Segmentation of Floorplans and Heritage Sites: An Approach to Unbalanced Dataset

Sujith Gunjur Umapathy[1], Alexander I. Iliev[1, 2]

[1] SRH University Berlin, Charlottenburg, Germany
[2] Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia, Bulgaria
sujithgunjur.umapathy@stud.srh-campus-berlin.de,
ailiev@berkeley.edu

Abstract: To identify structural deficiencies, floorplan digitization is essential for legacy buildings, and historically significant and culturally rich sites. However, these technologies heavily depend on vector graphics. For an efficient and economical solution, we present an approach with multi-loss functions to handle unbalanced classes in floorplans using U-Net architecture.

Keywords: Deep Learning, Unbalanced Dataset, Semantic Segmentation, Focal Loss, Jaccard Loss.

1. Introduction

Floorplans are typically designed to show structural elements such as walls, doors, windows, rooms, staircases, activity-specific areas, etc. For historically significant places like tombs, pyramids, and ruined temples, the floor plan could indicate pillar positions, and pillar strength along with the access paths to important places. With advancements in tools and technologies, water lines, plumbing paths, and electrical lines are also represented on a floor plan (BigRentz, Inc, 2022). Developing a floorplan is a two-stage process:

1) Design the architectural plan for a house, building, or workspace.
2) Use CAD tools and develop vector graphics of the floor plan.

These vector graphics could either be in PDF or SVG format. From these procedures, we can identify certain pitfalls*, i.e.*, a) Floorplans may not have been prepared for old buildings/houses, b) Floor plan design and vectorization could be an expensive task, c) Floorplan vectorization tools like CAD provide a list of prerequisites that need to be met before vectorizing (*i.e*., architectural notations, metrics, markings, and file format), d) Floorplan vectorization is time-consuming. To ease out the process of floorplan vectorization and floorplan semantics simplification, the concepts in AI can be used ML/DL model functions in the following manner:

1) ML models/algorithms are trained on a specific dataset.
2) ML models classify and learn from the errors.

While minimizing the error, measured with the help of a loss function, the algorithm learns. Once these models are trained to solve a problem statement, we can ingest the models with a problem statement and receive prediction/classifications.

Techniques from DL are popularly used on datasets of higher dimensionality and higher volume. DL models tend to provide more accurate results (BigRentz, Inc, 2022). However, at the same time, these models must train on a larger dataset to receive predictions with higher confidence (Fig. 1).

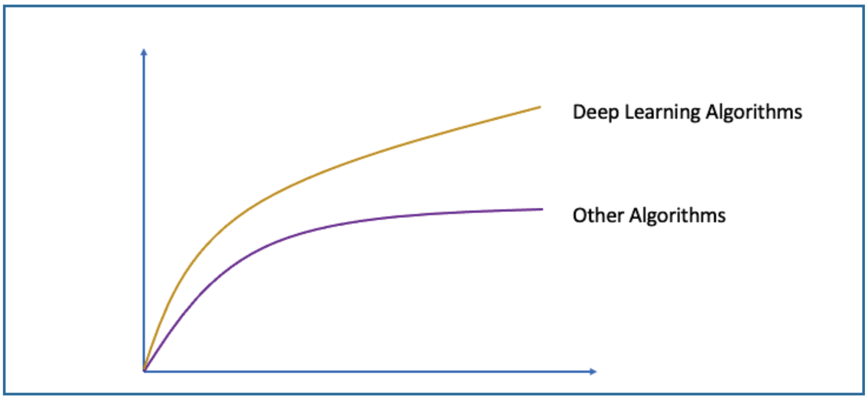

Fig. 1. ML v/s DL Performance

Floorplans rightly qualify for a DL solution, as floorplans are of higher resolution with higher Dots per inch (DPI). Numerous DL solutions have been explored in the science community. Scientific writing using the multi-class uncertainty loss for segmenting floorplan images (Kalervo, Ylioinas, Häikiö, Karhu, & Kannala, 2019) inspired our work. Further, this led to a research topic of segmenting floorplans accurately using DL techniques and SOTA loss functions. An early scientific paper on vectorizing floor plans (Liu, Wu, Kohli, & Furukawa, 2017) provides deeper insights into training a DL model for semantic segmentation. It is also noteworthy that the technique mentioned in (Liu, Wu, Kohli, & Furukawa, 2017) can be applied for 3D floorplan segmentation and generation. However, only 1000 images (Liu, Wu, Kohli, & Furukawa, 2017) were used to train and test the model. This can result in poor generalization, as highlighted in (Kalervo, Ylioinas, Häikiö, Karhu, & Kannala, 2019). In computer vision, three main DL techniques can be used to solve the problem of vectorizing a floorplan (Fig. 2). Namely:

