Utrecht University

Self-Supervised Learning of River Morphology with Visual Pretext Tasks

Mark J. van der Pijl
Utrecht University
Utrecht, The Netherlands

0. *Abstract*

Unsupervised visual representation learning remains a significant challenge in computer vision, particularly in recognizing the morphology of different river types from geographical data. Current research has focused on pretext tasks for self-supervised learning using colored images, leaving a gap in unannotated geographical data analysis. This study addresses this gap by applying several self-supervised models to unannotated geographical data. Through a series of experiments, the models' abilities to learn useful visual representations are evaluated. The effectiveness of these methodologies in geographical morphology recognition is critically assessed.

1. Introduction

Over the past decades, computer vision systems have advanced exponentially [1]. Models designed for tasks such as object recognition and detection often outperform humans on complex benchmarks [2], [3], [4]. However, these models rely heavily on large, annotated datasets and are typically task-specific. For instance, ImageNet [5] is trained to recognize 1,000 different categories. Consequently, implementing such models on data with different dimensions, such as black-and-white images, can be challenging.

The Faculty of Geosciences of Utrecht University is particularly interested in recognizing the morphology of the vast variety of rivers on earth. Many studies classify these rivers in four categories: Anastomosing, Braided, Wandering and Meandering rivers [6], [7]. However, a single river is often made up of more than one morphology type and can have meandering sections leading into braided or anastomosing and back to meandering, which makes the classification of a single morphology challenging.

This paper focuses on self-supervised learning, a subclass of unsupervised learning, to try and recognize the different river morphologies.

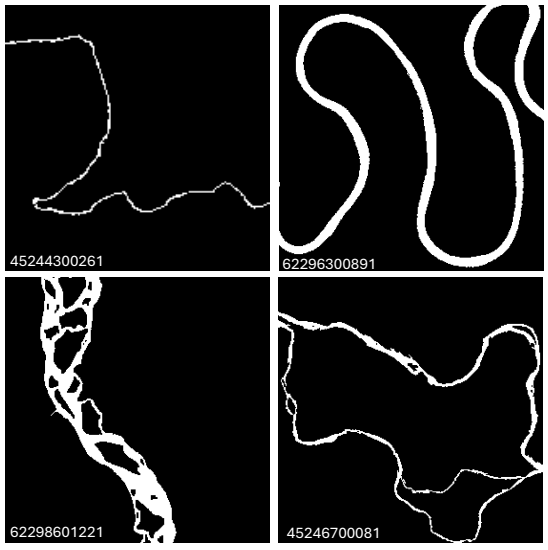

Figure 1: Examples of the four river morphologies with id (left to right, top to bottom): Wandering, Meandering, Braided and Anastomosing. Wandering rivers exhibit a sporadic morphology, while Meandering rivers follow a sinusoidal pattern. Braided rivers feature multiple channels within a single riverbed, whereas Anastomosing rivers have multiple channels that split and merge.

Self-supervised learning techniques produce unsupervised representations of the data that can be used as a transfer learning model for later annotated modeling. The self-supervised learning framework does not require annotated data in order to formulate pretext learning tasks such as image rotation [8] or jigsaw-solving [9], for these tasks generate their own predictive measurements; the rotation of the image and the position of the jigsaw-pieces, respectively. As a result, the middle layers of the Convolutional Neural Networks (CNNs) trained for solving these pretext tasks encode complex semantic visual representations that can be used for solving other tasks, like predicting the type of morphology of a river once an annotated dataset is available.

In Chapter 5 the experimental results of the trained pretext tasks are presented. To hold the reproducibility of these experiments, the code is provided in the appendix [10].

2. Related Work

Self-supervision is a learning framework that generates supervised signals for pretext tasks automatically, aiming to learn representations that effectively address downstream tasks. Due to its generic structure, self-supervision covers a broad range of applications. A pioneering effort in self-supervised learning is the use of pretext tasks to generate supervisory signals from the data itself. Doersch et al. (2015) [11] introduced a method where the pretext task involved predicting the relative positions of image patches, enabling the network to learn useful visual representations. These representations, which include spatial hierarchies, edges, and object parts, benefit various computer vision tasks such as object detection, segmentation, and classification. Similarly, Noroozi and Favaro (2016) [9] demonstrated that solving jigsaw puzzles as a pretext task helps in learning semantic features that transfer well to other tasks.

The application of self-supervised learning to remote sensing and geographical data has seen some preliminary work. For example, Ayush et al. (2020) [12] applied self-supervised techniques to satellite imagery, demonstrating the potential for these methods to extract meaningful features from unannotated geographical data. Their aim was to improve land cover classification and change detection in satellite images by learning robust representations without relying on labeled data. Their work underscores the feasibility of our approach and highlights the gap that our study aims to fill by specifically targeting river morphology recognition.

Finally, A study has been conducted by Chen Zheng (2024) [13] on Alzheimer's disease classification using 3D convolutional neural networks, which makes use of rotation classification and image reconstruction of black-and-white images during pretext training. These methods proved to be valuable for their research by enhancing the network's ability to understand and process 3D medical images, which are crucial for accurate disease classification. This highlights the versatility and potential of self-supervised learning techniques, making them promising for our task at hand in river morphology recognition.

3. Dataset

The data utilized in this experiment originates from The Surface Water and Ocean Topography River Dataset (SWORD) [14] and the Global River Widths from Landsat (GRWL) Database [15]. The images from GRWL depict the Earth in black-and-white, with the surface shown in black and water in white. The SWORD dataset contains the centerlines of these images. All river reaches, cut up segments of rivers between confluence points, in Asia and South America have been considered because of the extensive variety of river types found on these continents.

Every reach is clipped based on the extreme coordinates of the river. A box is created around these extremes with an additional buffer of 10% to account for errors in the original dataset. Subsequently, the reaches are filtered according to labels given by the SWORD dataset. The dataset includes river types and lake flags that identify the type of water body. Only rivers are considered, while lakes and other water bodies are excluded. Following this, any images containing empty or gray pixels are discarded, which is a result of the discrepancies between the centerline dataset of SWORD and the binary satellite images from GRWL. These images either contain null data or residual lake clips. Finally, only images with at least one dimension of 30 pixels are included, as smaller images contain insufficient morphological information. The filtering results in 24,709 unique river segments. Examples of the resulting images are presented in Figure 1.

