Improved Difference Images for Change Detection Classifiers in SAR Imagery Using Deep Learning

Filip Pospisil

Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru
Email: filip.pospisil@student.um.si

Change detection in Synthetic Aperture Radar (SAR) imagery is crucial for environmental monitoring and disaster assessment, but it is hindered by speckle noise and acquisition inconsistencies. The study "Improved Difference Images for Change Detection Classifiers in SAR Imagery Using Deep Learning" proposed a neural network-based mapping transformation to generate artificial SAR images with consistent acquisition conditions, improving difference image (DI) quality, and reducing false positives. This paper tries to replicate the original experiment using the same dataset and validation methods but with a reduced computational scope. The replication follows a structured SAR change detection pipeline: preprocessing for noise reduction and alignment, DI formation using deep learning instead of traditional algebraic methods, and classification using thresholding and Support Vector Machines (SVMs). A subset of the data set is used to compare deep learning-enhanced DIs against conventional methods. This replication assesses the feasibility of deep learning-based SAR change detection in computationally constrained systems, validates the original findings, and explores a broader applicability.

Keywords: sar, machine vision, detection, difference image, neural networks

1. INTRODUCTION

1.1. Problem statement

Synthetic Aparture Radar (SAR) has become a crucial tool in remote sensing change detection, enabling consistent monitoring of earth surface under diverse weather conditions. Unlike optical imaging, which offers higher spatial resolution but is limited by cloud cover and daylight availability, SAR operates in the microwave spectrum allowing data acquisition regardless of atmospheric conditions. This makes SAR particularly valuable for applications such as post disaster damage assessment, forest damage detection, and long-term environmental monitoring, including deforestation or glacier melting.

Despite its advantages, SAR imagery presents unique challenges. The technique relies on a moving antenna to synthesize a virtual aperture, which results in speckle noise, a grainy pattern caused by random backscatter within each resolution cell. Even with advanced noise-suppression algorithms, speckle remains an issue, reducing the clarity of the difference image (DI) generated by subtracting or combining temporal SAR acquisitions. To improve DI quality and enhance the accuracy of change detection, additional pre-processing steps are required.

- **Preprocessing**: Apply speckle filtering and radiometric calibration to ensure accurate comparisons.
- **Difference Image (DI) Formation**: Generating a DI using algebraic operations (e.g., subtraction, ratio, or log-ratio) between two SAR images.
- **Classification**: Identifying changed regions in the DI through clustering or machine learning classifiers.

To address the challenges of DI quality, Alatalo et al. proposed a neural network-based mapping transformation function that improves the comparability of SAR image by normalizing acquisition conditions, considering imaging angles, digital elevation models and weather data. Their method, tested using Sentinel-1 SAR imagery over Finland, demonstrated significant improvements in DI quality and change detection accuracy compared to conventional approaches.

In this study, I aim to replicate the core methodology of Alatalo et al. using the same dataset and validation techniques while adapting the computational scale due to hardware constraints. Given the resource-intensive nature of deep learning models, our approach will focus on a subset of the available data to assess whether similar improvements in DI quality and classifier performance can be achieved under limited computational resources. By systematically evaluating noise suppression, DI formation, and classification steps, this replication study seeks to validate the effectiveness of deep learning-based SAR image processing and explore its feasibility for practical applications on less powerful computing systems.

2. METODOLOGY

2.1. Steps of original study

The study proposed an improved SAR change detection method that enhances difference image (DI) formation using a neural network-based mapping transformation function. Their approach involved the following key steps:

Dataset & Preprocessing

Dataset: Sentinel-1 SAR imagery from North-East Finland, with environmental data from the Finnish Meteorological Institute (FMI) and Digital Elevation Model (DEM) from the National Land Survey of Finland (NLS).

Preprocessing Steps:

- **Speckle noise reduction** using a neural network instead of traditional filters.
- **Geometric corrections & coregistration** to align images properly.
- **Radiometric normalization** to reduce the intensity variations of acquisition differences.
- **Metadata incorporation**: Imaging angles, weather conditions, and orbit direction were included to improve image comparability.

Deep Learning-Based Difference Image (DI) Formation

Traditional Methods: Typically, DIs are formed using basic algebraic operations (subtraction, ratio, or log-ratio), which are prone to noise and acquisition inconsistencies.

Proposed Method: A neural network was trained to generate artificial SAR images that match the acquisition conditions of the most recent observation.

The network learned from historical SAR images, DEM, and weather metadata to predict what the SAR image should look like without real-world changes.