a. ***Object Recognition***: Detect the position of each class (walls, doors, windows);
b. ***Semantic Segmentation***: Identify the class to which a pixel belongs to;
c. ***Instance Segmentation****:* Identify the subcategory of each class instance (different types of walls and windows).

Out of all the given techniques, semantic segmentation can be used to vectorize floorplans efficiently. This conclusion is derived from the following assumptions:

1) Instance segmentation would require annotations on a much more detailed level.
2) Annotators must be trained to identify different instances from the same class.

3) Semantic segmentation can give a blueprint of the floorplan semantics. Statistical methods can be used to derive the subcategory of instances.

Fig 2. Classification Methods

1.1 Semantic Segmentation

Semantic segmentation is a branch of DL, wherein the machine is asked to label each pixel into a specific class/category from the nominated list of candidates, *i.e.*, for example, if we want to segment an animal from the background, the underlying machine learning problem is to solve for the equation to identify if the pixel belongs to the class of animal or background (Fig. 3).

The expectation in floorplan segmentation can be seen in Fig. 5. The figure illustrates the segmentation of walls wherein every pixel belonging to the class -Wall needs to be classified correctly by the model classifier.

The Intersection over Union (IoU) score is a popular metric against which the model's performance is evaluated in segmentation. The IoU score defines the overlapping factor of the original image annotation against the predicted image. The IoU score lies between 0 and 1, translating the intersection from 0 to 100%. A classifier is reliable when the IoU score is near 1 (Fig. 4).

Fig 3. Semantic Segmentation (Consortium, 2015)

$$IOU = (Area\ of\ Overlap) \div (Area\ of\ Union)$$

Eq. 1 IoU score (Rezatofighi, et al., 2019)

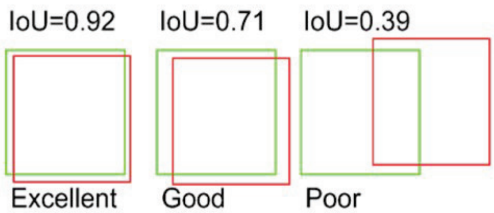

Fig 4. Intersection over Union

Dice co-efficient is also a popular metric to evaluate a segmentation model.

$$Dice = 2(|A \cap B|) \div (|A| + |B|)$$

Eq. 2 Dice co-efficient (Moore & Bell, 2020)

Where A ∩ B represents the common elements between sets A and B.

2. Floorplan Dataset

Cubicasa5K (Kalervo, Ylioinas, Häikiö, Karhu, & Kannala, 2019) produced a novel open-source dataset of floorplan images and annotations. This dataset (https://zenodo.org/record/2613548) comprises 5000 high-quality floorplan images along with their annotation SVG. However, this dataset comprises of a wide range of annotations for door, windows, different types of room etc. From the distribution chart (Fig. 6), we can identify the unbalance in the class labels. From the graph, it is evident that the door and window class have the highest distribution. If this scenario is not taken care during training, then the model tends to slide towards this distribution which can affect the Mean IoU on other classes like walls. Cubicasa5K also published an open-source code (https://github.com/CubiCasa/CubiCasa5k) in PyTorch and the trained weight graph in pickle format. However, utilizing the implementation/ weight files directly was a challenge due to the difference in the deep learning libraries.

Our source code was written using TensorFlow with a requirement of HDF weight files. Open Neural Network Exchange (ONNX - https://github.com/onnx/onnx) standard format could not be used due to legacy dependencies in the CubiCasa source code. This demanded training a DL model using the TensorFlow library. Data distribution in the CubiCasa dataset (Kalervo, Ylioinas, Häikiö, Karhu, & Kannala, 2019) can be seen in Table 1. We have extracted three classes of this data distribution, mainly walls, windows, and doors. Along with these three classes, background constitutes the fourth class.

