Transformers and CNNs for semantic segmentation of cracks

Sravan Josh Koka
University of Houston
Houston, TX
skoka2@uh.edu

Pooyan Ghodrati
University of Houston
Houston, TX
pghodrati@uh.edu

Sujitha Ravichandran
University of Houston
Houston, TX
sravich8@uh.edu

Rakesh Kumar Alasakani
University of Houston
Houston, TX
ralasaka@uh.edu

Abstract

Semantic segmentation holds a crucial role in assessing the structural integrity of buildings. The emergence of transformer architectures, initially prevalent in natural language processing, has shown promising capabilities in visual tasks. This project delves into the use of both convolutional neural networks (CNNs) and transformer models for semantic segmentation of structural damage, focusing on crack detection in building images. We employed the DeepLabV3 model, a CNN architecture, alongside transformer-based models such as UNetFormer and MaskFormer. Our findings reveal a notable performance in identifying coarse cracks, while segmenting finer damage proved more challenging. This exploration not only demonstrates the adaptability of transformer architectures to visual data but also underscores their potential in refining the accuracy of building inspections.

This endeavor is inspired by key research in image segmentation and damage detection. With the rapid evolution of deep learning, evaluating the practical application of these advanced models in vital areas, particularly in construction and infrastructure upkeep, is essential. Our work aims to add meaningful insights to this field, particularly in the practical application of these technologies in real-world scenarios.

Introduction

The safety and longevity of infrastructure are paramount concerns in urban development and public safety. As the built environment ages, the structural integrity of buildings becomes susceptible to various forms of damage, with cracks being among the most common and concerning signs of potential failure. The detection and assessment of such structural damages are typically conducted through visual inspections, which are not only time-consuming but also prone to human error. In the digital era, automated methods for monitoring the health of structures are - increasingly vital, and semantic segmentation stands at the forefront of this technological revolution.

Semantic segmentation, the process of partitioning a digital image into multiple segments to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze, is a critical tool in the detailed assessment of building conditions. This technique has been instrumental in various domains, including medical imaging and autonomous driving, and is now being directed towards civil infrastructure to address the challenges of damage detection and classification.

The advent of deep learning has catalyzed significant advancements in image processing. Notably, the introduction of transformer architectures has marked a paradigm shift in the field. Initially conceived for natural language processing tasks, transformers have recently been adapted for visual purposes, demonstrating exceptional capabilities in image recognition and segmentation tasks.

In this project, we explore the synergy of convolutional neural networks (CNNs) and transformer models in the semantic segmentation of structural damage. Specifically, we focus on the application of the DeepLabV3, a proven CNN architecture, in tandem with emerging transformer-based models such as UNetFormer and MaskFormer, to detect and segment cracks in building structures.

Through this approach, we aim to leverage the strengths of both CNNs, which are adept at capturing spatial hierarchies, and transformers, which excel in modeling long-range dependencies, to advance the
field of structural integrity assessment.

Background

The endeavor to automate the assessment of structural integrity post-seismic events marks a pivotal shift from traditional, manual inspections to sophisticated, machine learning-powered methodologies. At the heart of this shift is the evolution of semantic segmentation within computer

vision, a domain that has witnessed a significant transformation with the advent of Convolutional Neural Networks (CNNs). Groundbreaking works such as those by Chen et al. (2017) on the DeepLab series have been instrumental, introducing atrous convolutions and spatial pyramid pooling to capture complex structural patterns in images more effectively.

As the field has matured, the revolutionary impact of transformer models, which redefined natural language processing, has begun to permeate visual tasks. The Vision Transformer (ViT) introduced by Dosovitskiy et al. (2020) underscores this trend, showcasing how models that excel in capturing long-range dependencies can be adept at semantic segmentation challenges. This cross-pollination of CNNs with transformer architectures has led to the development of innovative models such as UNetFormer and MaskFormer, which leverage the spatial hierarchies learned by CNNs and the contextual understanding provided by transformers to advance the capabilities of semantic segmentation (Xie et al., 2021; Cheng et al., 2021).

Within the context of post-disaster structural assessment, specifically the aftermath of the 2017 Mexico City earthquake, the ability to accurately categorize varying damage types from visual data is paramount. Gupta et al. (2019) have previously demonstrated the potential of deep learning models in identifying and classifying structural damages in such scenarios, setting the stage for subsequent research. Our project is anchored by these developments, seeking to harness the strengths of both CNNs and transformer-based models to enhance the precision of automated damage detection systems. By applying these advanced deep learning models to the richly annotated datasets of earthquake-damaged structures, our work aims to contribute meaningful advancements to the critical field of urban disaster response and infrastructure rehabilitation, continuing the conversation initiated by the leading-edge research in this area.