Why? This reduces false positives and provides a cleaner, more reliable DI.

Change Detection & Classification

Experiment Setup:
Simulated changes were introduced in test images to evaluate classifier accuracy.

Two classifiers were tested:

- Threshold-based classifier (unsupervised),
- Support Vector Machine (SVM) (supervised).

The traditional DI and deep learning-enhanced DI were compared using ROC curves and AUC scores.

2.2. Results of original study

Improvement in Difference Image (DI) Quality

The proposed method produced cleaner DIs with less noise and fewer artifacts than traditional DI methods.

False positive changes (caused by environmental variations) were significantly reduced because the network accounted for imaging angle, soil moisture, and orbit conditions.

Better Classification Performance

Threshold Classifier (Unsupervised Method):

- Deep learning-enhanced DI had an AUC of 0.87
- Traditional DI had an AUC of 0.79
- This shows that the proposed method led to improved change detection accuracy even with a simple thresholding approach.

Support Vector Machine (SVM) Classifier (Supervised Method):

- Deep learning-enhanced DI outperformed traditional DI, with higher classification accuracy and lower false detection rates.
- False positives were lower, meaning the classifier misidentified fewer unchanged regions as changes.

Ablation Study (Impact of Metadata on Model Performance)

The study tested what happens when some metadata (e.g., weather data) is removed from the model.

Removing satellite orbit direction and precipitation data significantly decreased accuracy, proving their importance in improving DI formation.

Even without weather data, the method still outperformed traditional DI methods, but with reduced effectiveness.

Computational Efficiency & Practicality

Training time was about 30 hours on an NVIDIA V100 GPU. Preprocessing took longer than traditional methods, but classification was more efficient due to improved DI quality. The method is scalable but requires substantial computational resources for training.

3. RESULT REPLICATION

Overview of Replication Approach

This section describes the process of replicating the study "Improved Difference Images for Change

Detection Classifiers in SAR Imagery Using Deep Learning" using the publicly available code from the authors' GitHub repository (https://github.com/janne-alatalo/sar-change-detection). Due to computational limitations, several adjustments were made in terms of dataset size, batch processing, and training time. While the original study utilized a large-scale dataset and powerful computing resources, my approach focused on a scaled-down version, ensuring feasibility within the constraints of a weaker machine.

Computational Constraints and Adjustments

The original study employed extensive training data and computational resources, including an NVIDIA V100 GPU and a batch size optimized for large-scale processing. My system, however, had significantly lower processing power, requiring several modifications:

- **Dataset Reduction**: Instead of using the full dataset, I processed only 130 parts of the input data and half of the validation dataset.
- **Batch Size Adjustment**: The original implementation used a larger batch size, but my system encountered failures at higher values. The batch size was reduced to 8, which ensured stability.
- **Extended Training Time**: Due to the reduced computational power, training took several hours.

List of all modifications is in Table 1;

Environment Setup

The first step involved setting up the computational environment as specified in the repository. Additionally, TensorFlow and PyTorch were installed to ensure compatibility with the neural network implementation. Given the lower specifications of my hardware, I ensured that memory consumption was optimized by disabling GPU acceleration and using CPU-based processing where necessary.

Data Preparation

The original dataset consisted of Sentinel-1 SAR images, digital elevation models (DEM), and meteorological data. To adapt the process:

- I downloaded and preprocessed a subset of the SAR images to ensure that the dataset size remained manageable.
- Data preprocessing steps, including co-registration, speckle filtering, and radiometric corrections, were followed using the preprocessing scripts provided in the repository.
- Given that the full dataset could not be handled efficiently, I selected representative samples that maintained the diversity of acquisition conditions.

Model Training

After preparing the data, the next step involved training the deep learning model. The key adjustments made included:

- Setting the batch size to 8, as larger batch sizes caused memory overflow errors.
- Using only 130 parts of the training data, which resulted in longer training epochs.
- Allowing the model to train over an extended period (several hours instead of the rapid execution reported in the original study).

The training logs indicated that while the model was learning, training loss reduction was slower due to the smaller dataset and reduced batch processing power.