4. Methodology and Evaluation

This chapter explores the self-supervised learning models that are designed to recognize river morphology from unannotated geographical data. The focus is on training Deep Convolutional Neural Networks (DCNNs) and Deep Convolutional Inverse Graphics Networks (DC-IGNs) through various pretext tasks. These pretext tasks, including image rotation, inpainting, and jigsaw puzzle solving, are crafted to enable the models to learn visual representations without the need for labeled datasets.

a. CNN Models

The pretext tasks are trained using DCNNs, with each task necessitating a specific structure. For instance, a rotation pretext task can be learned using a DCNN, whereas image reconstruction requires a DC-IGN structure.

A DCNN consist of multiple 2D convolutional layers, as proposed by Fukushima et al. (1982) [16], later formalized by Lecun et al. (1998)[17]. Each layer is followed by a ReLu activation function [18] and a 2D max pooling layer [19], [20] for batch normalization. The output of the final pooling layer is then used in a fully-connected layer comprising multiple layers with ReLU activation functions in between. The final layer employs a Sigmoid activation function [21] instead of ReLu. The sigmoid activation function, serving as a binary output function, has demonstrated superiority over the other activation functions [22], despite slower learning speed during backpropagation [23]. This final layer is then transformed into the appropriate number of predicted values.

A DC-IGN [24] is a neural networks designed primarily to reconstruct its input. Its structure comprises three sections: an encoder network, fully connected layers, and a decoder network. The first two sections resemble the suggested DCNN. The final fully connected layer serves as the input for the decoder section, which reconstructs the image using 2D transposed convolutional layers. These upsampling layers generate the output feature map [25]. Each layer concludes with a ReLu activation function, and the output of the final ReLu function is reshaped to the appropriate output size.

b. Pretext Tasks

Various pretext tasks are employed during the development of the experiments. Some tasks are executed exactly as initially designed, while others are adapted to investigate new opportunities and research areas. This strategy enables a thorough assessment of different techniques and their effectiveness in diverse contexts, thereby expanding the study's scope. The five pretext tasks under examination are briefly outlined below.

Rotation DCNN

Gidaris et al (2018) [8] proposed a pretext method that involves creating four copies of the original image, each rotated by 0°, 90°, 180°, and 270°. These images are then fed into a single network tasked with predicting the applied rotation. A robust model should learn to recognize patterns within the images, regardless of their rotation, demonstrating its ability to understand and interpret the underlying structure of the images. Figure 2 provides a visual representation of the DCNN for the rotation pretext task.

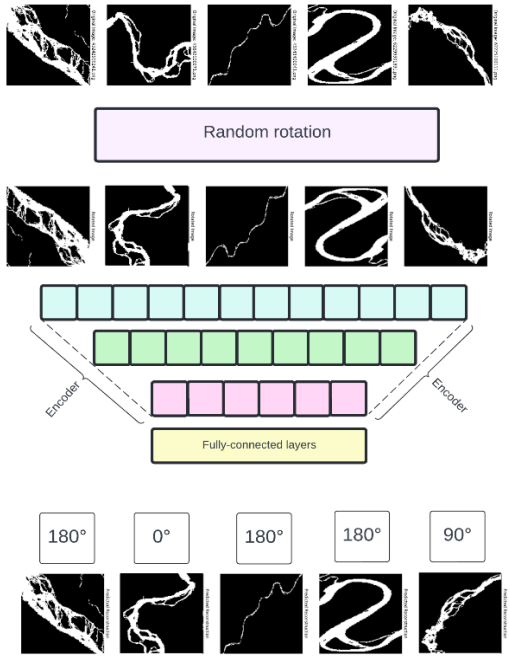

Figure 2: A visual representation of the DCNN network for the Rotation pretext task

Inpainting DC-IGN

Image inpainting is a generative pretext task proposed by Pathak et al. (2016) [26] ,which aims to learn representations by filling in a missing patch of the original image. This technique involves three variants of masking: central patch, random patches, and random regions. This research focuses solely on the central patch variant. The model learns to reconstruct the missing central part of the image, enhancing its ability to understand contextual information. Figure 3 provides a visual representation of the network's structure.

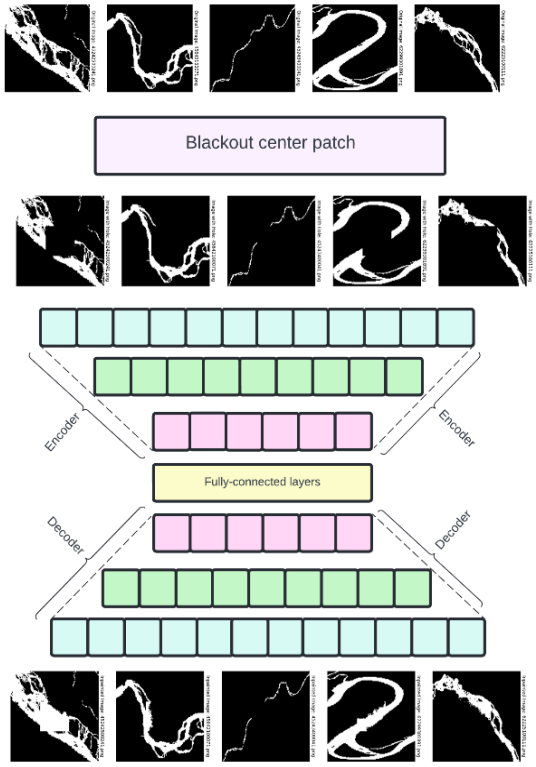

Figure 3: A visual representation of the DC-IGN network for the Inpainting pretext task

High-Context Inpainting DC-IGN

A self-proposed variant of the random region inpainting method has been developed to reconstruct the original image by filling in the masked patch and the rest of the image pixel by pixel. The blacked-out patch is chosen based on the region of the original image containing the most information. Similar to the original inpainting task, a competent model should learn the spatial relationships between the blacked-out and unedited patches and accurately fill in all the pixels. This approach aims to enhance the model's ability to understand and recreate complex patterns within the image. Figure 4 provides a visual representation of the DC-IGN for the High-Context Inpainting pretext task.

