The Effectiveness of Data Augmentation Methods in Multi-Defect Classifications from Façade Images Using Transfer Learning

Beyza Kiper, Semiha Ergan
New York University, USA
bkiper@nyu.edu

Abstract. Falls from façades due to defects pose safety threats to public and require regular inspections. Conventional inspection methods are manual and based on the expertise of inspectors resulting in undetected defects and subsequent incidents and accidents. Opportunities that enable vision-based identification of defects, such as deep learning (DL), are available but require abundant labelled images for robust solutions. Yet, data collection and labelling for domain-specific tasks are expensive and time-consuming, resulting in limited training data and/or imbalanced datasets. Previous studies have successfully employed various DL architectures to increase model accuracies for detecting façade defects but were observed to be limited due to imbalanced and/or small datasets. The aim of this study is to mitigate the problem of data scarcity by deploying various combinations of data augmentation techniques and evaluating the accuracy of models developed using the augmented data produced by these techniques. We applied transfer learning using Mask R-CNN and incorporated two novel data augmentation approaches (CutMix and MixUp) along with traditional techniques such as geometric transformations. The accuracies of models in multi-defect detection are evaluated.

1. Introduction

The problem of falling debris from building façades in densely populated cities is a serious issue that has resulted in deaths (most recent happening in 2019 in NYC) and injuries, despite the implementation of compulsory facade inspection programs (Otterman and Haag, 2019). Frequent complaints about debris falls from façades is another indicator of unsafe façades (e.g., the Department of Buildings (DOB) has received over 1,800 citizen complaints annually regarding façade safety in the last decade)(DOB, 2022). The current traditional approach to inspections is plagued with limitations, including unsafe working conditions and inconsistent results. The urgent need to improve facade inspections calls for the exploration and implementation of safer and more accurate autonomous inspection options, such as the use of drones, robots, and deep learning techniques.

Autonomous defect inspection enables a more robust and dependable approach to overcome the limitations of conventional inspection methods. DL techniques can automatically identify façade defects while drones and robots can efficiently and quickly collect vast volumes of data from building façades to be used in these algorithms. However, the main challenge is to ensure that the training of these models includes sufficient coverage of various types of defects, as the frequency of defects varies significantly among different façade materials (e.g., concrete, brick, glass) and defect types (e.g., spalls, cracks). Certain defects (e.g., rolling block) are less frequently observed than others (e.g., cracks), leading to bottlenecks in the DL-based learning process and exacerbating data scarcity, resulting in overfitting and bias, particularly for underrepresented defect types. Previous research has primarily focused on using various DL models to learn from limited data samples to alleviate these issues at the algorithm level. Yet, the performance of these models is often limited by the size of the initial dataset used for training. This is because these models rely on their ability to generalize to new samples based on a limited number of labelled samples, which is highly dependent on the diversity and size of the initial training dataset. As a result, if the initial dataset is too small or lacks diversity, the model may not generalize well to new samples or classes, ultimately leading to poor performance and accuracy. To ensure robust and generalizable models, it is crucial to train them on a wide range of scenarios and to expand the training dataset to increase its variety and size.

Data augmentation and transfer learning are two techniques commonly used in the machine learning domain to address data scarcity problems and improve model accuracy and performance. Data augmentation involves applying various transformations to existing data samples to increase the dataset's diversity and size. This helps balance the sample distribution across different classes. Transfer learning involves adapting a pre-trained model on a large and diverse dataset to solve a related problem with a smaller dataset. This approach leverages the learned features and weights from the pre-trained model, enhancing the model's ability to learn from limited data and addressing data scarcity. This research aims to determine the optimal combination of data augmentation methods to address data imbalance issues in accurately detecting façade defects. To achieve this, the study uses Mask R-CNN transfer learning in combination with two new data augmentation methods (i.e., CutMix and MixUp) and traditional geometric transformations (i.e., random rotation and flipping), with various configurations.

