Maastricht University

Department of Advanced Computer Science

Data Science for Decision Making

Detect Damaged Buildings
The Importance of Unet's Architecture

Author:	*Student Number:*
Christos Koromilas	i6345703

April 23, 2024

Contents

1 Introduction

In the realm of disaster management, the ability to swiftly and accurately assess damage is crucial for effective response and aid deployment. This report delves into a groundbreaking approach that leverages deep learning, applying it to the analysis of satellite imagery for assessing building damage across a spectrum of disasters – from natural calamities like earthquakes, tornadoes, volcanoes, and tsunamis, to human-made crises such as war.

The versatility of this model lies in its training on diverse disaster scenarios. By learning from the varied patterns of destruction caused by natural disasters, the model is designed to develop a robust understanding of damage in multiple contexts. This multifaceted training approach aims to enhance the model's accuracy and reliability, regardless of the disaster type.

The central ambition of this project is not just to innovate in the field of damage assessment but to pivot this technology towards real-world applications in war-torn areas. While the initial focus is on natural disasters, the ultimate test of the model's efficacy will be its application to satellite images from war zones. Here, the challenge is unique: the model must adapt its learning from natural disasters to the often more complex and varied damage patterns seen in conflict areas.

This endeavor is more than a technological feat; it's a humanitarian effort aimed at providing rapid, accurate insights into damaged areas. These insights are vital for directing aid and resources effectively, especially in regions where on-the-ground assessment is hindered by ongoing conflict. By bridging the gap between technological advancement and humanitarian needs, this project seeks to offer a powerful tool in the hands of organizations and governments striving to mitigate the impact of disasters and conflicts on vulnerable populations.

2 Previous Experience in DL/CV

2.1 Courses

- **Computer Vision Course**
  - Implemented a *Face Emotion Recognition* project where a deep learning model was trained to identify and classify different human emotions from facial expressions in images and videos.
  - Developed an *Image Stitching* algorithm to combine multiple overlapping images into a high-resolution panoramic image, handling issues of transformation and alignment.
- **Signal and Image Processing Course**
  - Gained hands-on experience in fundamental *Image Processing* techniques, including image filtering, transformation, and analysis for various applications.

2.2 Research Group Project

- A cutting-edge research project this semester that focusing on *Deep Learning Models for Predicting Odors of Molecules.* Involved in the development and training of deep neural networks to understand and predict the relationship between molecular structures and their perceived odors.

3 Dataset

This project commenced with the acquisition of the Xview2 dataset from its official website.([1]) This dataset encompasses a rich collection of pre- and post-disaster images across various disasters, years, and locations. Accompanying JSON files provide extensive metadata, including building polygons, weather conditions, year, place, and disaster type.

4 State of the Art

The core objective of this project is to devise a model capable of classifying the extent of damage to buildings by utilizing pre and post-disaster images alongside corresponding masks. Given the nature of the dataset at hand, a segmentation-centric approach emerges as the most apt strategy. This involves deploying segmentation models followed by the training of a Siamese network model that operates on pre and post-disaster data. This two-fold methodology is instrumental in identifying buildings and subsequently, assessing the level of damage inflicted.

4.1 UNet Architecture

A pivotal component in the segmentation phase is the UNet architecture. Originally conceived for biomedical image segmentation, UNet has carved a niche for itself in the domain of semantic segmentation of satellite imagery. Its architecture is adept at processing both local and global contexts, a feature that is paramount in achieving precise segmentation—a critical requirement for interpreting satellite data with accuracy ([2]).