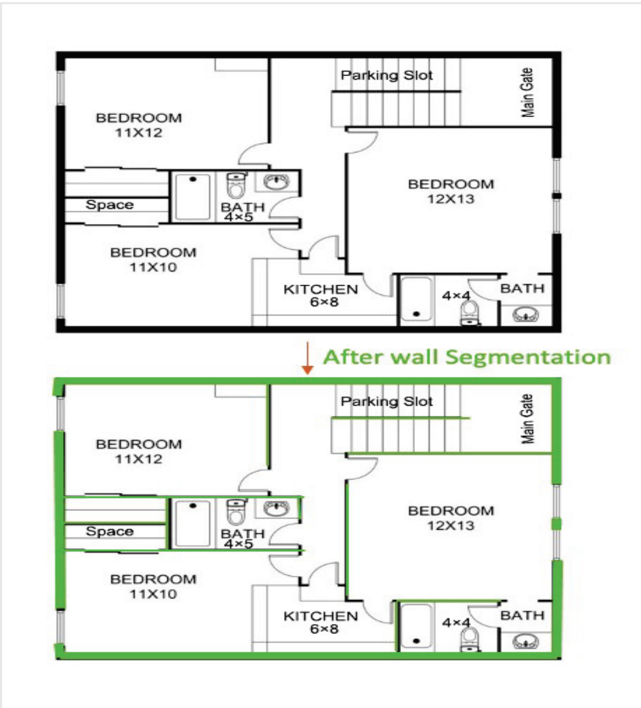

Fig 5. Floorplan segmentation illustration

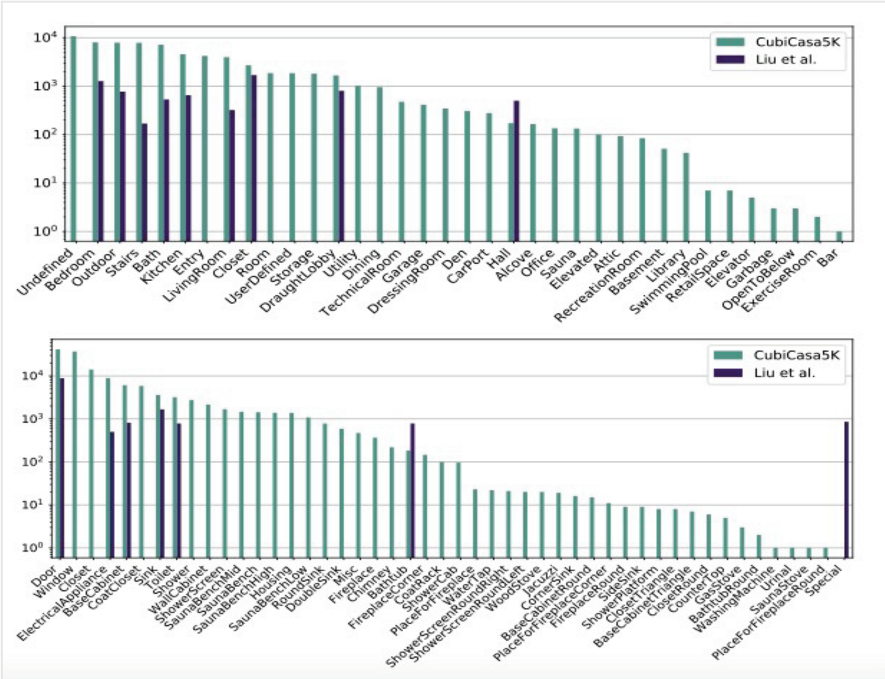

Fig. 6. CubiCasa dataset distribution (Kalervo, Ylioinas, Häikiö, Karhu, & Kannala, 2019)

3. Methodology

Standard U-Net architecture was used for faster and more precise segmentation (Ronneberger, Fischer, & Brox, 2015). However, extensive trials were performed to choose the backbone for the U-Net architecture. Post trails, ResNet-34 was selected as the backbone. The underlying steps to derive the pretraining weights file are as follows:

