Model_Transfer_A_to_P	90.0	0.52
B	Best_Model_Transfer_B_to_D	93.0	0.51
B	Best_Model_Transfer_B_to_P	90.0	0.49
C	Best_Model_Transfer_C_to_D	97.0	0.16
C	Best_Model_Transfer_C_to_P	87.0	0.24

The models trained with 'D' and 'P' images (train sets 'A' and 'B') were exposed to a wider variety of features, potentially including some inherent noise or variability present in dataset 'P'. This exposure likely helped the models generalize better to adversarial perturbations because they encountered more diverse feature patterns during training.

In contrast, models trained solely on dataset 'D' (train set 'C') may have learned patterns highly specific to dataset 'D'. While this specificity contributed to high baseline accuracy, it also made these models more susceptible to adversarial attacks, which target these learned patterns. The train-test combination that performed the worst in terms of accuracy was `best_model_transfer_C_to_P`. This result makes sense because the model was trained on dataset 'D' images and tested on a different type of images, which were in dataset 'P'.

The experimental models demonstrated strong performance across various train-test combinations, achieving high accuracy and varying levels of robustness, reflecting a trade-off between optimizing for accuracy on standard test sets and resilience under adversarial conditions. The accuracy on standard test sets ranged from 91% to 97%, while robustness accuracy, measured under adversarial conditions, ranged from 0.29 to 0.81.

The performance of the experimental models is summarized in Table 5. Best_model_transfer_A_to_D and Best_model_transfer_C_to_D achieved the highest accuracy (97%) on standard test sets, showcasing their ability to learn highly specific patterns in the training data. However, this focus on dataset-specific patterns came at the cost of lower robustness accuracies, at 0.39 and 0.34, respectively. These results highlight a trade-off: the models excelled in familiar scenarios but struggled with adversarial perturbations that targeted the specific patterns they had learned.

In contrast, Best_model_transfer_B_to_D offered a more balanced trade-off, achieving an accuracy of 93% on the standard test set while maintaining the highest robustness accuracy (0.81). This model's exposure to both datasets 'D'

Table 5: Experimental Architecture Performance: Accuracy and Robustness Metrics

Model	Train-Test Combination	Accuracy (%)	Adversarial Robustness
A'	Best_model_transfer_A_to_D	97.0	0.39
A'	Best_model_transfer_A_to_P	95.0	0.40
B'	Best_model_transfer_B_to_D	93.0	0.81
B'	Best_model_transfer_B_to_P	93.0	0.57
C'	Best_model_transfer_C_to_D	97.0	0.34
C'	Best_model_transfer_C_to_P	91.0	0.29

and 'P' during training likely helped it generalize better, enabling it to handle adversarial conditions effectively without sacrificing too much standard test accuracy.

Models trained and tested with dataset 'P', such as Best_model_transfer_A_to_P and Best_model_transfer_B_to_P, demonstrated robustness accuracies of 0.40 and 0.57, respectively, with standard test accuracies of 95% and 93%. These results suggest that the inherent variability of dataset 'P' helped these models adapt better to adversarial challenges. However, this came with a slight reduction in accuracy compared to models tested on dataset 'D', indicating another instance of the trade-off between robustness and dataset-specific optimization.

The Best_model_transfer_C_to_P model, which was trained exclusively on dataset 'D' and tested on dataset 'P', achieved the lowest accuracy (91%) and robustness accuracy (0.29). This outcome highlights the challenge of adapting to a different dataset type during testing, as the model's learned patterns were not well-suited to the features in dataset 'P'.

Table 6: Relative Robustness Chart

Intervention	Models Compared	Relative Robustness
Size	A - B on P	0.0
Size	A'-B' on P	2.0
Diversity	B - C on P	3.0
Architecture	A'- A on P	5.0

Finally, we review Table 6 which shows the relative robustness for the experiments that include the three interventions: training set size, training set diversity, and architectural improvements.

Experiment 1: To evaluate how the size of the training set impacts relative robustness on the shifted test set, the accuracy of the large and diverse training set 'A' is compared to the small and diverse training set 'B'. Since these two models have the same accuracy on the shifted dataset, this intervention did not show an increase in relative robustness. When the same comparison is made for the testing on the standard set or with the experimental models there is an increase in accuracy for the larger 'A' models.

Experiment 2: To evaluate how the diversity of the training set impacts relative robustness, the accuracy from the small and diverse training set 'B' is compared to the small and homogeneous training set 'C'. Since the accuracy of model B on the test set is 3% greater than the accuracy of model C on the shifted test set 'P' (90% - 87%, this intervention shows an increase in relative robustness. A similar increase is seen between the experimental 'B' and 'C' accuracies on the shifted test set 'P' (93% - 91%).

Experiment 3: To evaluate how the architectural changes impact relative robustness, the accuracy of the A' and A model on the shifted test set 'P' are compared resulting in a relative robustness of 5.0. Similar improvements in accuracy are seen across the other two training sets when comparing baseline and experimental models on the shifted test set 'P'.

Overall, the experimental models underscore a clear trade-off between optimizing for accuracy on standard test sets and achieving robustness under adversarial conditions. Models that learned more diverse patterns, like Best_model_transfer_B_to_D, managed to strike a better balance, while those optimized for specific datasets excelled in accuracy at the expense of robustness. These insights emphasize the importance of considering this trade-off in designing models for real-world applications.

7 Conclusion

Our findings highlight the strong capabilities of our best model in handling the binary classification task of detecting cracked versus uncracked bridge deck and pavement images. The application of deep learning, particularly transfer

learning, in structural health monitoring (SHM) tasks shows promise in reducing maintenance costs and preventing catastrophic structural failures.