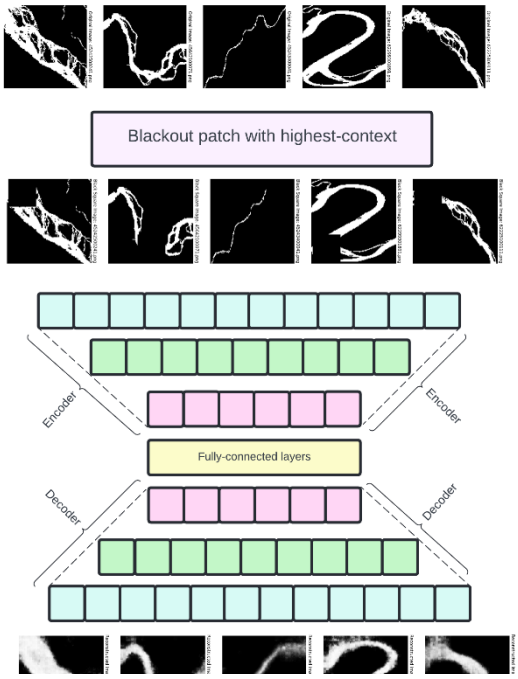

Figure 4: A visual representation of the DC-IGN network for the High-Context Inpainting pretext task

Jigsaw DCNN

The Jigsaw DCNN task involves dividing an image into nine equally sized patches arranged on a 3x3 grid and then randomly shuffling these patches. The primary objective is for the model to learn and predict the relative spatial positions of these nine patches to determine their original order accurately. This task requires the model to focus on understanding the spatial relationships between the patches, rather than just their sequential order. To train the model effectively, multiple permutations of the shuffled patches are fed into it, as proposed by Noroozi et al. (2016) [9]. However, increasing the number of permutations leads to a larger volume of input data, which in turn demands more computational power. A visual representation of the jigsaw concept can be found in Figure 5.

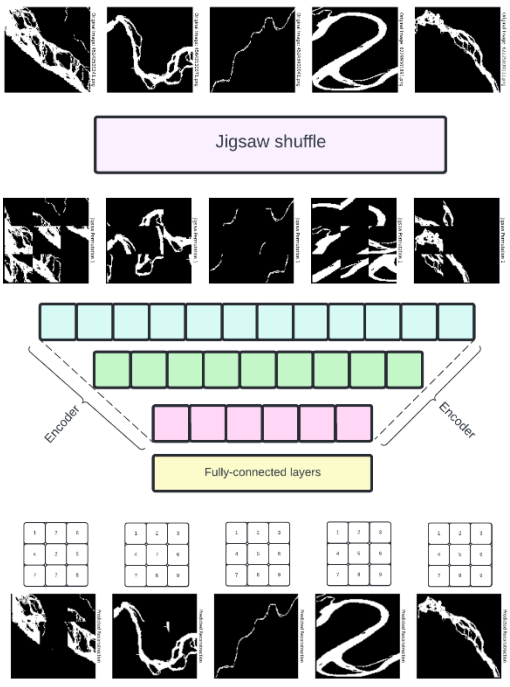

Figure 5: A visual representation of the DCNN network for the Jigsaw pretext task

Jigsaw DC-IGN

The Jigsaw DC-IGN task shares similar properties with the regular Jigsaw DCNN task, but with a different focus. Instead of predicting the relative positions of the patches, the Jigsaw DC-IGN task aims to reconstruct the original image pixel by pixel. This is achieved using the encoder, fully connected layers, and decoder structure of the DC-IGN. By reconstructing the image in this manner, the model is able to learn global patterns in the data, capturing more comprehensive contextual information. The process of reconstructing the image enhances the model's understanding of spatial relationships within the image. A visual representation of this structure is shown in Figure 6.

c. Assessing Learned Visual Representations

During training and validation, the Cross Entropy Loss [27], [28] is used for efficiently validating the trained layers. The DCNN models make use of a simple implementation of the Cross Entropy Loss. The DC-IGNs use the Cross Entropy Loss in combination with an extra weight for correctly predicting the white pixels in an image, because a one-to-one ratio of black and white pixels resulted in fully black generated images, a result of the skewed ratio in black and white pixels in the data.

Finally, a set of five different reaches with notable patterns are manually chosen as a test set for visual assessments. These images are not seen by the model during training and validation, making them an effective visual test set. Five images are selected because they provide a manageable number for detailed visual inspection, allowing for thorough assessment without being overwhelming. The original, mutated and predicted images are visualized side by side in Chapter 5.

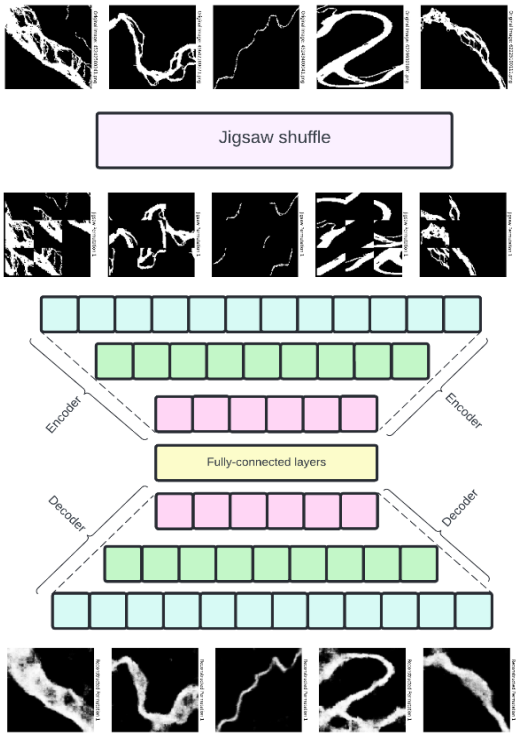

Figure 6: A visual representation of the DC-IGN structure of the jigsaw shuffle pretext task

Table 1: Train, validation and test loss for the different models of the pretext tasks. The DCNN models use a regular Cross Entropy Loss, where the DC-IGN the inbalance of black and white pixels models use a Weighted Cross Entropy Loss to account for the inbalance of black and white pixels. For each model, the first epoch where the validation loss does not decrease is chosen as the best model.

	DCNN models					DC-IGN models						
	Rotation	Jigsaw				Inpainting	High-Context inpainting	Jigsaw				
		10x	25x	50x	100x			1x	5x	10x	25x	100x
Train	0.7547	0.0101	0.0259	0.0247	0.0516	0.1427	0.0955	0.0457	0.1059	0.1047	0.1334	0.1445
Validation	0.7844	0.0466	0.0751	0.1110	0.1093	0.1622	0.1051	0.0558	0.1228	0.1050	0.1449	0.1500
Best epoch	14 of 25	9 of 10	7 of 10	9 of 10	9 of 10	7 of 20	15 of 20	14 of 50	13 of 20	19 of 25	18 of 25	15 of 50

[1]*Loss of the best found epoch*

[2]*Loss of the last ran epoch*

5. Experiments and Findings

This section presents the results of the small-scale study, divided into two parts due to the differences in model structure and evaluation methods. All pretext tasks are trained on the unannotated SWORD dataset. The dataset is split into a 90:10 ratio for the training and validation sets, respectively, using a fixed random seed to ensure consistent comparisons. Five images are excluded from the training and validation process for testing, as detailed in Chapter 4.c. This setup allows for a systematic evaluation of the models' performance across different tasks and ensures the reliability of the obtained results. All found results are summarized in Table 1.

a. Evaluation on DCNN models

Two different DCNN pretext tasks are experimented on: the rotation and the jigsaw tasks. Each are trained and validated using the same data and similar model structures, where only the last outcome layer is modified to fit the task at hand.