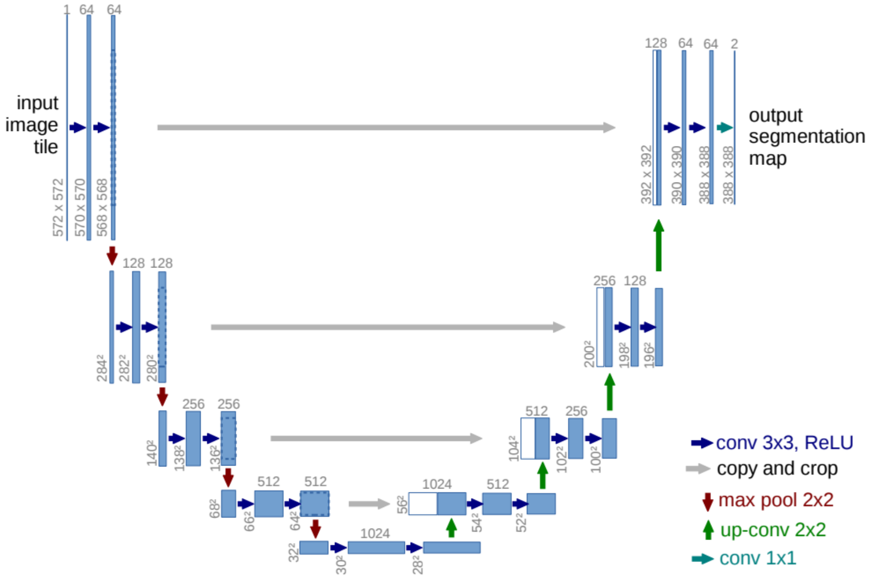


Figure 1: UNet Architecture

4.2 Leveraging Transfer Learning

To augment the performance of our model, we are considering the adoption of transfer learning. This involves utilizing models pre-trained on robust datasets like ImageNet as backbones for our UNet models. Pre-trained models such as ResNet50, ResNeXt50, and SE-ResNeXt50 are under consideration. Transfer learning has gained widespread acceptance for its capacity to mitigate the constraints posed by limited datasets, thereby enhancing the robustness and versatility of models tasked with satellite imagery segmentation ([3]).

4.3 ResNet50

ResNet50 marks a milestone in deep neural network design, noted for its ingenious use of residual connections that enhance both efficiency and effectiveness. This architecture has garnered acclaim for its superior performance in capturing intricate details within high-resolution satellite imagery, showcasing its versatility across various image recognition tasks [4]. At the heart of ResNet's innovation is the concept of learning residual functions related to the inputs of layers, rather than learning unreferenced functions. This approach is operationalized through stacking residual blocks, which fosters learning of residual mappings and allows for constructing deep networks without the risk of performance degradation. ResNet-50, in particular, embodies this philosophy in a 50-layer deep structure, leveraging these residual blocks to enable profound learning capabilities.

4.4 ResNeXt-50 (32x4d)

ResNeXt-50 (32x4d) introduces an architectural paradigm that adeptly balances depth with breadth, a design particularly suited for the complex nature of satellite imagery analysis. The cornerstone of ResNeXt lies in its use of grouped convolutions, which significantly amplifies its capacity for feature representation and positions it as a formidable tool for image analysis tasks [5]. What sets ResNeXt-50 (32x4d) apart is its modular building block design, which harmonizes a set of transformations sharing the same topology. Distinguished from ResNet by the introduction of a new dimension known as ”cardinality” – the number of unique transformations – ResNeXt integrates this parameter with depth and width, thereby enriching the model's representational power.

4.5 SE-ResNeXt-50 (32x4d)

SE-ResNeXt-50 (32x4d) emerges as an exemplary model by amalgamating the robustness of ResNeXt with the innovative concept of 'Squeeze-and-Excitation' (SE) blocks. This synergy refines the model's capacity to concentrate on pivotal features within satellite imagery, striking a balance between depth and width in its 32x4d configuration to adeptly capture the nuanced local and global patterns in the data [6]. This variant extends the ResNeXt framework by integrating SE blocks, which dynamically recalibrate channel-wise features, thereby imbuing the network with enhanced adaptability and responsiveness. The SE-ResNeXt model, with its advanced recalibration mechanism, stands at the forefront of deep learning architectures, setting new benchmarks in model performance and interpretability.

4.6 Siamese Networks: A Dual Approach

In the realm of temporal change detection within satellite imagery, Siamese Networks have come to the fore. These networks, characterized by parallel Unets with shared weights, are particularly beneficial in scenarios involving environmental monitoring and urban development, showcasing their versatility and effectiveness ([7]).

4.7 Specialized Models: BDAnet and Xview2 Challenge Winner

Models tailored for specific use-cases, such as BDAnet and those triumphing in the Xview2 challenge, underscore the specialized capabilities that can be harnessed for Building Damage

Assessment. These models underscore the transformative potential of satellite imagery in delivering critical insights, especially pertinent in post-disaster evaluations ([8, 9]).