1) Structure directories for training, testing, and validation datasets. 4200 training images and 400 validation images were used in the training process.
2) The dataset comprised high-quality floor plan images; convoluting the images in higher dimensions was memory intensive. Hence the images were resized to 256 x 256 tiles and fed into the U-Net model.
3) Also, due to high volumes of data, a TensorFlow data generator was used to perform augmentations and tiling on the fly and flow in batches of 16 for model training.
4) Data were pre-processed with vector normalization and one-hot encoded using keras's to_categorical() to finally derive a shape of (n, h, w, 4).
5) With Adam optimizer and Categorical Focal plus Jaccard loss, the model has trained for ~80 epochs with early stopping. The focal loss was mainly included to reduce the effect of class imbalance (Lin, Goyal, Girshick, He, & Dollár, 2017) within the dataset.
6) Hyperparameter tuning was done to fine-tune the model metrics (*i.e.*, IoU score).

Table 1: Dataset size comparison

	(Dodge, Xu, & Stenger, 2017)	(de las Heras, Terrades, Robles, & S'anchez, 2015)	(Lin, Goyal, Girshick, He, & Dollár, 2017)	(Kalervo, Ylioinas, Häikiö, Karhu, & Kannala, 2019)
Images	500	122	815	5000
Room	NA	1320	7466	68877
Wall	NA	6089	16139	147024
Windows/ Doors/ Icons	NA	2345	15040	136676

$$Jaccard = |A \cap B| \div |A \cup B|$$

Eq. 3 Jaccard loss

As a part of the prediction phase, the user needs to load the full-scale floor plan images of dimension (h, w). These images are tiled to 256 x 256 and fed into the model for prediction. The predicted image is restored to the actual input size (h, w).

3.1 Overfitting

To overcome the problem of overfitting, we had to experiment with techniques such as:

1) ***Adding dropout layers*** (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014): In this phase, we had to experiment by dropping nearly 510% of the information from the Dense layer. Though the overfitting reduced in number, the test IoU score and validation IoU score was not crossing 75%.
2) ***Regularization*** (Ying, 2019): L2 Lasso Regression technique was used to add a penalty term to the loss function. Ridge Regression couldn't be used as we had minor classes, and we could not afford to remove features.
3) **Avoiding aggressive augmentation** (Zaworski, 2021): Augmented images promote generalization. However, aggressive augmentation can result in adverse effects. This causes the training and validation curves to diverge.

3.2 Combined Function

When classes are highly imbalanced, the model tends to predict in favor of the majority classes. However, several techniques include assigning the sample weights inversely to class frequency or by oversampling/ under sampling techniques. However, a novel method was proposed in this paper (Lin, Goyal, Girshick, He, & Dollár, 2017) for binary classification by implementing the Focal loss objective function. With the floor plan dataset, imbalanced classes are constituted with the background and walls occupying most of the sample set. By using focal loss, sample weight balancing or sample weight addition is not required. This loss also helps in reducing the false positives and false negatives (Carniato, 2021). Jaccard loss, also known as IoU loss, can be used to optimize the segmentation metric. IoU loss is like Binary cross-entropy (BCE) but without the quadratic component. IoU loss can be used to improve the segmentation model (van Beers, Lindstrm, Okafor, & Wiering, 2019). The combined loss function performed better than BCE in our multi-class segmentation problem (Fig. 7 and Fig. 8) and the results can be seen in Table 2. Further, we compare against the CubiCasa 5K implementation against our implementation in Table 3. Additionally, we can also see the SOTA IoU score from (Kalervo, Ylioinas, Häikiö, Karhu, & Kannala, 2019) in Table 4.

4. Results

In this section we will show the results from our model, as well as the graphical representation of the IoU score and the model loss curves for training and validation.

Table 2. Our segmentation results

Train Mean IoU	Validation Mean IoU	Test Mean IoU
84% (Weighted 4 class)	82% (Weighted 4 class)	81%

Table 3. Comparison of results (Kalervo, Ylioinas, Häikiö, Karhu, & Kannala, 2019)

Model	Val Mean IoU	Test Mean IoU
Cubicasa5K - Rooms	61.0	57.5

Table 4. CubiCasa Results (Kalervo, 2019)

Model	Recall	Precision
Dataset - Rooms	80.9	78.5
CubiCasa 5K (Kalervo, 2019)	90.0	87.6
Ours	91.3	85