Rotation DCNN

There are only four possible permutations when rotating an image: 0°, 90°, 180°, and 270°. This limited variation enables the randomization of the rotation of input images for each epoch during the training of the Rotation pretext task model. As demonstrated in Figure 7, the optimal model was identified at epoch 14, achieving a Cross-Entropy Loss of 0.7844 on the validation set. Notably, this validation loss is quite high, suggesting a certain level of uncertainty in the model. Consequently, stopping the training after 14 epochs yielded an effective model. This is evidenced by its perfect predictions on the five test cases, illustrated in Figure 8.

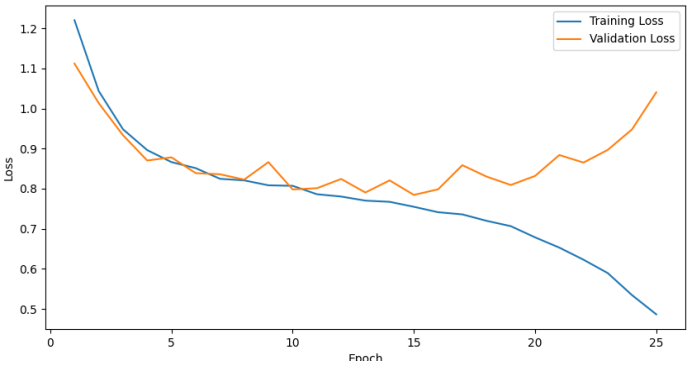

Figure 7: Train and Validation Cross-Entropy Loss of the Rotation Pretext Task

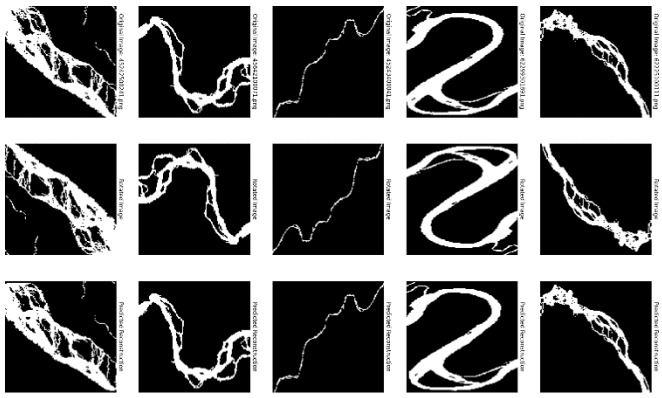

Figure 8: Results of the 14th epoch of the Rotation DCNN Pretext Task. The top row shows the input images, the middle row the rotated images, and the third row the predicted images.

Jigsaw DCNN

The Jigsaw Pretext task model is trained using 10, 25, 50, and 100 permutations of the jigsawed image to explore how the number of permutations influences the model's performance. As depicted in Figure 9, the learning curves change significantly as the model processes an increasing number of permutations. The graphs clearly show that all four models ultimately converge to a similar conclusion, achieving comparable loss values. However, the time required for convergence increases with the number of permutations. Specifically, models trained with fewer permutations reach optimal performance more quickly, while those with higher permutations require more epochs to achieve similar results. This suggests that while handling more permutations might offer a more comprehensive learning process, it also demands additional epochs to achieve optimal performance. These observations underline the trade-off between the number of permutations and the training efficiency of the models.

The optimal models achieved Cross-Entropy Losses of 0.0466, 0.0751, 0.1110, and 0.1093 after the 9th, 9th, 7th, and 9th epochs for 10, 25, 50, and 100 permutations, respectively. The first three models performed identically on the test set, demonstrating consistent accuracy and reliability. However, the model trained with 100 permutations exhibited slightly inferior performance compared to the other models. This discrepancy is particularly evident in the leftmost test river depicted in Figure 10, where the predictions of the 100 permutation model are less accurate. This indicates that while increasing the number of permutations might enhance the model's learning capacity, it can also introduce complexity that might not necessarily translate to improved performance on external data.

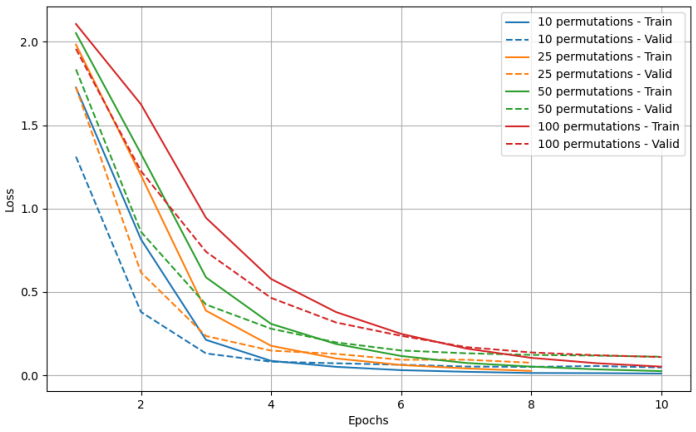

Figure 9: Training and Validation Cross-Entropy Loss for the Jigsaw DCNN Pretext Task across four distinct models: 10, 25, 50, and 100 permutations

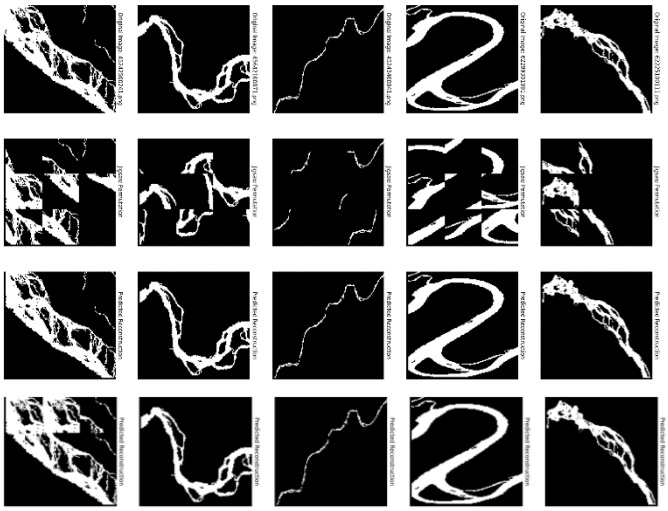

Figure 10: Jigsaw DCNN Pretext Task results. The top row shows the input images, the middle row the rotated images, the third row the predicted images for 10, 25, and 50 permutations, and the fourth row for 100 permutations.

b. Evaluation on DC-IGN models

The DC-IGN pretext task models are evaluated separately due to their added complexity. These tasks include inpainting, high-context inpainting, and jigsaw DC-IGN tasks.