Data Description

Our dataset comprises of a curated collection of images that document the structural damages sustained by buildings in the aftermath of the Mexico City Earthquake in 2017. This dataset was meticulously compiled from the resources of Datacenter Hub and augmented with photographs contributed by Professor Vedhus Hoskere to ensure a comprehensive visual representation of the damage.

The dataset is meticulously annotated at the pixel level, delineating the damaged areas into various categories. There are two primary annotation sets that classify the nature of damage observed. The fine damage annotations discern subtle details such as cracks, grooves, and exposed rebar, using a color-coding scheme where, for instance, black signifies no damage, red is used for cracks, and orange indicates exposed rebar. In contrast, the coarse damage annotations encompass broader damage types such as spalling and voids, employing a distinct color palette where black once again represents no damage, and colors like tomato and yellow are used for spalling and voids, respectively.

These annotations are critical for training and evaluating semantic segmentation models, providing a rich source of labeled data to develop algorithms that can accurately identify and categorize different types of structural damage. For this research, the annotations have been converted from their original color representation to single integer values corresponding to each damage class. This conversion is a preprocessing step intended to simplify the computational analysis and enhance the efficiency of the subsequent semantic segmentation process.

We also have statistics on the pixel distribution across different classes of damage annotations. This statistical breakdown is instrumental for understanding the dataset's composition, enabling a balanced approach in model training and evaluation.

Together, these elements of the dataset form the backbone of our project, allowing us to push the boundaries of current methodologies in structural damage detection and providing a substantial foundation for our exploration into the application of CNNs and transformer models for semantic segmentation in the context of urban disaster analysis and response.

Methodology

In this project, we tried a few different CNN, and Transformer methodologies to see how well they will react to the dataset that we have. At first, we started making a CNN model from scratch and faced the fact that apart from taking time and using too many resources, it would not end up with any considerable results with our available dataset. So, we tried to go through pre-trained models and fine-tune them to see how well can get the results.

1- ResNet 18 (CNN)

We started with a basic model. It is called, ResNet-18 architecture, which is a deep convolutional neural network (CNN), that was introduced first in 2016 (A. Einstein et al.). This model achieved state-of-the-art performance on the ImageNet classification task and quickly became a popular choice for various computer vision applications, including crack detection.

On the ImageNet dataset, they evaluated residual nets with a depth of up to 152 layers. This was 8 times deeper than VGG nets (Simonyan and Zisserman 2014), and they could result in 1st place on the ILSVRC2015 classification task.

Using Resnet on crack detection

Here, we can mention some of the ResNet-18 successful adoptions for crack detection:

- combined ResNet-18 with other modules to create a crack detection system for bridges, demonstrating robustness against noise and illumination variations (Zhang et al. 2023).

Fine Tunning

To use this as our network, we fine-tuned the ResNet-18 by adding a Conv-2D to the final structure weights and trained the model with our data set. To decrease the size of the model and solve different types of image ratios (horizontal and vertical), we resized the image sizes to 214*214 pixels and used a batch size of 8. 100 epochs were used in our model to train the model and extract the weights matrix. We used the both fine damage and coarse damage masks together as our data set.

2- DeepLabV3 ResNet50 (CNN)

The ResNet50 architecture itself was introduced in 2016 (He et al. 2016), and then the DeepLabV3 framework was added to it in 2017 (Chen et al. 2017).

Using DeepLabV3 ResNet50 on crack detection

- Liu et al. (2023) incorporated DeepLabV3 ResNet50 into a hybrid architecture with transformers for dam crack detection, showcasing improved accuracy and robustness against noise and illumination variations (Xiang et al. 2023).