2. Background

2.1 Earlier studies on handling data scarcity

Deep learning models have been increasingly utilized in façade inspections to automate the detection of a variety of defects, including but not limited to cracks, spalling, efflorescence, delamination, peeling, and blistering. These defects are primarily identified through two main approaches: object detection and semantic segmentation. Object detection techniques involve localizing the defects within an image by drawing bounding boxes around them, while semantic segmentation goes a step further by labelling each pixel in the image with the corresponding defect type, providing a more comprehensive understanding of the defect characteristics such as area of effect and shape. Both approaches have been applied in the context of convolutional neural networks (CNNs), which excel at automatically extracting relevant features for defect classification. While these studies have demonstrated success in distinguishing multiple defect classes and improving the efficiency of façade inspections, there is still room for enhancement in model performance.

To address data scarcity, researchers have proposed solutions that can be classified into two categories: data-level and algorithm-level solutions. Algorithm-level solutions improve deep learning performance with limited data through techniques like few-shot classification(Cui et al., 2022), meta-learning(Guo et al., 2020), semi-supervised learning(Guo et al., 2021), and transfer learning(Wang et al., 2022). These methods target underrepresented classes and use uncertainty filters to enhance model accuracy. However, these solutions are mainly used for image classification rather than detection and segmentation tasks. Transfer learning repurposes pre-trained models for new tasks and can be combined with data augmentation for detection and segmentation tasks. Data-level solutions improve a model's learning by expanding data size, variety, and representation. Some prevalent data-level techniques include data synthesis, data resampling, and data augmentation. Synthesis uses GANs to create realistic synthetic data. Resampling adjusts data distribution by oversampling/undersampling under/overrepresented classes, while augmentation transforms existing data to increase diversity and size using vanilla augmentations like geometric transformations (e.g., flipping, cropping, translating, colour change, etc.) MixUp (Zhang et al., 2017) and CutMix (Yun et al., 2019) augmentations have been successful in enhancing the performance of DL algorithms and addressing data scarcity issues, gaining popularity in computer science. However, the efficacy of these approaches in increasing the performance of detecting façade defects with defect characterization (i.e., through segmentation models), is yet to be explored.

2.2 Data Augmentation Techniques in a Nutshell

Data augmentation techniques aim to increase the size of training datasets through methods such as data warping or oversampling. Simple transformations like cropping and flipping were initially successful examples of data augmentation(Shorten and Khoshgoftaar, 2019). Data warping involves preserving the associated label while transforming an image using techniques like geometric and color transformations or random erasing. Oversampling methods increase the representation of underrepresented classes in a dataset. They generate artificial images and labels by either creating new samples from scratch (using generative adversarial networks) that resemble real images or by mixing existing samples (using methods such as CutMix and MixUp), which combine two or more images to create a new, synthetic image. This study evaluates the effectiveness of data augmentation methods, including geometric transformations and image mixing, for defect detection segmentation. Based on our knowledge, this is the first study in the automated facade defect detection domain that uses image mixing methods for data augmentation to increase the performance of a deep learning algorithm.

Geometric Transformations. These augmentation techniques involve making changes to the geometric attributes of images. They are typically simple to implement and have great potential to improve model performance. In this study, we explored two traditional geometric transformation methods: horizontal flipping and random rotation. Horizontal flipping flips an image horizontally along both rows and columns, while random rotation includes rotating an image randomly by a specified angle between 0° and 360°. The centre of rotation can be defined manually, but it is usually set to the centre of an image.

Mixing images. This technique merges different images to create inter-class (i.e., a synthetic image generated by mixing two images) samples, enhancing the model's generalization and ability to handle imbalance issues. In this paper, we used two different mixing image strategies: MixUp and CutMix augmentation. Data augmentation techniques such as MixUp and CutMix generate new images by combining random pairs of training images and their corresponding labels. MixUp involves linearly interpolating the pixel values and labels of two images to create a new image. In CutMix, patches of one image are cut and pasted onto another, with the corresponding label being proportional to the size of the patch. Figure 1 illustrates both augmentation techniques.