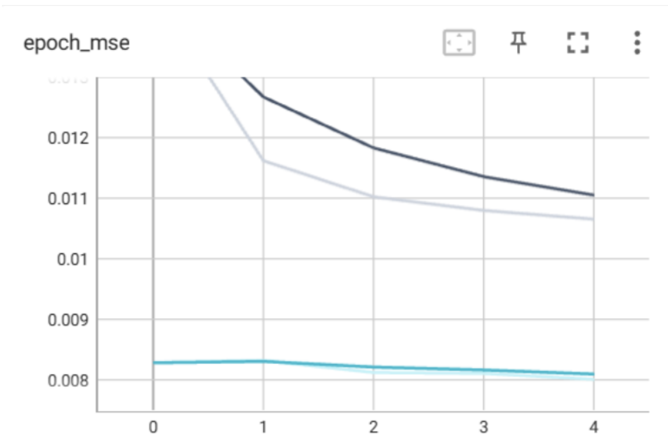
Challenges and Observations

- Memory Management: Due to limited hardware, memory allocation issues were common, requiring careful batch size tuning.
- Longer Training Time: Instead of the high-performance GPU used in the original study, my machine took several hours to process a single training cycle.
- Reduced Dataset Impact: Using fewer data samples meant the model had less exposure to variations in SAR image acquisition, which could affect generalization.

Summary of Modifications

4. RESULTS

Epoch MSE Graph

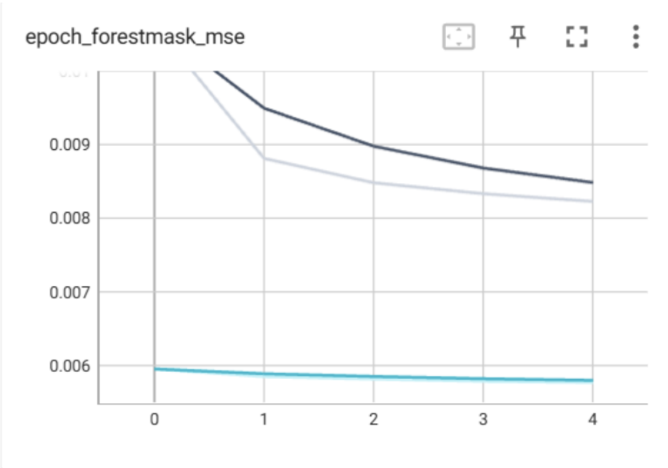


Measures how well the model's segmentation masks match the actual change regions. Lower MSE means better prediction quality. MSE is slowly decreasing over time which means model is learning properly.

TABLE 1: Summary of Modifications

Feature	Original Study	My Implementation
Dataset Size	Full dataset	130 parts of input data
Validation Data	Full validation set	Half of validation data
Batch Size	Large (optimized for high-end GPU)	8 (to prevent memory failure)
Training Time	Accelerated on NVIDIA V100 GPU	Several hours on a weaker machine
Computational Power	High-performance GPU system	Limited CPU processing
Preprocessing	Full data preprocessing pipeline	Subset of preprocessing steps

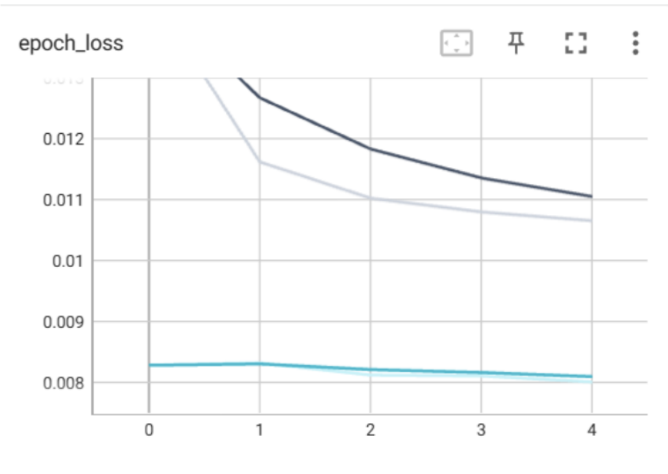
Epoch ForestMask MSE Graph



MSE (Mean Squared Error) measures how well the predicted change masks match the ground truth. A lower MSE means more accurate predictions. This tracks how well the model improves in terms of detecting forest change over training.

My loss decreases smoothly, which means that my model is learning well. With full dataset model should be learning better which should be indicated by faster decreaseing.