Finetuning

Training test split was set to 80, 20, and tested hyperparameters where:

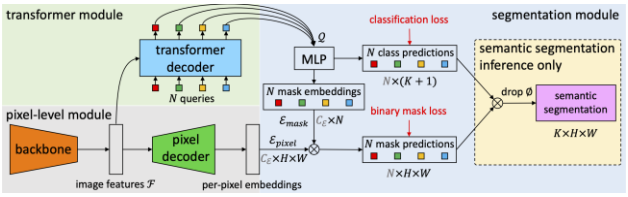
Image resize: (1024,1024), (512,512)
Batch size = 4, 8, 16
Number of epochs = 1, 10, 16, 100
Learning rate = 0.001, 0.0001

The best hyperparameters after training the above selected ones were as follow:

Batch size = 16
Learning rate = 0.001
Num epochs = 100
Num classes = 8

3- MaskFormer (Transformer)

The model we have used was introduced first in 2021. MaskFormer approach seamlessly converts any existing per-pixel classification model into a mask classification. Using the set prediction mechanism proposed in DETR (Carion et al. 2020), Mask Former employs a Transformer decoder (Vaswani et al. 2017), to compute a set of pairs, each consisting of a class prediction and a mask embedding vector. The mask embedding vector is used to get the binary mask prediction via a dot product with the per-pixel embedding obtained from an underlying fully convolutional network (NeurIPS-2021-per-pixel-classification-is-not-all-you-need-for-semantic-segmentation-Paper_2).



Using MaskFormer on crack detection:

- Explored the use of MaskFormer for retrieval of hybrid model (Qingguo et al. 2019).

Training errors

I trained a model called MaskFormer to recognize different types of damage in pictures of buildings. It was supposed to sort the damage into categories, a bit like drawing lines around each problem area. I had a bunch of photos to train it with, but there were a lot more pictures of some types of damage than others, which made it a bit tricky.

The MaskFormer did pretty well in figuring out the big, obvious damage when I ran it on the training set. But when I tried to get it working on the finer details, I hit a wall because there wasn't much out there to guide me on how

to do it right.

4- UNeTFormer (Transformer)

UNeTFormer was first introduced in 2022. This model uses residual blocks to improve the training depth of the network. To improve the segmentation accuracy of the network for small objects, a feature pyramid module combined with an attention structure is introduced. This improves the model's ability to learn deep semantic and shallow detail information. First, the proposed model is compared against other deep learning models and on two datasets, of which one was collected in a tank environment and the other was collected in a real marine environment (Zhao et al. 2022).

Finetuning

- Challenges:

Complex Architecture: TransMUNet's architecture required careful implementation, and adjusting hyperparameters was challenging.
Multiclass Adaptation: Similar to U-Net, adapting TransMUNet for multiclass segmentation was not straightforward.

- Solutions:

We leveraged an existing codebase and fine-tuned the model to match our requirements.
Custom data loaders and preprocessing steps were implemented to handle input data and mask variations.
Hyperparameters were fine-tuned, and a custom loss function was introduced for multiclass segmentation.

5- CrackSeg (Transmunet)

CrackSeg, a convolutional-transformer network augmented with Dilated Residual Blocks (DRB) and a Boundary Awareness Module (BAM), is designed to address the challenges of crack segmentation in complex environments. Its architecture is adept at capturing intricate local features of cracks while also integrating global contextual information. This dual capability enables CrackSeg to surpass the performance of existing algorithms on standard datasets, establishing it as a highly effective tool for precise crack segmentation tasks.

Finetuning

The implementation of CrackSeg in our project is based on the foundational code from the repository. However, to adapt the model to our specific requirements and constraints, several custom modifications were necessary:

- Data Preprocessing and Loader Customization:

To accommodate the hardware limitations and optimize computational efficiency, we resized the input images to a uniform dimension of 256×256 pixels.
A custom data loader class, Crack_loader, was developed to handle the loading and preprocessing of images and masks. This class includes image augmentations and transformations to enhance the model's robustness to variations in real-world data.

- Label Adjustment for Multiclass Segmentation

The original model was configured for only a binary segmentation. To extend its application to multiclass segmentation, we modified the label processing methodology.
A function, load_mask, was introduced to convert RGB mask images into a unique class map, ensuring compatibility with the multiclass segmentation framework.

- Debugging and Consistency

Throughout the implementation process, we encountered and resolved several challenges, such as ensuring the consistency of tensor shapes and data types across the model pipeline.
Special attention was given to the debugging of mask generation to ensure accurate and reliable segmentation results.

- Model Fine-Tuning:

The CrackSeg model was fine-tuned on our dataset, with adjustments made to hyperparameters such as batch size, learning rate, and the number of epochs to suit our specific use case. We employed a custom loss function, Custom_DiceBCELoss, to effectively train the model for multiclass segmentation tasks.