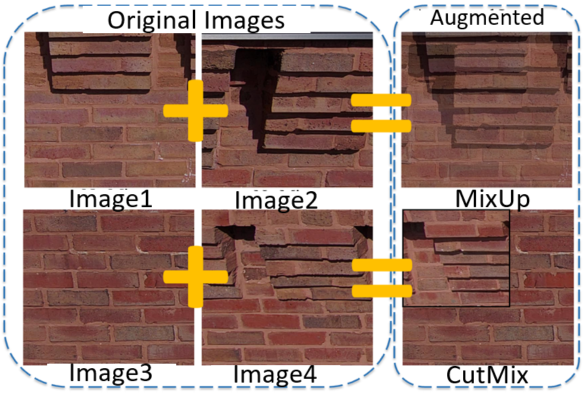

Figure 1: MixUp and CutMix augmentations

2.3 Figures Mask R-CNN Model Architecture and Transfer Learning in a Nutshell

Mask Region-based Convolutional Neural Network (Mask R-CNN) is a two-staged, state-of-the-art deep learning model designed for object detection and instance segmentation tasks. (He et al., 2017). The overview of the model's architecture is shown in Figure 2 (He et al., 2017). The model has two main stages. In the first stage, the model generates proposals for potential object regions (for example rectangular bounding boxes enclosing areas of a façade that have signs of damage) from the input image, and in the second stage, it predicts the class of objects, defines bounding boxes, and generates pixel-level masks based on the generated proposals.

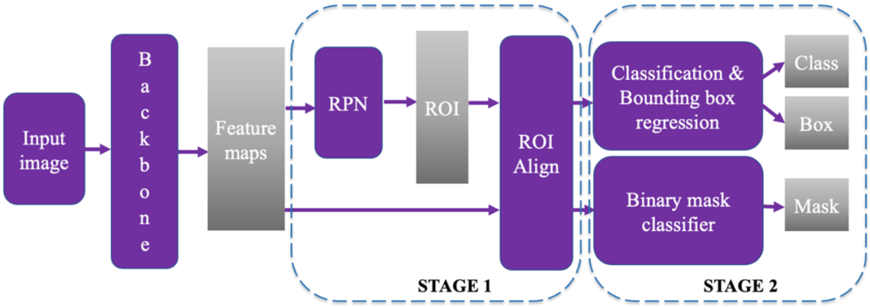

Figure 2: Mask R-CNN model architecture *(source: He et al., 2017)*

In computer vision, a pre-trained convolutional neural network (CNN) is typically used as a backbone architecture during training to generate feature maps from an input image. Feature maps are representations (such as edges, corners, textures, or more complex patterns that are specific to an object) capturing patterns in the image that help the model understand and identify objects. The Region Proposal Network (RPN) then utilizes these feature maps to produce a collection of object-bounding boxes or "Region of Interests" (RoIs), indicating potential areas of the image containing an object. The RoIs are passed through the RoI Align layer to fix any spatial misalignment between extracted features and the input image. Then, the model adjusts bounding box coordinates and assigns an object category and binary mask classifier-branches, which generate pixel-level segmentation masks for each object simultaneously (He et al., 2017). In transfer learning, a pre-trained backbone trained on a large dataset is used to extract feature maps. The RPN, classification and bounding box regression, and binary mask classifier-branches are added based on our dataset and trained from scratch on our custom dataset.

3. Methods

The objective of this study is to enhance the accuracy of a model capable of executing instance segmentation on images of building façades. The model is designed to detect and classify two types of defects: erosion and efflorescence. Erosion is characterized by small cavities on the surface of the facade due to deterioration over time, while efflorescence occurs when salts dissolved by water become visible as white substances on the porous façade material such as brick. Examples are provided in Figure 3. To achieve our goal, we combined data augmentation techniques, which are geometric transformations (horizontal flips and random rotations) and image mixing methods (MixUp and CutMix) with a pre-trained Mask R-CNN model architecture using the ResNet-50 FPN backbone.

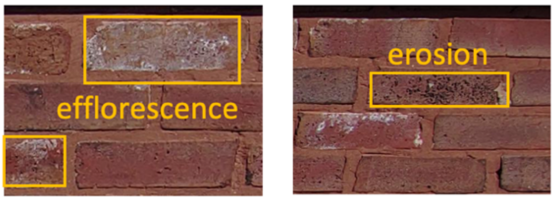

Figure 3: Images showing efflorescence and erosion defects

Research method is composed of a comprehensive three-staged workflow: i) Dataset preparation for training, ii) Configuring hyperparameters, implementing augmentations and trainings and iii) Evaluation of model performances trained with the augmentation methods. First, we pre-processed our façade dataset by resizing, cleaning, and annotating to ensure optimal model performance. Second, we used augmentation methods, including flips, random rotations, and image mixing (i.e., MixUp and CutMix) to increase the diversity of defects in the dataset and improve the model's generalization and learning capabilities. We trained our model with different combinations of augmentation methods and fine-tuned hyperparameters for optimal performance. We evaluated the model's effectiveness by analyzing performance metrics (e.g., average precision and recall) on validation dataset and comparing it with alternative approaches after each training to identify the optimal augmentation configuration that fits to this problem domain.