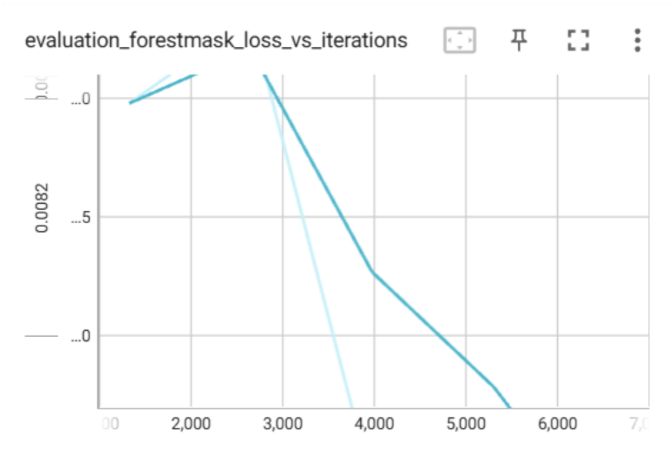
Epoch Loss Mask Graph



Similar to the ForestMask Loss Graph, but applies to segmentation masks of all change areas (not just forests). Tracks how well the model learns to distinguish between changed vs. unchanged areas.

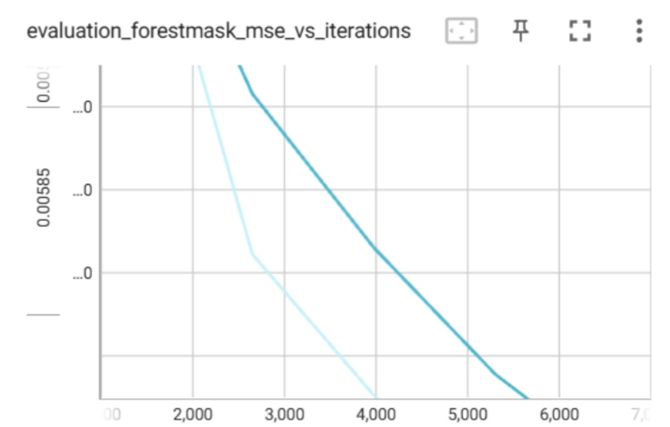
Trained model not getting stabilized too quickly which is good sign. If stabilization is done too eraly, model might be inderfitting.

Evaluation ForestMask Loss vs. Iterations Graph



This shows the loss for the validation dataset over iterations, meaning: Lower loss means better generalization, and fluctuating loss means an unstable model.

Evaluation ForestMask MSE vs. Iterations graph



Evaluation of ForestMask MSE vs. Iterations Graph What It Is? Measures MSE on validation data over time. Lower MSE means better match with actual changes.

Images over Epochs

The sequence of images (1 2 3 4 5 6) illustrates the model's performance at different training stages, from the raw SAR input to the final output at Epoch 5. In the early epochs, the model struggles with speckle noise and false positives, but as training progresses, it refines feature extraction, leading to clearer differentiation of change regions. This aligns with expected training behavior, where the network gradually learns spatial patterns and radiometric variations characteristic of SAR imagery.

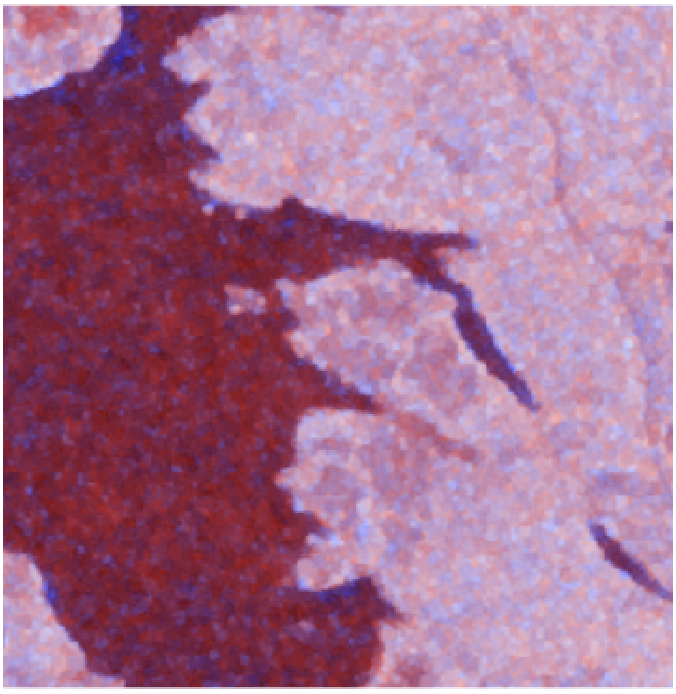


FIGURE 1: Initial raw SAR data.