For structural health monitoring, when high-quality training data was verified through manual inspection, our best model demonstrated high accuracy and recall for both positive and negative classes. Among the pretrained architectures we experimented with from Keras applications, VGG16 emerged as the most successful. The architecture's high parameter count, shallow depth, and relatively small size likely provided an optimal environment for the training data. While other architectures such as ResNet50 excel in broader image classification tasks, they did not perform as well as VGG16 for this specific dataset.

The precision for the positive class was the weakest performance metric. This outcome aligns with our emphasis on recall for the positive class, as identifying cracked images (true positives) is more critical than minimizing false positives in the context of SHM. Our experiments also revealed that increasing dataset size improves general accuracy and traditional performance metrics but does not necessarily enhance robustness to adversarial attacks. This insight is supported by robustness metrics observed for our best model trained on sets 'A' and 'B'. Despite train set 'B' containing only 10% of the data from train set 'A', its robustness was higher, potentially due to better generalization and reduced overfitting in train set 'B'.

We used 50% of the data from the 'D' and 'P' folders in our dataset, excluding folder W from the analysis. Including the 'W' folder in future studies may provide additional useful data for the model and enrich the analysis of SHM tasks. The decision to use 50% of the data was influenced by memory and training time constraints, particularly during data augmentation, where larger subsets caused kernel crashes. Distributed computing or cloud environments may alleviate these issues in future work.

Our findings illustrate the challenges posed by relative distribution shifts. For example, the performance drop observed in `best_model_transfer_C_to_D` compared to `best_model_transfer_C_to_P` demonstrates the difficulty of testing a model on data with a different distribution than the training set. Additional data augmentation or domain adaptation techniques may help mitigate this issue. In real-world scenarios, training data may not be as clearly separated as in our dataset, where bridge deck images were separated from pavement images. By introducing heterogeneity in train sets 'A' and 'B', we aimed to simulate this real-world complexity.

The relative robustness results show how interventions such as training set size, training set diversity, and architectural improvements can lead to increased accuracy on datasets with natural distribution shifts. Although the standard architecture did not confirm that size improves robustness on shifted datasets, the experimental architecture showed this relationship so further experiments should be conducted to confirm this relationship. The diverse training sets led to improvements on the shifted test sets which confirms our prior understanding and demonstrates that training on similarly distributed data will improve accuracy. Finally, the additional fine-tuning and architectural changes did improve performance on the shifted dataset which can lead to additional studies into tradeoffs between advanced modeling and computational costs versus accuracy.

Regarding adversarial attacks, the model's performance significantly dropped after applying Foolbox perturbations. However, we attribute this to the severity of the perturbations, which are more aggressive than typical noise encountered in SHM environments, such as dim lighting or inclement weather. To contextualize these robustness metrics, future studies could use other adversarial attack methods, such as those in the Python library CleverHans, to apply less extreme perturbations.

Overall, our analysis underscores the potential of transfer learning in SHM. We recommend future research to implement a wider variety of robustness metrics and adversarial methods to better quantify the effectiveness of transfer learning models under distribution shifts. This work demonstrates the viability of transfer learning for enhancing infrastructure safety and maintenance efficiency, offering valuable insights for Virginia and similar regions.

8 Contributions

Eric developed the baseline files, implemented the Foolbox evaluation, and contributed to the preparation of the results and conclusions sections of the paper.

Kanitta designed the enhanced CNN architecture, analyzed and presented the results, contributed to the methods section, and transferred the text into overleaf.

Dan identified and relocated mislabeled images to the "not used" folder, prepared the methods and experiments sections, and reviewed the results and conclusions adding paragraphs regarding relative robustness.

Nicholas designed the CNN architecture diagrams and reviewed the final paper for formatting and errors.

References

[1] R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt, "Measuring Robustness to Natural Distribution Shifts in Image Classification," 2020. [Online]. Available: https://arxiv.org/abs/2007.00644

[2] J. Jia and Y. Li, "Deep Learning for Structural Health Monitoring: Data, Algorithms, Applications, Challenges, and Trends," 2023. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC10650096/

[3] J. Rauber, W. Brendel, and M. Bethge, "Foolbox: A Python toolbox to benchmark the robustness of machine learning models," 2018. [Online]. Available: https://arxiv.org/abs/1707.04131

[4] M. Abuakr, M. Rady, K. Badran, and S. Y. Mahfouz, "Application of Deep Learning in Damage Classification of Reinforced Concrete Bridges," 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2090447923001867

[5] M. S. Khan, H. Abbas, and M. Sadiq, "Manual inspection limitations in infrastructure health monitoring," 2016.

[6] P. Arafin, "Deep Learning-Based Concrete Defects Classification and Detection Using Semantic Segmentation," 2023. [Online]. Available: https://journals.sagepub.com/doi/10.1177/14759217231168212

[7] C. Liu, Y. Dong, W. Xiang, X. Yang, H. Su, J. Zhu, Y. Chen, Y. He, H. Xue, and S. Zheng, "A Comprehensive Study on Robustness of Image Classification Models: Benchmarking and Rethinking," 2023. [Online]. Available: https://arxiv.org/abs/2302.14301

[8] https://github.com/EricRod12/deep-learning-final-project/tree/main

[9] Utah State University, "All Datasets: Deep Learning Models for Structural Damage Detection and Classification," 2023. [Online]. Available: https://digitalcommons.usu.edu/all_datasets/48/