5 Research Questions

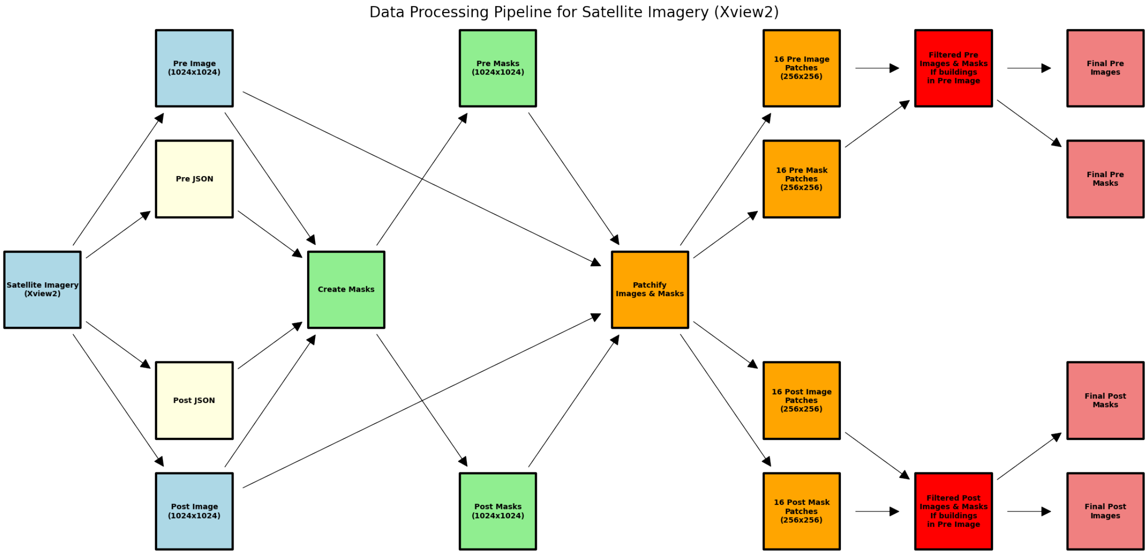
In this section, we aim to present the research questions:

- How the different backbones affect the performance of the segmentation model Unet?
- What if we use the backbones that are pretrained on imagenet with frozen and not frozen weights?
- How to explain the models predictions?
- Is the best model able to detect as a siamese network with shared weights, the changes on images?

6 Implementation and Approach

6.1 Data Preparation

To facilitate training segmentation models, masks were extracted from each JSON file corresponding to the images. The dataset includes both pre- and post-disaster images, with post-disaster images annotated with building damage levels ranging from 1 to 3 (0: background) and pre-disaster images with the masks of buildings. For that, I used part of the code of Xview2 winner to extract the polygons [10]. The dataset images are of size 1024x1024 pixels. To enhance processing efficiency, images were subdivided into smaller patches of 256x256 pixels, resulting in 16 smaller images per original image. I used the library patchify [11]. Additionally, images devoid of buildings were identified and excluded to balance the dataset.


Data Processing Pipeline for Satellite Imagery (Xview2)
Satellite Imagery (Xview2)
Pre Image (1024x1024)
Pre JSON
Post JSON
Post Image (1024x1024)
Create Masks
Pre Masks (1024x1024)
Post Masks (1024x1024)
Patchify Images & Masks
16 Pre Image Patches (256x256)
16 Pre Mask Patches (256x256)
16 Post Image Patches (256x256)
16 Post Mask Patches (256x256)
Filtered Pre Images & Masks If buildings in Pre Image
Filtered Post Images & Masks If buildings in Pre Image
Final Pre Images
Final Pre Masks
Final Post Masks
Final Post Images

For the segmentation task, only pre-disaster images were employed. Utilizing Google Colab for computational resources, images were stored in zip files for expedited loading and handling.

The dataset, comprising approximately 110,000 pre-disaster images of size 256x256, was prepared using the splitfolders library for systematic file organization [12] in order to split the data into train, validation and test.

6.2 Data Augmentation

To enhance the model's generalization capabilities and robustness against overfitting, extensive data augmentations were applied during the training phase. These augmentations simulate various conditions the model might encounter in real-world scenarios and include geometrical transformations and color space augmentations to provide a diverse set of training samples. The Albumentations library [13] was utilized for its comprehensive set of augmentation techniques and efficient processing. The augmentations included horizontal and vertical flips, random rotations, color jittering for brightness, contrast, saturation, and hue adjustments, Gaussian blur, elastic transformations, and random grid shuffling. These augmentations were carefully chosen to ensure they are meaningful for the domain-specific features of the dataset, thus preparing the model for a wide range of visual variations.

6.3 Model Development

The models developed in this project are primarily based on the U-Net architecture, utilizing different backbones: ResNet50, ResNeXt50 (32x4d), and SE-ResNeXt50 (32x4d), sourced from the segmentation_models_pytorch library [14]. These models, built on state-of-the-art principles, underwent various experiments to assess the impact of pretrained backbones. For SE-ResNeXt50 (32x4d), the model was trained using a pretrained (Imagenet) and frozen backbone to allow for comparative analysis.