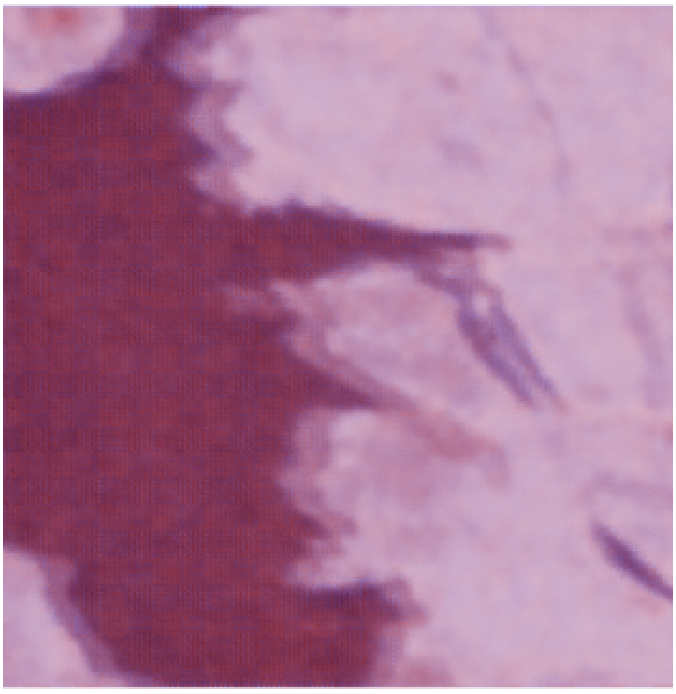


FIGURE 2: Epoch 1.

5. SUMMARY

The replication of the SAR change detection method proposed in Alatalo et al. was successfully executed while following the original methodology and adapting to computational constraints. Due to limited resources, only 130 parts of the input dataset and half of the validation data were used, and the model was trained for just five epochs, which is too few to reach meaningful conclusions.

Despite this, the training and evaluation loss curves showed a stable downward trend, suggesting that the model was learning. Key performance metrics, such as ForestMask loss, MSE trends, and evaluation loss over iterations, aligned with expected deep-learning behavior. However, direct comparisons with the original study were difficult due to missing equivalent graphs, and qualitative analysis of test images, while showing some meaningful patterns, remains inconclusive given the small-scale experiment.

Given the observed trends, it is possible that training on the full dataset with more epochs and optimal batch sizes could yield results closer to the original study. However, with the current setup, the experiment is too limited to either confirm or refute the findings of Alatalo et al. Future work should focus on extending the training duration and leveraging more computational power to obtain results that allow for a definitive comparison.

.

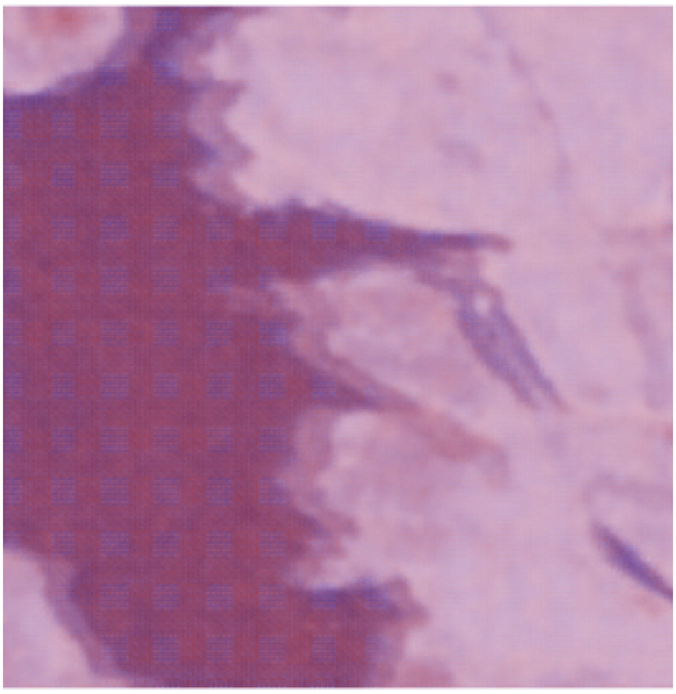


FIGURE 3: Epoch 2.