3.1 Dataset Preparation for the Training

A dataset consisting of 228 raw images of building façades with 4,056 x 3,040 resolution was captured via an unmanned aerial vehicle. To enhance the diversity of our dataset, an algorithm similar to a sliding window of size 512x512 with 20% overlap was employed to extract multiple views from the same image. Subsequently, we eliminated any images that did not include façade portions or contained unwanted elements such as shadows. We further disregarded the images that did not contain erosion and efflorescence resulting in a total of 496 cropped images. After pre-processing the images, we annotated them using Computer Vision Annotation Tool (CVAT) to generate segmentation masks for our deep-learning model training. We registered our custom dataset, which consists of images and their corresponding instance and label segmentation masks. Data registration is an essential step when working with deep learning libraries, enabling to convert the data into a format compatible with chosen framework. This process ensures that the dataset is formatted correctly, and that the library can access and process the data during training and evaluation. Figure 4 gives and overview of the process for preparing the dataset with an example of the cropping process of an original high-resolution image with sliding window from (a) a raw dataset, (b) 512x512 cropped images, (c) its corresponding segmentation (i.e., object class boundaries) and instance (i.e., individual object instances) masks after annotations, and (d) the corresponding registered image, representing the combined image with overlaid segmentation and instance masks for clear visualization of object classes and individual instances in the dataset.

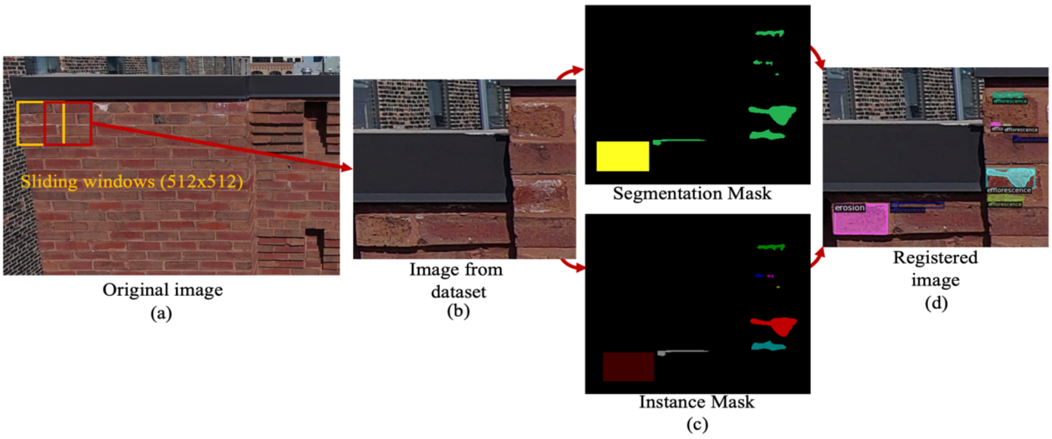

Figure 4: The process of preparing the training dataset. Each colour in the segmentation mask represents a different defect class, each colour in instance mask represents a unique defect instance.

3.2 Finetuning, Configuring Hyperparameters, Augmentations and Training

We used ResNet-50 FPN pre-trained on COCO dataset as the backbone for our Mask R-CNN, where ResNet-50 FPN is a pre-trained CNN that has been shown to be effective for a wide range of computer vision problems. We applied transfer learning and fine-tuned the base model on our custom dataset. In this process, we trained the model to identify the specific classes of erosion and efflorescence, by adjusting the pre-existing weights of the pre-trained model and setting the number of output classes to two, representing defects. Overall, this approach enabled the model to accurately detect and segment erosion and efflorescence and accelerate the learning process while using datasets generated by a combination of data augmentation techniques. We fine-tuned the model's hyperparameters, including the number of workers, learning rate, and maximum iterations, to optimize its performance. The number of workers refers to the parallel data loading processes, the learning rate determines the step size of the optimization algorithm during training, and the number of iterations is the total times the algorithm processes the data and updates the model weights. We set the number of workers as 2 (default recommended) and the number of classes to 2 classes (erosion and efflorescence).