Model Details:

- Unet with Backbone: ResNet50 params: 23M
  - Imagenet pretrained (frozen weights of backbone)
  - Imagenet pretrained
  - No pretrained
- Unet with Backbone: ResNeXt50 (32x4d) params: 22M
  - Imagenet pretrained (frozen weights of backbone)
  - Imagenet pretrained
  - No pretrained
- Unet with Backbone: SE-ResNeXt50 (32x4d) params: 25M
  - Imagenet pretrained (frozen weights of backbone)

6.4 Training and Validation Parameters

For the training and validation of the model the following parameters and evaluation metrics were used:

- **Batch Size**: Set to 32 to balance computational load and model effectiveness.
- **Worker Processes**: Employed 8 workers for efficient data handling.

- **Optimizer**: Used Adam with a learning rate of 0.001.
- **Learning Rate Scheduler**: Applied ReduceLROnPlateau for dynamic learning rate adjustment based on validation loss.
- **Training Epochs**: Configured to run for 50 epochs.
- **Early Stopping**: Patience of 10 epochs, to prevent overfitting.
- **Gradient Accumulation Steps**: Set to 4, optimizing training on smaller batches.
- **Model Saving Interval**: Scheduled to save model weights every 2 epochs.
- **Loss Function**: Jaccard loss, to measure the similarity between predicted and actual segmentation masks.
- **Metric**: Dice coefficient, assessing the overlap between predicted segmentation and ground truth.

These parameters and metrics were crucial for efficiently training the segmentation models and accurately evaluating their performance. Using the Jaccard Score as a loss function and the Dice Score as a metric provides a comprehensive evaluation from two slightly different perspectives, offering a robust assessment of each model's performance.

6.5 Explainability

To comprehend and validate the decision-making process of the trained models, an explainability approach was incorporated. Saliency maps were generated to visually highlight the areas in the images where the model focused most during the prediction. This visualization technique provides insights into the model's behavior, allowing us to understand which parts of the image are deemed significant by the model and contributing to the accuracy of its predictions. It serves as a powerful tool to ensure the model's attention aligns with the relevant features in the image, ultimately building trust in the model's predictions and aiding in the identification and mitigation of any biases present in the training process.

6.6 Challenges in Siamese Network Implementation

Early in this project, I planned to use a Siamese network to improve the assessment of building damage after disasters. However, creating this network turned out to be quite challenging. The main issues were the network's complexity and the need to build it from scratch. I looked at how others have built similar networks, but they often had access to better resources or used special libraries that were hard for me to work with.

Because of these difficulties, I decided to focus more on segmentation models. These models still gave me useful information and results. I had hoped that adding a Siamese network would make my damage assessments better and more detailed, giving me a fuller picture of the situation.

Even though I couldn't implement the Siamese network this time, my work has set a strong foundation for the future. I still think that including a Siamese network in my project could be very beneficial, and I plan to explore this possibility further in the future.

7 Results and Discussion

This section presents the performance evaluation of different models based on their training and validation phases. Loss and Dice score metrics were used to gauge the models' performance over epochs.

7.1 Model Performance Train-Valication

The models' performance was assessed using the Jaccard loss and Dice score during the training and validation phases.

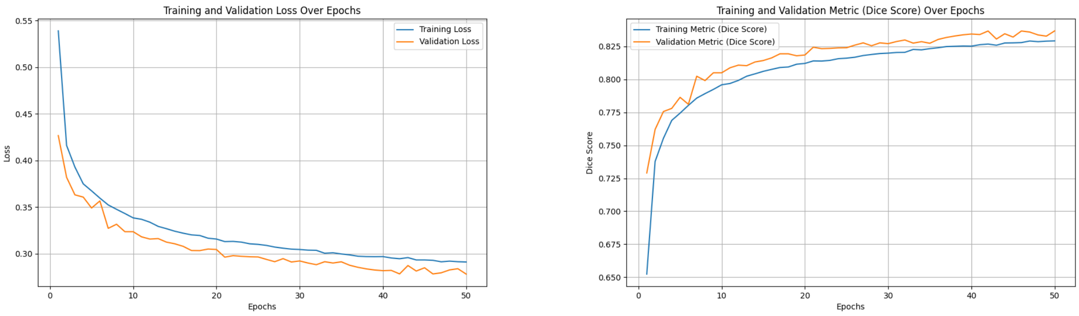


Figure 2: Loss and Dice score curves for Unet with ResNet50 backbone (Imagenet pretrained and frozen).