Inpainting DC-IGN

The inpainting task is designed without permutations because it inherently involves filling in a single, specific area. In this case, the center patch of the image is blacked out, similar to the background color of the river images, which are also black. Therefore, the center patch remains constant. To guide the model more effectively, a weighted loss ratio of 5:1 is applied, giving more importance to white pixels over black pixels during the training process.

The best model achieved a Cross-Entropy Loss of 0.1622 on the validation set after 7 epochs. Training for more than 7 epochs causes the model to overfit, as illustrated in Figure 11. While the training loss continues to decline steadily, the validation loss stabilizes and eventually worsens. Figure 12 showcases the model's generative capabilities, which are optimal when the river is broad and continuous without multiple channels, as seen in the fourth example. However, the model struggles with complex, braided river structures, like those in the first and last examples. Additionally, when the river is relatively thin, the model often misinterprets it two separate rivers or as noise, resulting in no predictions at all.

Interestingly, an overfit model appears to have better generative capabilities for more complex riverbeds. As shown in Figure 13, the model attempts to predict the black islands within the braided river structure while maintaining decent performance on other types of rivers.

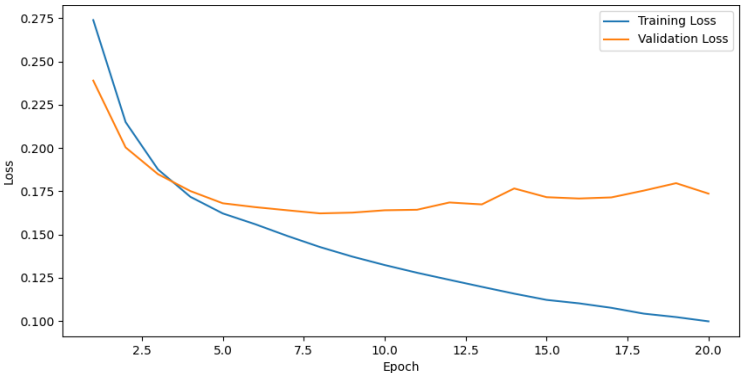

Figure 11: Train and Validation Cross-Entropy Loss of the Inpainting DC-IGN Pretext Task

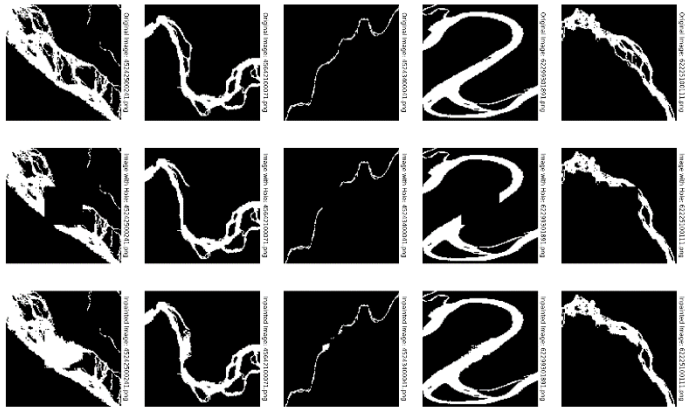

Figure 12: Results of the 7th epoch of the Inpainting DC-IGN Pretext Task. The top row shows the input images, the middle row the rotated images, and the third row the predicted images.

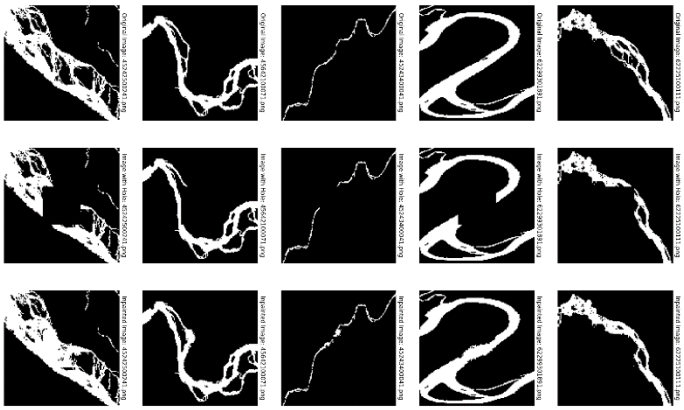

Figure 13: Results of the 19th epoch of the Inpainting DC-IGN Pretext Task. The top row shows the input images, the middle row the rotated images, and the third row the predicted images.

High-Context Inpainting DC-IGN

The high-context inpainting task shares the same design as the center patch inpainting. A patch covering 1/9 of the total image area, containing the most white pixels, is identified. This patch is then blacked out and provided as input to the model, resulting in a single possible permutation of the input image.

Figure 14 illustrates the model's steady improvement, with a slight bump observed at the 7th epoch. This bump is likely due to the cyclic learning rate activating when the validation loss surpasses the training loss. This approach temporarily degrades the model's performance for an epoch, allowing it to explore alternative strategies, which ultimately has a positive effect. The optimal model is achieved after 15 epochs, with a Cross-Entropy Loss of 0.1051 on the validation set.

The best found model seems to have issues recognizing the overall pattern of the rivers. The blacked-out patch is often reconstructed with noticeable inaccuracies, particularly in complex river structures. As seen in Figure 15, the third, fourth and last examples show significant discrepancies between the predicted and actual river patterns. The model struggles with capturing the detailed features of braided rivers, leading to blurred or incomplete reconstructions. Despite its steady improvement in Cross-Entropy Loss, these visual inconsistencies highlight the challenges faced by the model in generalizing across diverse and complex river structures.