3.3 Evaluation of Model Performances Trained with Augmentation Methods

Upon completing each training iteration, we assessed the performance of our trained models on the validation set with never used images. We employed the COCO Evaluator, which is a widely used evaluation tool for assessing object detection and instance segmentation models. During this stage, we thoroughly examined the overall effectiveness of the different augmentation techniques implemented. The performances of the trained models were assessed on the validation dataset by comparing the predicted objects with the ground-truth objects in terms of detection and segmentation. In this study, metrics such as intersection over union (IoU), precision, recall, and average precision (AP) were employed to gauge the effectiveness of instance segmentation and object detection. The overall process of calculating AP and AR remains the same for both tasks; the only difference occurs in the way of calculating IoU. AP is a widely used evaluation metric in object detection and instance segmentation tasks because it considers both precision and recall. It measures the area under the precision-recall curve and provides a single number that summarizes the overall performance of a model. The higher the AP value, the better the model's performance.

IoU is a metric used to evaluate model performance in object detection and segmentation. In object detection, it measures the overlap between the predicted and ground-truth bounding boxes, while in segmentation, it measures the overlap between the predicted and ground-truth masks. When the prediction is correct, and the IoU value is higher than the given threshold, the prediction is considered a true positive (TP); when IoU is below the threshold, it is deemed a false positive (FP). The undetected ground truth objects are considered to be False Negative (FN). Using the TP, FP, and FN values, we calculated the precision and recall as follows:

$$\text{Precision} = TP/(TP + FP) \tag{1}$$

$$\text{Recall} = TP/(TP + FN) \tag{2}$$

AP has been calculated for different IoU thresholds and object sizes. AP50 and AP75 are average precision metric, calculated for the corresponding IoU thresholds, set to 0.50 and 0.75, respectively. When the IoU threshold is set to 0.50, it means that a predicted bounding box/segmentation mask is considered correct if it overlaps with the ground truth annotation by at least 50%. Similarly, when the IoU threshold is set to 0.75, it means that a predicted bounding box/segmentation mask is considered correct only if it overlaps with the ground truth annotation by at least 75%. AP values for different object dimensions offer an understanding of the model's performance across a range of defect sizes, including small (area < 32x32 pixels), medium (32x32 pixel<= area <= 96x 96 pixels), and large (area > 96x96 pixels). In this study, AP-small (AP_{small}) mainly attributes to performance for small-sized efflorescence instances due to their relatively small pixel area.

Average Recall (AR) is calculated at a fixed number of maximum detections (e.g., 1, 10, or 100) across different IoU thresholds and object sizes, and it is associated with the model's performance in identifying true positive instances. We also calculate the AP values for specific classes to assess how well the model is across different defects, so AP-efflorescence and AP-erosion are calculated only considering instances belonging to the target classes.

4. Experimental Design

4.1 Initial Dataset

We used a UAV captured 228 high-resolution (i.e., 4,056x3,040) raw façade images of a brick building. Multiple perspectives of the same image were generated using a 512x512 sliding window with a 20% overlap. After pre-processing, we annotated 526 images with 512x512 resolution, identifying 5,203 instances of erosion and 2,585 instances of efflorescence.

4.2 Model Hyperparameters and Augmentation Technique Configurations

For training and evaluation of models, we utilized Detectron2, a popular library for object detection and segmentation that is built on PyTorch (an open-source Python machine-learning framework for generating and training deep learning models.) We employed the Mask R-CNN architecture with the ResNet-50 FPN backbone as our base model.