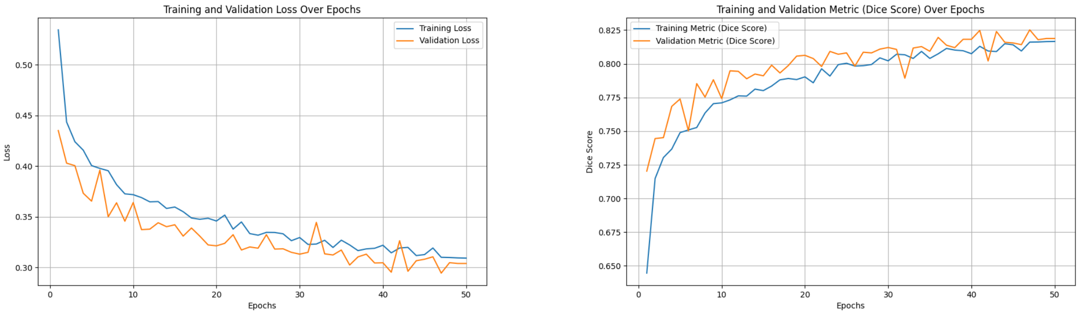


Figure 3: Loss and Dice score curves for Unet with ResNet50 backbone (Imagenet pretrained and finetuned).

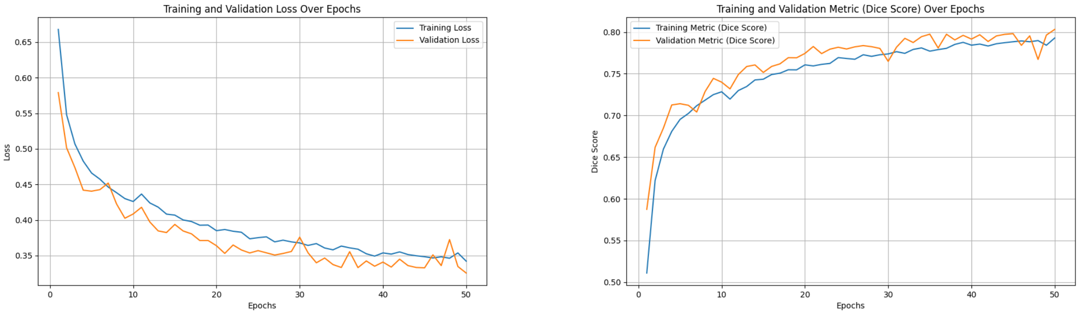


Figure 4: Loss and Dice score curves for Unet with ResNet50 backbone (no pretrained weights).

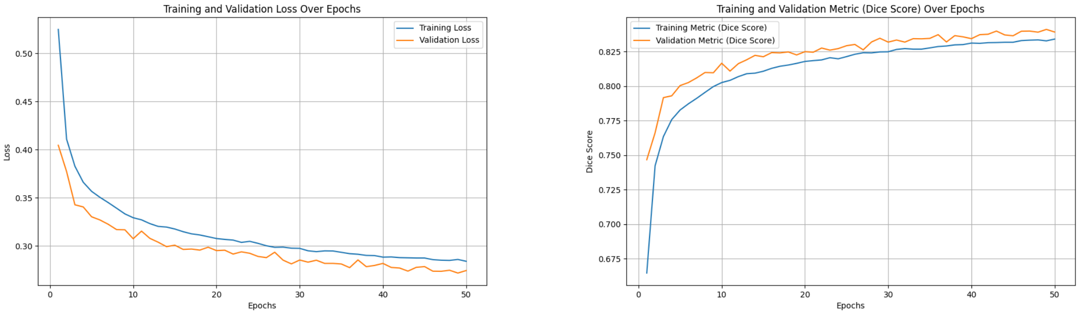


Figure 5: Loss and Dice score curves for Unet with ResNeXt50 (32x4d) backbone (Imagenet pretrained and frozen).

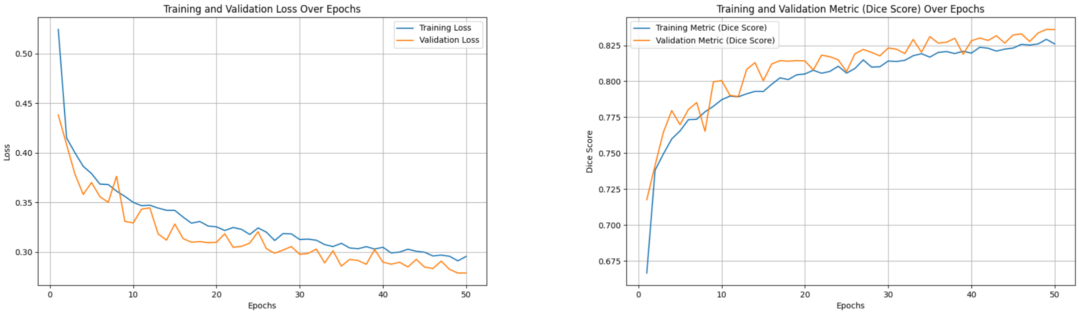


Figure 6: Loss and Dice score curves for Unet with ResNeXt50 (32x4d) backbone (Imagenet pretrained and finetuned).