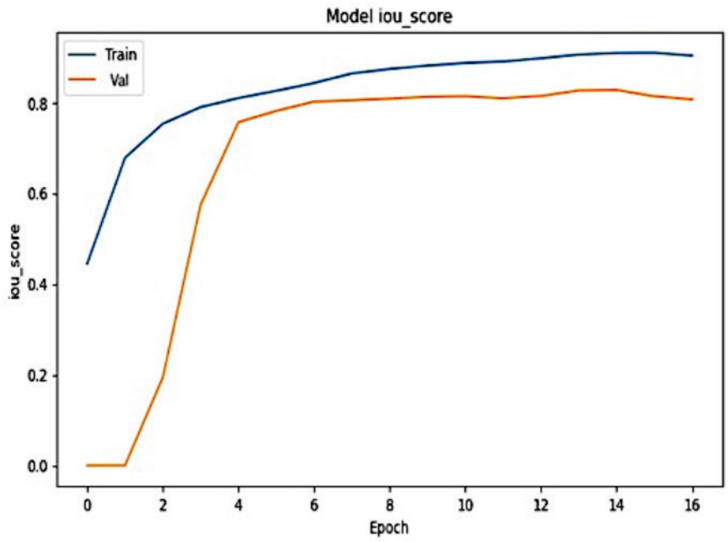

Fig. 7. IoU score of our model

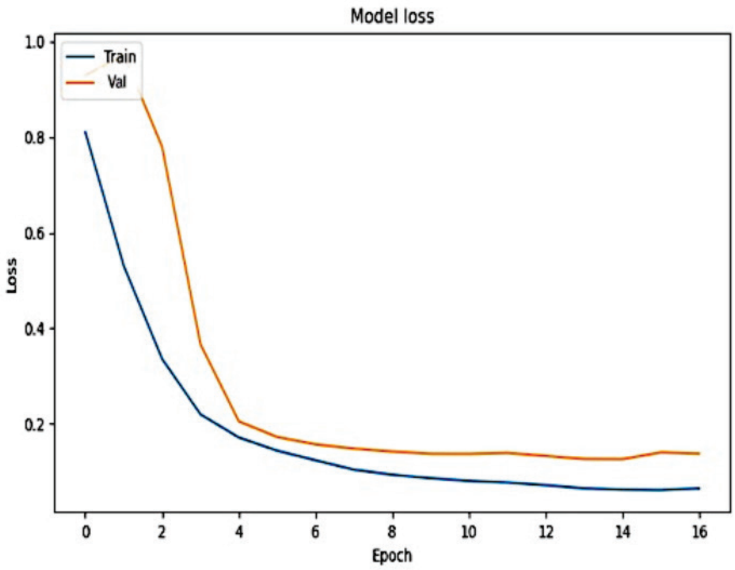

Fig. 8. Loss from our model

4.1 Prediction

Floor plan images were tiled in (h`, w`) dimension and were predicted using the pre-trained model. Segmentation results on the test dataset with walls (red), doors (cyan) and windows (yellow) can be seen in Fig 9. and Fig 10.