For training configurations, we used a configuration file from Detectron2's for the Mask R-CNN with ResNet-50 FPN architecture and specified our training and validation datasets. We experimented with several base learning rates (0.0001 and 0.0002) and set different numbers of iterations (500-1000-1500-300-5000). We also tested different batch sizes, 128 and 512, while we left the number of workers as the default value of two. Finally, we set the number of output classes to two since we only dealt with two types of defects. We incorporated data augmentation techniques to improve the model’s generalization, such as random horizontal flips and random rotations between 0° and 360°. We defined a custom augmentation pipeline and added it to our training configuration. We considered two extra augmentations, MixUp and CutMix and defined four augmentation combinations as follows:

- G: Geometric Transformations (horizontal flipping, random rotation)
- G-Mix: Geometric Transformations + MixUp
- G-Cut: Geometric Transformations + CutMix
- All: Geometric Transformations + MixUp + CutMix

For the G-Mix and G-Cut configurations, we performed MixUp or CutMix augmentations in 20% of the images, while the remaining 80% of the images did not have either of these augmentations. In the 'All' combination, 20% of the images had mixing augmentations applied. Within this 20%, half of the images were augmented with MixUp, and the other with CutMix. Additionally, we used recommended alpha values of 0.1 for MixUp and 1.0 for CutMix.

5. Results and Discussion

We presented our multi-defect detection model's performance in two tables, highlighting its segmentation capabilities. Table 1 provides results on the model's proficiency in segmenting defects, using metrics defined above as the percent *AP50, AP75,* AP_{small}, AP_{medium}, and AP_{large}. These metrics help us gauge the model's proficiency in segmenting defects concerning IoU thresholds (at 50 and 75), as well as its performance on the small, medium, and large regions examined by models in defect segmentation. Table 2 combines both metrics and includes models’ performance on segmentation per defect type: efflorescence and erosion. This table features *AP*, representing the overall segmentation performance, along with metrics for segmentation performance on specific defect types: $AP_{efflorescence}$, and $AP_{erosion}$ (in %). By analysing these metrics, we can determine suitable augmentation combinations that enhance overall performance of models.

The results in Table 1 show that the G-Mix augmentation combination provides the best performance at the 50% IoU threshold, while the G-Cut combination excels at the 75% IoU threshold, with a relatively lower performance then AP50 scores, but closer to G-Mix performance at 75% threshold. The 'All' combination offers competitive performance across both IoU thresholds and object sizes, especially in medium objects. It is worth mentioning that model performance is significantly lower when segmenting defects using small regions in the analysis (i.e., AP_{small}). This requires an explanation, and it only makes sense when Table 2 is analysed per defect type.

Table 1: Segmentation performance of models trained with given augmentation methods for both defects

Augmentation Methods	**AP**	**AP50**	**AP75**	**AP$_{small}$**	**AP$_{medium}$**	**AP$_{large}$**
G	25.01	39.41	25.96	**3.71**	27.74	27.04
G-Mix	**25.12**	**40.57**	25.55	**4.38**	28.26	23.73
G-Cut	24.07	36.6	**26.42**	2.27	**29.21**	23.95
All	24.55	38.61	26.30	2.66	29.09	25.76

Table 2 presents overall segmentation (*AP*) performance of models along with their performance per defect type. The results indicate that the G augmentation provides the highest AP for segmentation. AP results for small, medium, and large regional analysis showed in Table 1 that a closer look is needed for understanding the low AP_{small} results. For this purpose, we looked at model segmentation performance per defect. Table 2 shows that the *AP* scores drop because of the efflorescence defects where the highest performance for it is via G-Mix (i.e., $AP_{\text{efflorescence}}$ =9.32%). Regarding erosion defects, the models are much better in segmenting this defect type, where the models trained with 'All' data augmentation combination get the best segmentation performance ($AP_{erosion}$ = 43.13%).

Table 2: Overall segmentation performance of models trained with given augmentation methods across defects and specific to each defect type.

Augmentation Methods	**AP**	**$AP_{\text{efflorescence}}$**	**AP_{erosion}**
G	25.01	8.00	42.03
G-Mix	25.12	**9.32**	40.93
G-Cut	24.07	5.23	42.91
All	24.55	5.99	**43.13**

The notable differences in segmentation performance of models for efflorescence and erosion, particularly on AP_{small} can be attributed to several factors, such as the characteristics of the defects (e.g., size and aspect ratio) and the imbalanced dataset. Efflorescence defects are in general regional, spanning over a façade cutting through several brick surfaces and should be labelled as such. When labelling is done at brick surface level and smaller group of pixels affected by efflorescence are observed per brick surface, models struggle to detect them per brick surface. This discussion indicates that the labelling effort should be changed to label efflorescence at the façade surface level instead of at brick level. This will be addressed in the future work of this effort. Regardless though, models trained with data augmented using G-Mix method perform better in detecting efflorescence as compared to the other combinations.