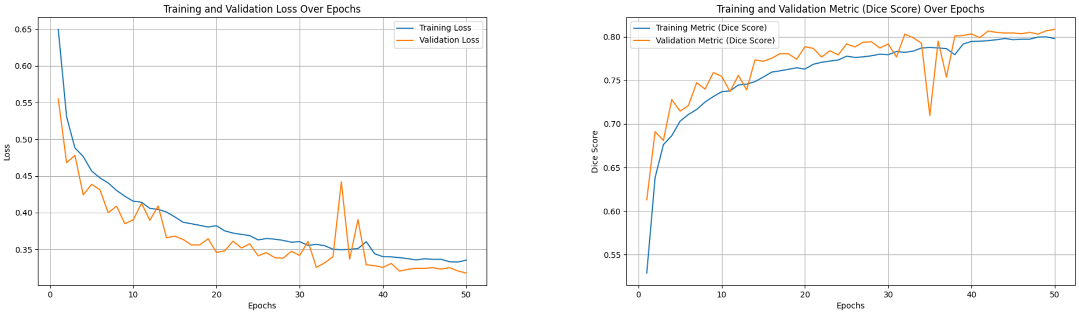


Figure 7: Loss and Dice score curves for Unet with ResNeXt50 (32x4d) backbone (no pretrained weights).

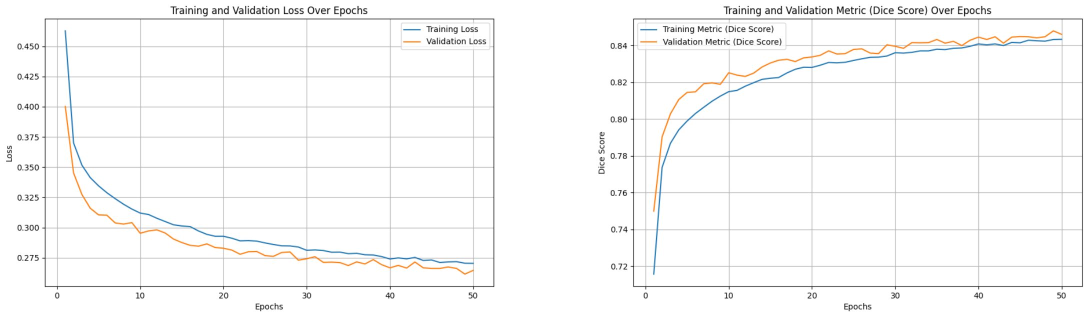


Figure 8: Loss and Dice score curves for Unet with SE-ResNeXt50 (32x4d) backbone (Imagenet pretrained and frozen).

The models that we trained from scratch – without any pretrained weights – took longer to get good and usually had higher loss and lower Dice scores. This makes sense because learning from scratch is tough, especially when the images are as complex as the ones we're using for damage assessment.

We noticed that for all models, the loss went down quickly at first and then slowed down. The Dice score, which tells us how accurate the model is, went up fast early on and then leveled off. This pattern is good because it means the models weren't just memorizing the training images – they were learning rules that help them do well on new images they haven't seen before.

7.2 Model Performance Test

This subsection provides a detailed analysis of the model performance on the test set. The models, differentiated by their architecture, backbone, and training strategy (pretrained weights and freezing status), were evaluated on the test set to compare their effectiveness. The primary metrics used for performance evaluation are accuracy and Intersection over Union (IoU). These metrics provide insights into the models' ability to accurately segment the images.

The performance results are summarized in Table 1, presenting a clear comparison across different model configurations.

Table 1: Performance comparison of different model configurations on the test set.