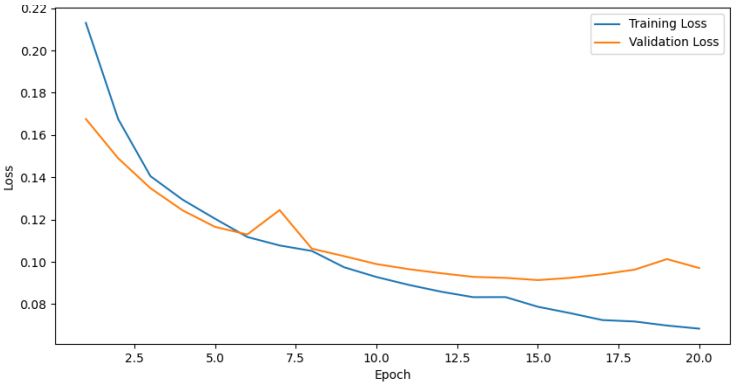

Figure 14: Train and Validation Cross-Entropy Loss of the High-Context Inpainting DC-IGN Pretext Task

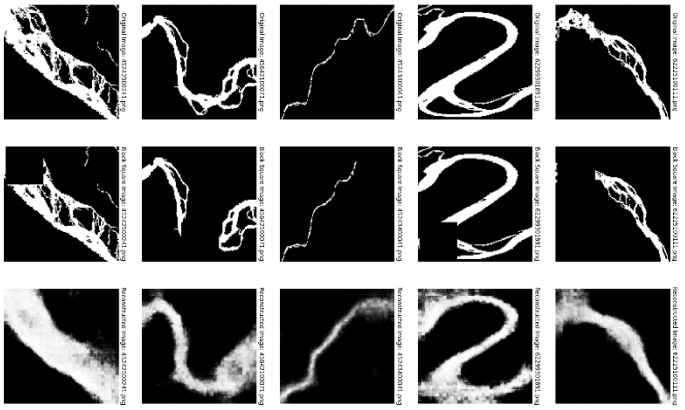

Figure 15: Results of the 15th epoch of the High-Context Inpainting DC-IGN Pretext Task. The top row shows the input images, the middle row the rotated images, and the third row the predicted images.

Jigsaw DC-IGN

The jigsaw DC-IGN pretext task is capable of generating multiple permutations. Specifically, 1, 5, 10, 25, and 100 permutations are created and used as inputs for the model. The training duration varies for different models; some are trained longer while others are stopped earlier based on their learning progress. This adjustment ensures that each model is trained for an optimal period, ceasing training when the model has either fully learned the task or reached a plateau in its learning curve.

When incorporating more than one permutation of the input data, the models encounter difficulties in identifying the overall morphology of the river bands. This issue is reflected in the learning curves shown in Figure 16. Models with 5, 10, 25, and 100 permutations exhibit similar trends, where increasing the number of permutations leads to a higher Cross-Entropy Loss in the best-performing models. Despite the additional data permutations, the models struggle to achieve lower loss values, indicating that a greater number of permutations complicates the learning process and hinders the models' ability to accurately capture the river morphology.

When examining the test images in Figure 17, it becomes clear that models utilizing more than one permutation struggle with accurately recognizing river morphology. In contrast, the model with just one permutation demonstrates a strong ability to identify the riverbend's morphology and capture details of complex multi-channel rivers, as seen in the first, second, and fourth images. Additionally, it effectively identifies a clear single-channel river, as evident in the third image. The increased number of permutations seems to introduce noise, making it harder for the models to generalize and maintain focus on essential river features.

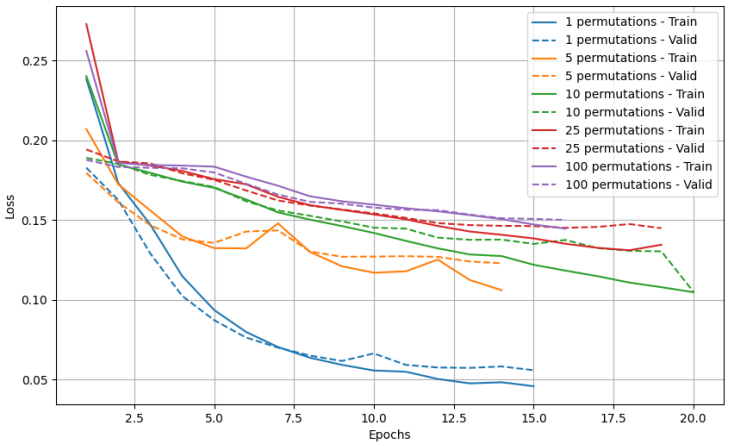

Figure 16: Training and Validation Cross-Entropy Loss for the Jigsaw DC-IGN Pretext Task across five distinct models: 1, 5, 10, 25, and 100 permutations

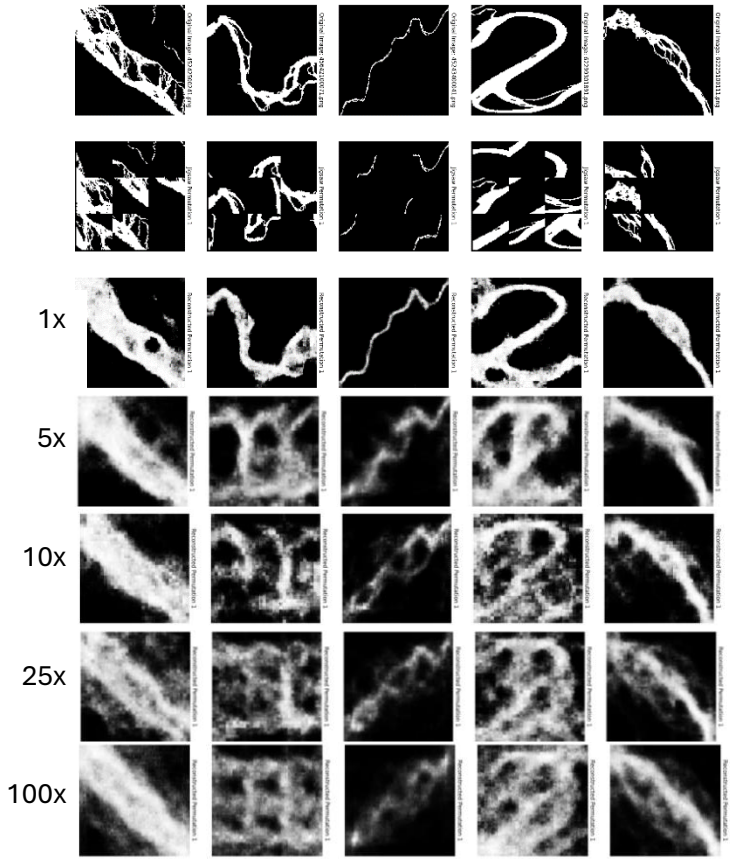

Figure 17: Jigsaw DC-IGN Pretext Task results. The top row shows the input images, the middle row the rotated images, the third to last row the predicted images for 1, 5, 10, 25, and 100 permutations.