In contrast, erosion defects are widely spread at brick surfaces occupy a more significant portion of the image pixel-wise, enabling more information for the model to learn and make accurate predictions. Aligning with this, our model provided better performance in large objects. This is reflected in the higher AP values for erosion compared to efflorescence, as well as the AP_{large} results, which represent the model's performance on larger defects like erosion Additionally, the dataset is imbalanced, due to a higher number of large-sized erosion and a lower number of smaller-sized efflorescence, where the efflorescence instances typically have a smaller pixel area compared to erosion. This imbalance might cause the model to be more biased towards segmenting larger defects (such as erosion), resulting in higher AP values for erosion and better performance in AP_{large} than AP_{small}.

As a result of this study, it is apparent that data augmentation methods significantly effect segmentation capabilities of models trained with datasets generated from these methods. Our evaluations in defect detection on façade surfaces showed that G-Mix method provides meaningful new images that help boost model performance in segmentation. However, selecting the most suitable augmentation combination depends on the particular needs of the application, such as on overall performance, or the importance of accurately segmenting specific defect types like efflorescence or erosion. Regardless, in situations where addressing the imbalanced dataset is crucial, exploring different augmentation strategies or techniques tailored for handling imbalanced data could lead to improved performance for the underrepresented class.

6. Conclusion

This study demonstrated the potential of combining data augmentation techniques and transfer learning with a pre-trained Mask R-CNN model for accurate façade segmentation, specifically for erosion and efflorescence defects. The integration of geometric transformations (e.g., horizontal flips and random rotations) and image mixing methods (MixUp and CutMix) contributed to an improved model performance by addressing data scarcity and increasing the diversity of the training dataset. The evaluation of different augmentation combinations showed that models trained with G-Mix method based augmented datasets performed better as compared to other methods generally, with variations observed when performances are compared for each specific defect type. As an ongoing work, we will explore defining a labelling strategy that fits the characteristics of each defect analysed and explore additional augmentation techniques and the incorporation of algorithm-level solutions to further improve the model's performance and generalizability across a broader range of façade materials and defect types.

Acknowledgments

The authors would like to thank Savani Gokhale and Keundeok Park for their assistance in data training. We would like to extend our appreciation to Janice Su, Shakya Amaratunga, Juntao Liu, and Anwar Benmallouk, who assisted with data annotation.

References

Cui, Z., Wang, Q., Guo, J., & Lu, N. (2022). Few-shot classification of façade defects based on extensible classifier and contrastive learning. Automation in Construction, 141, p.102381. https://doi.org/10.1016/j.autcon.2022.104381DOB. (2022).

DOB Complaints Received | NYC Open Data. https://data.cityofnewyork.us/Housing-Development/DOB-Complaints-Received/eabe-havv

Guo, J., Wang, Q., & Li, Y. (2021). Semi-supervised learning based on convolutional neural network and uncertainty filter for façade defects classification. Computer-Aided Civil and Infrastructure Engineering, 36(3), pp.302–317. https://doi.org/10.1111/mice.12632

Guo, J., Wang, Q., Li, Y., & Liu, P. (2020). Façade defects classification from imbalanced dataset using meta learning-based convolutional neural network. Computer-Aided Civil and Infrastructure Engineering, 35(12), pp.1403–1418. https://doi.org/10.1111/mice.12578

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision, pp.2961-2969. http://arxiv.org/abs/1703.06870

Otterman, Sharon; Haag, M. (2019). Woman Killed by Falling Debris Near Times Square - The New York Times. The New York Times. https://www.nytimes.com/2019/12/17/nyregion/woman-killed-times-square.html

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1), pp.1-48. https://doi.org/10.1186/s40537-019-0197-0

Wang, H., Li, M., & Wan, Z. (2022). Rail surface defect detection based on improved Mask R-CNN. Computers and Electrical Engineering, 102, p.108269.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 6023-6032. http://arxiv.org/abs/1905.04899