Architecture	**Backbone**	**Pretrained**	**Accuracy**	**IoU**
UNET	resnet50	Imagenet, not frozen	0.9651	0.8323
UNET	resnet50	Imagenet, frozen	0.9657	0.8384
UNET	resnet50	None	0.9602	0.8158
UNET	resnext50_32x4d	Imagenet, not frozen	0.9661	0.8387
UNET	resnext50_32x4d	Imagenet, frozen	0.9663	0.8406
UNET	resnext50_32x4d	None	0.9603	0.8151
UNET	**se_resnext50_32x4d**	**Imagenet, frozen**	**0.9682**	**0.8470**

The table provides a quick overview, allowing for an immediate comparison between the different configurations. It is evident from the table that the models pre-trained with Imagenet weights tend to perform better in terms of both accuracy and IoU. This trend is particularly noticeable in the model utilizing the SE-ResNeXt50 backbone, which showcases the highest accuracy and IoU scores, underscoring the effectiveness of the pretrained weights coupled with this specific architecture and backbone combination.

7.3 Test Images

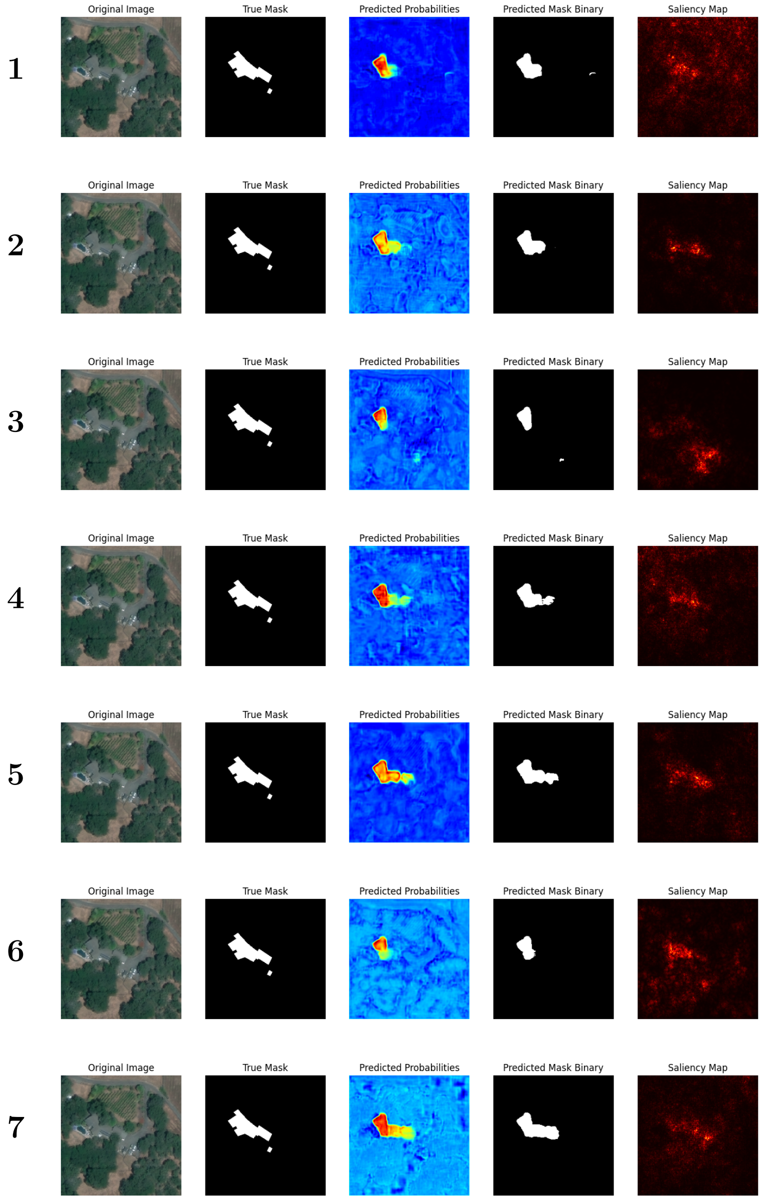


Figure 9: Image processing results for different models:
(1) UNet with ResNet-50 on ImageNet Frozen weights,
(2) UNet with ResNet-50 on ImageNet,
(3) UNet with ResNet-50 no pretrained,
(4) UNet with ResNeXt-50 32x4d on ImageNet Frozen weights,
(5) UNet with ResNeXt-50 32x4d on ImageNet,
(6) UNet with ResNeXt-50 32x4d no pretrained,
(7) UNet with SE-ResNeXt-50 32x4d on ImageNet Frozen weights.

8 Research Question Answers

8.1 RQ1: Different Backbones

We used three types of backbones for our U-Net models: ResNet50, ResNeXt50 (32x4d), and SE-ResNeXt50 (32x4d). The models with SE-ResNeXt50 (32x4d) usually did better, with lower loss and higher Dice scores. This might be because SE-ResNeXt50 (32x4d) is better at picking up complex patterns in the images.