- Performance Evaluation and Visualization:

To assess the model's performance, we implemented a mean Intersection over Union (mIoU) calculation function. A visualization function was also developed to display the original images alongside their true and predicted masks, providing a clear and intuitive understanding of the model's segmentation capabilities.

- Conclusion

The fine-tuned CrackSeg model demonstrates significant promise in the field of crack segmentation. The custom modifications and enhancements made to the original architecture have tailored it to effectively address the specific challenges and requirements of our project. This bespoke approach ensures that the model not only retains its inherent strengths but also exhibits improved performance and adaptability in diverse real-world scenarios.

6- Segformer

- Data Loading Error

Encountered an issue where the data was not properly loaded as PyTorch tensors. Attempted to modify the collate

function and data loading methods to convert data into PyTorch tensors, but the problem persisted.

- Attribute Error

Faced an AttributeError while trying to move data to a GPU device. Tried to identify the correct location for GPU device assignment to resolve the error.

- Color Palette Request

Solved by creating a color palette for class labels, representing them as integers and providing corresponding color names in comments.

Despite these challenges, efforts were made to debug the data loading issue and train the SegFormer model.

Results

After running 4 models, we pick the 2 of the best and expand them for the best results to save time and energy. The results for each model is as follow:

1. ResNet 18

After running the model described in the methodology part, we could get the following results:

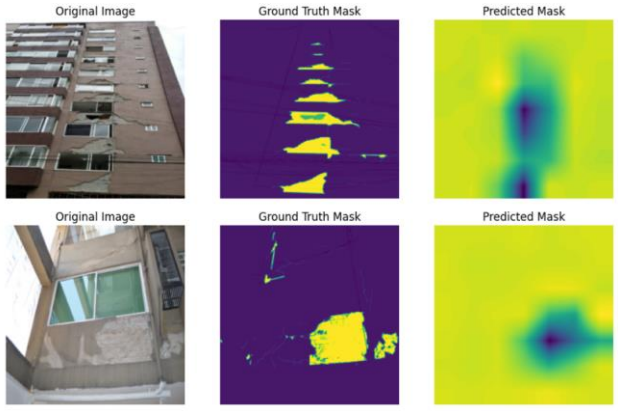


Figure 1 – two original pictures (left), and the ground truth masks related to them (middle) were plotted beside the generated images (right)

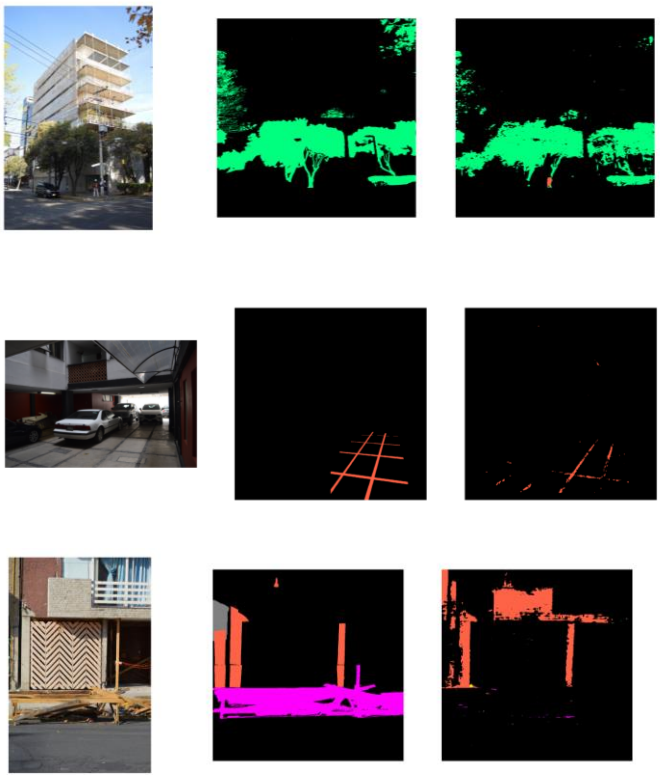
As we can see, a highlight related to the place of the damages was detected, however, not only the quality of the generated images is not good enough, but also, small damages and cracks are not predicted. This model needs to be trained way deeper to get better results, but as we have another more powerful model for CNN, we are going to update that one more.