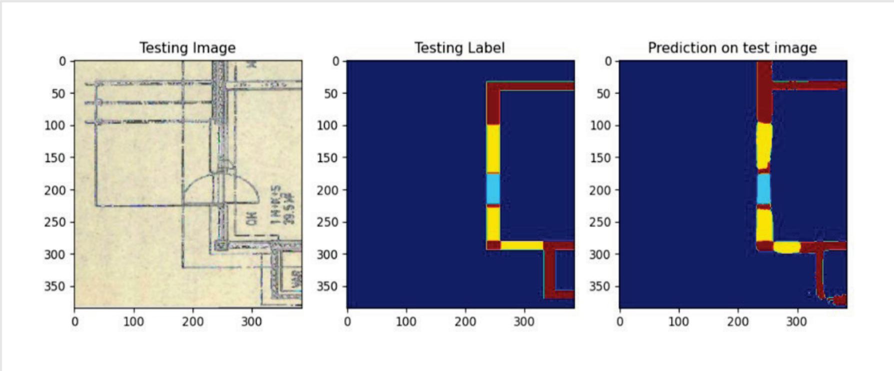

Fig. 9. Prediction-1 on test set

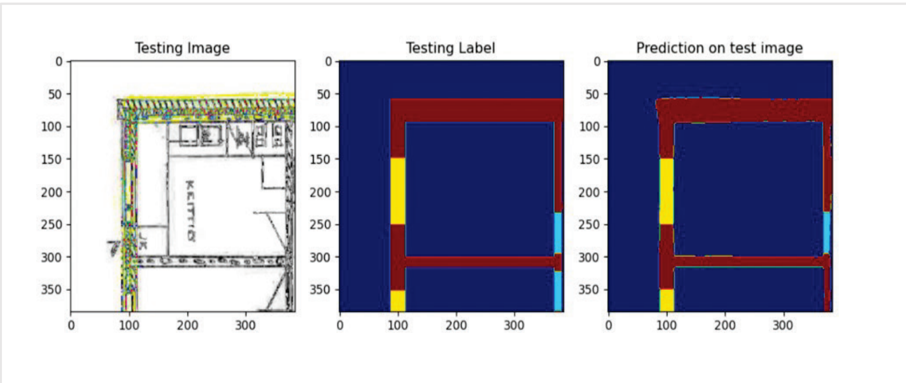

Fig. 10. Prediction - 2 on test set

5. Conclusion

Semantic Segmentation is a popular approach to enabling machine assistance in generating vector graphics for floorplans of legacy buildings, historically relevant sites, and segment remnants in archeological places.

Semantic segmentation is highly relevant when there are artifacts in floorplans or sitemaps. This makes it difficult for human annotators and CAD designers to interpret and redesign the document. However, trained scientific ML models generalize the concept of floorplan construction and site plans and help interpret the floorplan semantic.

This adds to the noble and novel approach of preserving documents about culturally rich and heritage places around the world.

Acknowledgements.

The research is supported by the Project BG05M2OP001-1.001-0003 "Center of Excellence in Informatics and Information and Communication Technologies" funded by the OP "Science and education for smart growth 2014-2020", EU through the European structural and investment funds.

References

BigRentz, Inc. (2022, July 10). *How to Understand Floor Plan Symbols*. Retrieved from BigRentz: https://www.bigrentz.com/blog/floor-plan-symbols

Carniato, L. (2021, April 6). *Multi-Class classification using Focal Loss and LightGBM*. Retrieved from Towards Datascience: https://towardsdatascience.com/multi-class-classification-using-focal-loss-and-lightgbm-a6a6dec28872

Consortium, C. (2015). *COCO - Common Objects in Context*. Retrieved from COCO - Common Objects in Context: https://cocodataset.org/#home

de las Heras, L.-P., Terrades, O., Robles, S., & S'anchez, G. (2015). CVC-FP and SGT: a new database for structural floor plan analysis and its groundtruthing tool. *International Journal on Document Analysis and Recognition*.

Dodge, S., Xu, J., & Stenger, B. (2017). Parsing floor plan images. *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, 358-361.

Kalervo, A., Ylioinas, J., Häikiö, M., Karhu, A., & Kannala, J. (2019). CubiCasa5K: A Dataset and an Improved Multi-Task Model for Floorplan Image Analysis. arXiv.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal Loss for Dense Object Detection. arXiv.

Liu, C., Wu, J., Kohli, P., & Furukawa, Y. (2017). Raster-to-Vector: Revisiting Floorplan Transformation. *International Conference on Computer Vision (ICCV)* (pp. 2214-2222). 2017 IEEE.

Moore, C., & Bell, D. (2020, March 15). *Dice similarity coefficient. Reference article*. Retrieved from https://doi.org/10.53347/rID-75056

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized Intersection over Union. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 1929-1958.

van Beers, F., Lindstrm, A., Okafor, E., & Wiering, M. (2019). Deep Neural Networks with Intersection over Union Loss for Binary Image Segmentation. In *Proceedings*

of the 8th International Conference on Pattern Recognition Applications and Methods (pp. 438--445).
Ying, X. (2019). An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*, 022022.
Zaworski, R. (2021, February 14). *Medium*. Retrieved from Medium: https://medium.com/snowdog-labs/data-augmentation-techniques-and-pitfalls-of-small-datasets-e5a657fc404f

Received: June 12, 2022
Reviewed: June 28, 2022
Finally Accepted: July 10, 2022