8.2 RQ2: Using Pretrained Weights

Models that started with weights from already trained networks (pretrained weights) learned faster and did better overall. When we allowed these models to tweak these weights even more (finetuning), they often did better than those with frozen weights. This shows that fine-tuning can really help the model get better at the specific task of figuring out damaged areas in images.

8.3 RQ3: The predictions of Models

Best on the saliency maps it is clear that the model SE-ResNeXt50 (32x4d) can predict the buildings better than other models, because the model focusing better on the buildings.

8.4 RQ4: Siamese Network

The implementation of the Siamese network it wasn't feasible because I couldn't get the output of the model after comparing each image. But for this part we suggest of using this model: SE-ResNeXt50 (32x4d) as it gives better results.

9 Conclusions

The results clearly show that models trained on ImageNet first do better. When a model learns from a big, varied set of images like ImageNet, it gets better at applying what it has learned to new pictures.

Keeping the prelearned ImageNet weights frozen during training seems to work even better. This might mean that the features learned from ImageNet are really good for the task and that not changing them helps the model stay good at applying what it has learned to new situations.

When comparing different types of model structures, the SE-ResNeXt-50 32x4d does better than the ResNeXt-50 32x4d, which does better than the ResNet-50. The SE-ResNeXt-50 has special blocks that help it pay more attention to the important parts of the data. These seem to really help the model spot the right patterns.

Saliency maps, which show where a model is focusing, confirm this. They show that the best models pay more attention to the important parts of the image. This is a sign of a model that has been trained well.

To sum up, choosing the right model structure and using pretrained weights can make a big difference in image-related tasks. These choices often lead to better and more efficient results.

The tests also suggest that for specific tasks like spotting changes in images, the SE-ResNeXt50 (32x4d) model with its pretrained weights is a good choice. Future research could try new ideas to improve models further or see if these findings help with other kinds of image work. For tasks that need to compare images, such as identifying damage, this model is recommended because it creates accurate outlines that help in evaluating the changes.

10 Future Work

This project has opened many paths for further exploration. In future work, I plan to dive into advanced Unet structures like Unet++ to possibly improve how the model captures details in images. Trying out stronger backbones such as SE-ResNeXt 101 is also on the list, aiming to get better at picking out important features in the images. Including more datasets, especially those from war-torn areas, will help the model learn from a wider variety of images. Experimenting with different types of image augmentations can make the model more robust, helping it learn from a broader range of scenarios. Another exciting direction is developing a Change Detection model, similar to BDAnet, to spot changes in areas over time. Lastly, I want to make the model's decisions clearer and more understandable by using techniques like LIME, SHAP, and Grad-CAM. This will help ensure the model is not just strong but also reliable and clear in how it makes decisions.

11 Learning Outcomes

This project, taught me a lot about the challenges of working with satellite images. I learned how important it is to understand the specific area you're working on. The project was a great chance to get deeper into PyTorch, a powerful tool for deep learning. I explored different segmentation models, learning how they work and why they're important for making sense of complex images. I also got into Siamese Networks and Change Detection models, seeing how deep learning can track changes in images over time. Understanding Transfer Learning and Fine Tuning showed me how to use existing models for new tasks, saving resources and improving results. Overall, this project was a real hands-on experience in deep learning, teaching me how to tackle real challenges with advanced AI methods. The difficulties I faced and what I learned from them have laid a strong foundation for my future work in this field.

References

[1] "xview2 challenge dataset." `https://xview2.org/`.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, pp. 234–241, Springer, 2015.

[3] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in Neural Information Processing Systems*, 2014.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[5] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

[6] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

[7] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *Advances in Neural Information Processing Systems*, pp. 737–744, 1994.

[8] Y. Xu, D. Lam, and S. Mukhopadhyay, "Bdanet: A general framework for building damage assessment," *IEEE*, 2020.

[9] D. Lam, Y. Xu, and S. Mukhopadhyay, "Xview2 challenge: A deep learning approach," *IEEE*, 2018.

[10] DIUx-xView, "xview2 baseline model and utilities." `https://github.com/DIUx-xView/xView2_baseline`.

[11] Dovahcrow, "Patchify: A simple library for patching and reconstructing images." `https://github.com/dovahcrow/patchify.py`.

[12] J. Filter, "Split folders: Split files of a directory into training, validation, and test datasets." `https://github.com/jfilter/split-folders`.

[13] "Albumentations: Fast and flexible image augmentations." `https://albumentations.ai/`.

[14] "Segmentation models pytorch." `https://segmentation-modelspytorch.readthedocs.io/en/latest/`.