2. DeepLabV3 ResNet50

In the upcoming figures, we will showcase the results of fine-tuning the DeepLabV3 ResNet50 model for crack detection tasks for both fine and coarse damage. The selected hyperparameters—batch size of 16, learning rate of 0.001, and 100 training epochs—have been optimized for performance. These visuals will demonstrate the model's proficiency in identifying and segmenting cracks across various surfaces, highlighting the efficacy of the model in handling multi-class segmentation problems with 8 distinct classes. Each figure underlines the model's advancements in accuracy and its ability to generalize well to different scales of crack detection challenges.

Results for Coarse Damage:

The model performed good on coarse damage datatset and particulary well for some specific classes



Results for Fine Damage:

The model does not perform that well as it did for coarse damage dataset

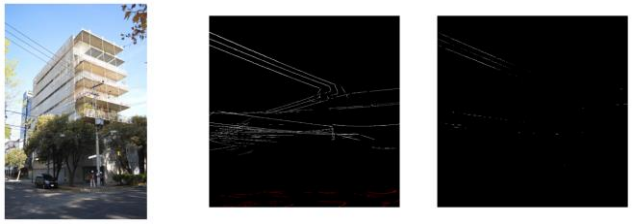


Figure 2 images, ground truth and predictions from deeplabv3

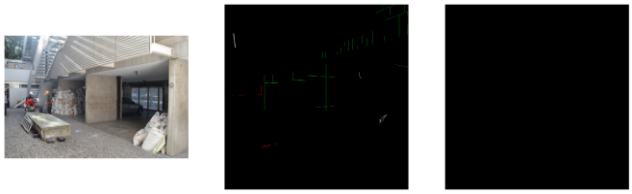


Table 1: Results for Deeplab V3

trial	model	epochs	MIoU
1	Deeplab V3 Resnet 50-coarse	100	0.42
2	Deeplab V3 Resnet 50-Fine	100	0.39

3. MaskFormer

In our trails with the MaskFormer model, we aimed to accurately segment structural damage within various images of buildings. The model was intended to distinguish and classify damage types effectively, which is reflected in the example output we have included in this report. In this output, the original image is displayed alongside the predicted mask generated by MaskFormer and the ground truth mask annotated by domain experts.

Upon review, we observed that our results did not align with our expectations. The predicted mask identified only a minimal area of damage, significantly less than what was detailed in the ground truth mask. This discrepancy suggests that the MaskFormer model did not perform the task of damage recognition to the desired standard.

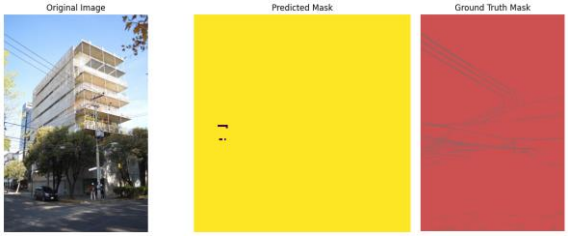


Figure 3 image, predicted mask, and the ground truth for Maskformer model on fine damage

4. UNeTFormer

The comparative analysis of trials suggests a significant finding: a less complex model with prolonged training surpassed the performance of a more complex model with shorter training in detecting coarse damages. This outcome underscores the model with a 64-depth decoder and 100 epochs of training as the best-performing configuration for coarse damage detection. Such insights not only inform the optimization of segmentation models for structural health monitoring but also underscore the delicate balance between model complexity and the necessary extent of training.

All the code was taken for the unetformer was taken from the git.

Figure 4 image, ground truth and the predictions for coarse data using unetformer

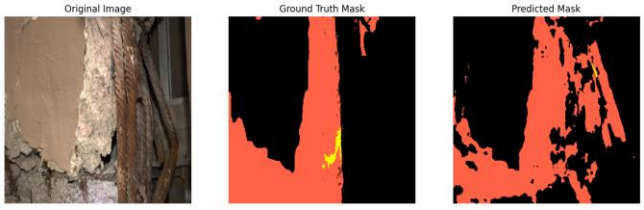


Figure 5 image, ground truth and the predcitions for coarse data using unetformer

Table 2: Results for UNetFormer

trial	model	epochs	MIoU
1	UNetFomer 64	100	0.45 (coarse)
2	UNetFomer128	20	0.35 (fine)

5. CrackSeg

From the left plot, we observe the training and validation loss curves. The rapid decline of both curves suggests that the model is learning effectively from the training data, and

the close proximity of the two lines indicates good generalization to the validation set. There is no apparent sign of overfitting, as the validation loss decreases alongside the training loss, which is an indication that the model is likely to perform well on unseen data.
The code was implemented from the following repository.