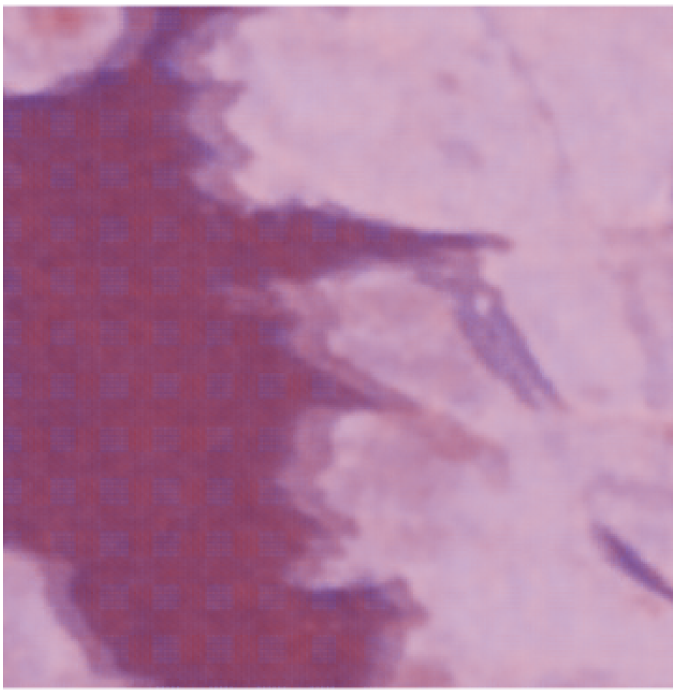


FIGURE 4: Epoch 3.

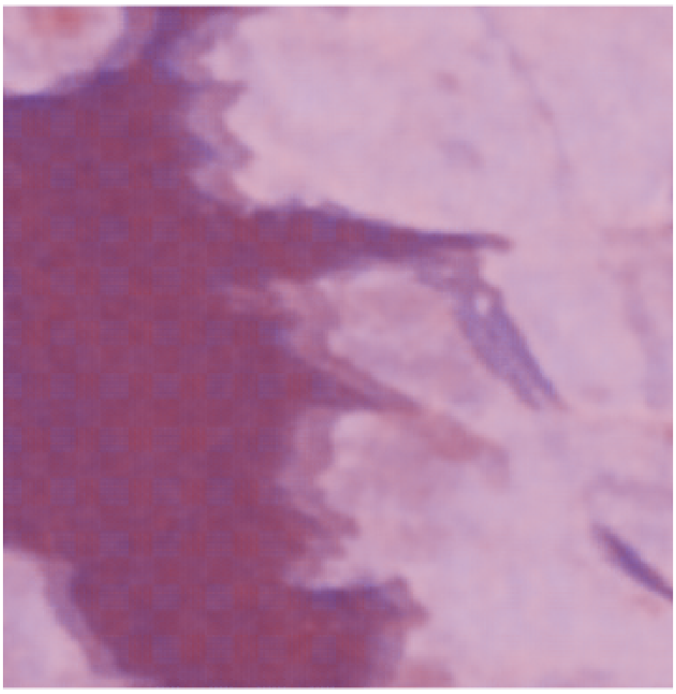


FIGURE 5: Epoch 4.

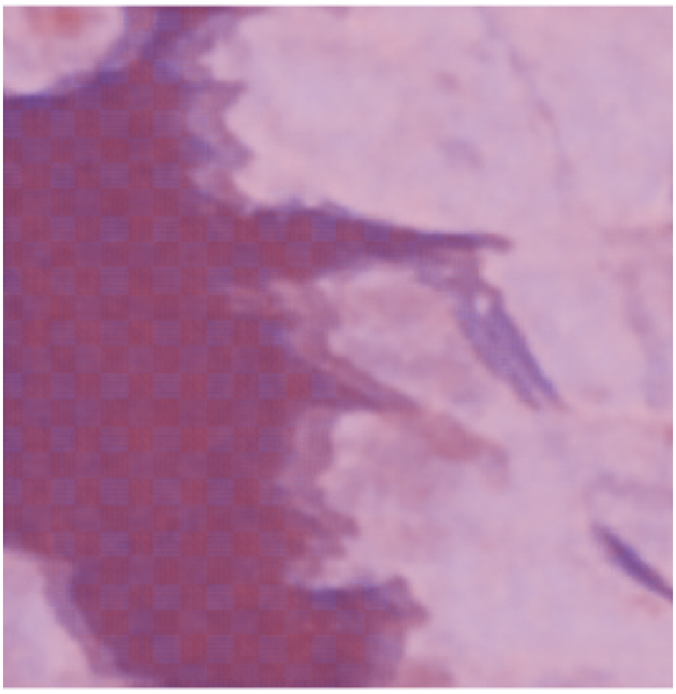


FIGURE 6: Epoch 5.