6. Discussion

This chapter discusses the key limitations and considerations encountered during the research, focusing on pooling layer effectiveness, challenges with the inpainting pretext task, and hardware constraints. These insights are crucial for understanding the study's limitations and guiding future improvements.

Pooling Layer Considerations

The 2x2 max-pooling layers were used to extract patterns in the data. Max pooling is particularly effective for edge detection and is widely employed in training neural networks. However, it is highly sensitive to noise. The dataset contains numerous side branches or tributaries, which can be interpreted as noise, leading the models to overfit on these features. This overfitting can diminish model performance. Considering this, alternative pooling methods, such as median pooling, may offer better results for this specific dataset. Median pooling could reduce the impact of noisy branches, providing a more robust feature extraction process.

Discarded Inpainting Pretext Task

A variant of the implemented high-context inpainting pretext task was quickly discarded due to the model's inability to generate any useful output. The task aimed to reconstruct just the blacked-out patch of the image based on surrounding context. However, the models struggled with this challenge, failing to produce coherent or meaningful inpainted regions. This inability likely stemmed from insufficient context or complexity within the dataset, rendering the task ineffective for training purposes. As a result, this approach was deemed unsuitable for further exploration in this research. However, the method might prove effective if more samples are added to the dataset.

Hardware Limitations

The research was conducted using a PC with an Intel Core i7-13700K (16 cores, 24 threads), 64GB DDR5 RAM, an NVIDIA GeForce RTX 4080 16GB GPU, and 1TB SSD storage. While the GPU facilitated model training, the single GPU setup limited the ability to perform multiprocessing efficiently. Utilizing multiple GPUs can significantly enhance training speed and model complexity by parallelizing computations and handling larger datasets, and exploring more complex models in the same time span. However, this isn't feasible with the current single-GPU configuration, highlighting a need for more advanced hardware in future research to fully leverage multiprocessing benefits.

7. Further Research

Given the limited timeframe of this research, certain methodologies and approaches were not fully explored. This chapter outlines those unexamined strategies and suggests future directions for enhancing the robustness and performance of the models.

Dataset comparison and Expansion Strategy

Much of the previous research in this area has utilized large datasets, often consisting of hundreds of thousands of images. For instance, C. Zheng [13], employed the LDM100K neuroimaging dataset, which contains 100,000 images, while Noraazi et al. [9] utilized a random subset of the ImageNet dataset comprising 1.3 million images. In contrast, the current study's pretext tasks were trained on a comparatively modest dataset of approximately 25,000 images. These images encompass the entire continents of Asia and South America.

To enhance the robustness and generalizability of the models, it is proposed to expand the dataset to include images from all continents. This expansion would increase the dataset size to approximately 100,000 images, aligning it more closely with the dataset sizes used in

related studies. Incorporating a larger dataset may significantly impact the learning curves of more complex models, particularly for tasks such as inpainting and jigsaw puzzle solving with multiple permutations. Such an increase in data volume could provide these models with a richer variety of examples to learn from, potentially leading to improved performance and more nuanced understanding of the river morphologies.

Skipping model validation

Skipping traditional model validation in favor of training the models on the entire dataset might be advantageous in this specific case. When using data from every continent, all relevant data is included, eliminating the need to validate and test the model against external data. However, one risk of this method is overfitting the models to noise in the data. This approach is particularly compelling if all global data is used and noise in the river segment clips is minimized, potentially leading to more accurate and comprehensive models. The results of fitting a model while ignoring validation is shown in chapter *Inpainting DC-IGN in* Figure 13*,* where the model appears to have better generative capabilities for more complex riverbeds.

Stratified Splitting of the dataset

Implementing stratified splitting of the dataset based on the average pixel value could enhance the models' performance. For instance, an image with a white pixel fraction of 0.10 may indicate a small, meandering river, whereas a white pixel fraction of 0.40 might suggest a braided or anastomosing river. By applying stratified splitting, an equal distribution of river types is ensured across both training and validation datasets. This approach balances the dataset, preventing bias towards any particular river morphology and eventually leading to more accurate and generalized model predictions. Stratified splitting thus aids in achieving more robust and reliable results.

Implementation of Clustering

To better compare the found pretext task models, one can implement a clustering method to find the best clusters of the final non-predictive fully-connected layer of the model. These clusters can be visualized using a 2D TSNE plot [29]and assessed using a Silhouette score [30] to see how well these models can categorize the input rivers in different groups.

Finetuning model

In a perfect scenario, the layers of the pretext task models are frozen and used as the first set of layers for a finetuning model. This model should take the learned visual representations of the entire dataset and use these to train a more successful model on a small annotated dataset.

8. Conclusion

This research advances the understanding and application of self-supervised learning in recognizing river morphology from unannotated geographical data. By employing various visual pretext tasks, the study evaluates the effectiveness of these methodologies in learning useful visual representations without labeled data.

The study assesses several pretext tasks using Deep Convolutional Neural Networks (DCNNs) and Deep Convolutional Inverse Graphics Networks (DC-IGNs), including rotation, jigsaw, inpainting, and high-context inpainting. These tasks challenge the models to understand and reconstruct complex patterns within the images.

For DCNN models, the rotation task proved effective, achieving a Cross-Entropy Loss of 0.7844 on the validation set after 14 epochs. This model generalized well, accurately predicting rotations on the test set. The jigsaw task also showed promising results with fewer permutations, achieving Cross-Entropy Losses

of 0.0466, 0.0751, and 0.1110 for 10, 25, and 50 permutations, respectively. However, the model with 100 permutations performed slightly worse, indicating that more permutations can complicate the learning process without improving accuracy.

The DC-IGN models faced more challenges. The inpainting task achieved a Cross-Entropy Loss of 0.1622 after 7 epochs but struggled with complex braided river structures. The high-context inpainting task showed steady improvement but had difficulties accurately reconstructing blacked-out patches, especially in intricate river systems. The jigsaw DC-IGN task with a single permutation performed well, capturing the morphology of riverbends and some of the complex multi-channel rivers. However, models with multiple permutations encountered difficulties, highlighting the trade-off between the number of permutations and model performance. Despite these challenges, these models could be superior in later fine-tuning tasks due to their complex structure.

Experiments showed that self-supervised learning techniques can indeed learn meaningful visual representations from unannotated geographical data. The jigsaw tasks were particularly effective for DCNN models, while the DC-IGN models showed potential but need further refinement for more complex tasks like inpainting and high-context inpainting.