Table 3 : Results for Crackseg

trial	model	epochs	MIoU
1	Transmunet - coarse	100	0.30

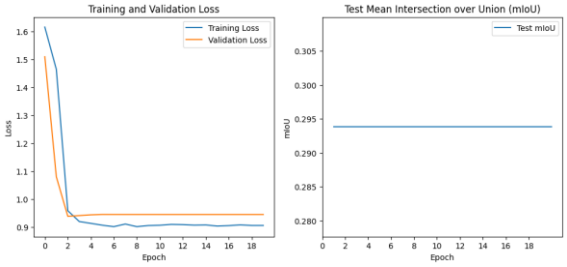


Figure loss6 and miou plot over epochs during CrackSeg training.

The decision to reshape input images to 256x256 pixels is noteworthy, particularly for a model with a substantial number of parameters, in this case, 22 million. This resizing step is often a practical consideration to balance computational efficiency with the ability to maintain sufficient detail for the model to perform accurate segmentation. By reducing the input image size, the model can process images more quickly, which is especially important given the large number of parameters that must be updated during training. However, care must be taken to ensure that important features, such as small cracks, are not lost or overly distorted in the resizing process, which could negatively impact the model's performance.

Conclusion

In our project, we tackled the task of detecting damage on structures, a critical issue for ensuring building safety. We faced the first challenge head-on: our data wasn't evenly distributed, meaning some types of damage were not as well represented as others. This can make it hard for models to learn about the less common damage types.

Despite this, we made good progress with a particular type of damage. Using a method called Deeplabv3 and another advanced technology known as Transformers, we managed to get solid results when identifying bigger, more obvious damage.

But not everything went smoothly. We tried to get some advanced models to work on our computers, which run on Linux, but hit several roadblocks. Some of these models were DiNAT, SwinTransformer, OnePeace, and Segformer. They are cutting-edge tools, but we couldn't get them to run properly, partly because our images had to be a certain size for the models to work with them, which was tricky to manage.

Another issue was that there isn't much research out there on identifying smaller or finer damage. This made it difficult for us to figure out which models would do a better job. Plus, most of the really good models that can generally be used for tasks like ours need to be run on Linux, and their large size meant we couldn't explore them as much as we wanted to because of limits on our computer's memory.

Looking to the future, we know there are better ways to train our models. We'd like to try using bigger, more powerful models without running into the problem of our computers running out of memory. We also want to try techniques like oversampling, where we make the less common types of damage appear more often in the training data so that the model can learn about them better. This is our way forward—making sure our models can learn from a balanced set of examples and become even better at spotting all kinds of damage.

Future works

For future work, we aim to refine our approach to address the challenges we encountered. A key focus will be on rebalancing our dataset to ensure that our models are not biased toward more frequent types of damage. By techniques such as data augmentation or synthetic data generation, we can give underrepresented damage types more prominence in the training process.

Another priority is upgrading our computational resources or optimizing our models to accommodate the large transformer-based architectures that promise improved accuracy. This may involve moving to cloud computing platforms that offer the necessary computational power without the limitations of our current hardware.

Additionally, we plan to delve into the emerging research on fine damage detection. As the body of work in this area grows, we'll integrate new findings and techniques to enhance our model's sensitivity to subtle damage cues.

Finally, we hope to foster collaboration with experts in material science and structural engineering. Their insights

could be invaluable in interpreting the complex patterns of damage that our models seek to understand, leading to more holistic and practical solutions in structural health monitoring.

Code Link

https://github.com/sujims22/Semantic_Segmentation_Cracks

Sharepoint

Link for sharepoint: CIVE-FP

Publication bibliography

A. Einstein; B. Podolsky; and N. Rosen: Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?

Carion, Nicolas; Massa, Francisco; Synnaeve, Gabriel; Usunier, Nicolas; Kirillov, Alexander; Zagoruyko, Sergey (2020): End-to-End Object Detection with Transformers. Available online at http://arxiv.org/pdf/2005.12872v3.

Chen, Liang-Chieh; Papandreou, George; Schroff, Florian; Adam, Hartwig (2017): Rethinking Atrous Convolution for Semantic Image Segmentation. Available online at http://arxiv.org/pdf/1706.05587v3.

NeurIPS-2021-per-pixel-classification-is-not-all-you-need-for-semantic-segmentation-Paper_2.