Despite limitations, this research underscores the potential of self-supervised learning in geographical morphology recognition. Training models on unannotated data opens new possibilities for large-scale geographical analysis, where obtaining labeled data is often difficult and time-consuming. Future research should focus on expanding the dataset, exploring alternative pooling methods, further evaluations using clustering or fine-tuning tasks, and leveraging advanced hardware to enhance the training process. Addressing these areas can further improve the robustness and performance of self-supervised models in geographical data analysis.

9. Acknowledgements

I would like to express my deepest gratitude to my supervisors at Utrecht University, D. Beelen and N.N.P. Collot d'Escury, for their continuous support, guidance, and insights throughout my research. Additionally, I am grateful to J.H. Nienhuis for reviewing my thesis. I also extend my thanks to my colleague, M. van Boeckel, for being an excellent discussion partner throughout my thesis work.

10. References

[1] Ray Kurzweil, “The Singularity Is Near: When Humans Transcend Biology,” 2005.

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” pp. 1026–1034, 2015.

[3] S. Dodge and L. Karam, “A Study and Comparison of Human and Deep Learning Recognition Performance under Visual Distortions,” in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, IEEE, 2017, pp. 1–7. doi: 10.1109/ICCCN.2017.8038465.

[4] A. S. Lundervold and A. Lundervold, “An overview of deep learning in medical imaging focusing on MRI,” *Z Med Phys*, vol. 29, no. 2, pp. 102–127, May 2019, doi: 10.1016/J.ZEMEDI.2018.11.002.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.

[6] L. Whitcomb, “ANASTOMOSING VS. BRAIDED STREAMS,” *Proceedings of the Pennsylvania Academy of Science*, vol. 21, pp. 64–68, 1947, [Online]. Available: http://www.jstor.org/stable/44112178

[7] J. M. Buffington and D. R. Montgomery, “9.36 Geomorphic Classification of Rivers,” in *Treatise on Geomorphology*, Elsevier, 2013, pp. 730–767. doi: 10.1016/B978-0-12-374739-6.00263-3.

[8] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised Representation Learning by Predicting Image Rotations,” Mar. 2018.

[9] M. Noroozi and P. Favaro, “Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles,” Mar. 2016.

[10] M. J. van der Pijl, “Self-Supervised-Learning-of-River-Morphology.” Accessed: Jul. 07, 2024. [Online]. Available: https://github.com/MJvanderPijl/Self-Supervised-Learning-of-River-Morphology

[11] C. Doersch, A. Gupta, and A. A. Efros, “Unsupervised Visual Representation Learning by Context Prediction,” May 2015.

[12] K. Ayush, B. Uzkent, M. Burke, D. Lobell, and S. Ermon, “Generating Interpretable Poverty Maps using Object Detection in Satellite Images,” Feb. 2020.

[13] C. Zheng, “Self-Supervised Pretext Tasks for Alzheimer’s Disease Classification using 3D Convolutional Neural Networks on Large-Scale Synthetic Neuroimaging Dataset,” Jun. 2024.

[14] E. H. Altenau, T. M. Pavelsky, M. T. Durand, X. Yang, R. P. de M. Frasson, and L. Bendezu, “The Surface Water and Ocean Topography (SWOT) Mission River Database (SWORD): A Global River Network for Satellite Data Products,” *Water Resour Res*, vol. 57, no. 7, Jul. 2021, doi: 10.1029/2021WR030054.

[15] G. H. Allen and T. M. Pavelsky, “Global River Widths from Landsat (GRWL) Database (Version V01.01) [Data set].,” *2018*. doi: http://doi.org/10.5281/zenodo.1297434.

[16] K. Fukushima and S. Miyake, “Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position,” *Pattern Recognit*, vol. 15, no. 6, pp. 455–469, Jan. 1982, doi: 10.1016/0031-3203(82)90024-3.

[17] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[18] K. Fukushima, “Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements,” *IEEE Transactions on Systems Science and Cybernetics*, vol. 5, no. 4, pp. 322–333, 1969, doi: 10.1109/TSSC.1969.300225.

[19] K. Yamaguchi, K. Sakamoto, T. Akabane, and Y. Fujimoto, “A neural network for speaker-independent isolated word recognition,” in *First International Conference on Spoken Language Processing (ICSLP 1990)*, ISCA: ISCA, Nov. 1990, pp. 1077–1080. doi: 10.21437/ICSLP.1990-282.

[20] H. Gholamalinezhad and H. Khosravi, “Pooling Methods in Deep Neural Networks, a Review,” Sep. 2020.

[21] S. Narayan, “The generalized sigmoid activation function: Competitive supervised learning,” *Inf Sci (N Y)*, vol. 99, no. 1–2, pp. 69–82, Jun. 1997, doi: 10.1016/S0020-0255(96)00200-9.

[22] A. Mahgoub, “Optimizing Bank Loan Approval with Binary Classification Method and Deep Learning Model,” *Open Journal of Business and Management*, vol. 12, no. 03, pp. 1970–2001, 2024, doi: 10.4236/ojbm.2024.123104.

[23] J. Han and C. Moraga, “The influence of the sigmoid function parameters on the speed of backpropagation learning,” 1995, pp. 195–201. doi: 10.1007/3-540-59497-3_175.

[24] T. D. Kulkarni, W. Whitney, P. Kohli, and J. B. Tenenbaum, “Deep Convolutional Inverse Graphics Network,” Mar. 2015.

[25] H. Gao, H. Yuan, Z. Wang, and S. Ji, “Pixel Transposed Convolutional Networks,” *IEEE Trans Pattern Anal Mach Intell*, pp. 1–1, 2019, doi: 10.1109/TPAMI.2019.2893965.

[26] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros, “Context Encoders: Feature Learning by Inpainting,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, 2016, Accessed: Jun. 23, 2024. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2016/papers/Pathak_Context_Encoders_Feature_CVPR_2016_paper.pdf

[27] A. Mao, M. Mohri, and Y. Zhong, “Cross-Entropy Loss Functions: Theoretical Analysis and Applications,” Apr. 2023.

[28] I. J. Good, “Rational Decisions,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 14, no. 1, pp. 107–114, 1952, [Online]. Available: http://www.jstor.org/stable/2984087

[29] L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE.,” *Journal of machine learning research*, vol. 9, no. 11, 2008.

[30] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *J Comput Appl Math*, vol. 20, pp. 53–65, Nov. 1987, doi: 10.1016/0377-0427(87)90125-7.