Image cytometry outlier detection

Machine learning solutions for anomaly detection in cell research.

Thibaud Collyn, Ben De Meurichy

[a]*Ghent University, Krijgslaan 281, Gent, B-9000, Belgium*

Abstract

As cytometry machines generate increasingly large datasets, the number of faulty/incorrect scans also increases. Therefore the demand for automated outlier detection has become critical for maintaining the quality and reliability of data.
In this study, we explore machine learning techniques that classify outliers directly from sample images.

We evaluated three distinct models to measure their effectiveness in correctly picking out the anomalies from the dataset: a Variational Auto-Encoder (VAE) using reconstruction error thresholds, a Self-Organizing Map (SOM) leveraging the distance to the best matching unit (BMU) and a classic binary Convolutional Neural Net (CNN) classifier.
To further enhance the detection performance, we propose an ensemble model that uses the three methods in a majority vote structure.
Having three distinct ways to decide the outliers will improve the rate of catching outliers in our dataset. This increases dataset reliability which supports research down the line.

Keywords: anomaly detection, image cytometry, clustering, som, vae, cnn

1. Introduction

When working with the rapidly growing datasets produced by state-of-the-art cytometry machines, it is essential to account for potential errors that can occur during the high-speed processing of cells.

Ensuring that the cells pass through the scan head at the right flow rate is challenging. This is why a lot of errors are introduced at this stage of the process.
These errors range from partial scans to multiple cells clumping together, resulting in incorrect feature data that render the samples unusable for further research.

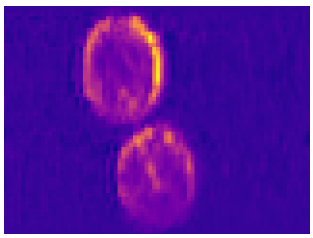

Figure 1: two cells

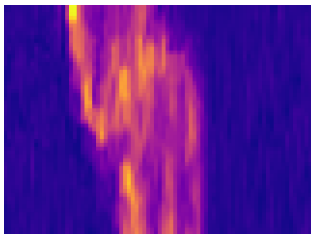

Figure 2: Partial cell

Although these faulty samples are easily detected with a quick visual inspection of the image cytometry scan, the sheer volume of data makes manual inspection impractical.

To address this challenge, we can take advantage of the extensive research available in the field of outlier detection. Approaches in this field range from statistical models applied to feature data to more advanced machine learning methods that operate directly on image data.
Given our background in computer science, we opted for a machine learning approach focusing on image data.

Our goal was to automate and imitate the visual inspection of the cells used to identify faulty samples.
To achieve this, we implemented a majority-vote model, simulating an environment in which multiple experts independently classify samples and compare their results.
This approach attempts to enhance robustness and reliability, mimicking real-world expert consensus while addressing the limitations of individual models.

In section 2 we will briefly discuss the data used in training our models. The rest of the section is dedicated to explaining the design of the three separate models that we used in the majority vote model. Section 3 will give a brief overview of the results of the separate models and the result of the majority vote model.

2. Method

2.1. Data analysis

Before addressing the outlier detection problem, it is important to first understand the data.
This is why we performed an analysis on the ratio of inliers to

outliers and the provided tiff image format.

2.1.1. Ratio

Having gathered information on the number of outliers in our dataset, we found that about ≈ 30% (figure 3) of our samples were anomalies. This indicates a notable imbalance in our dataset, but not one that requires us to take drastic measures. This information influenced our model decision, which is why we chose the VAE and SOM model that don't rely on outlier data for their training step.

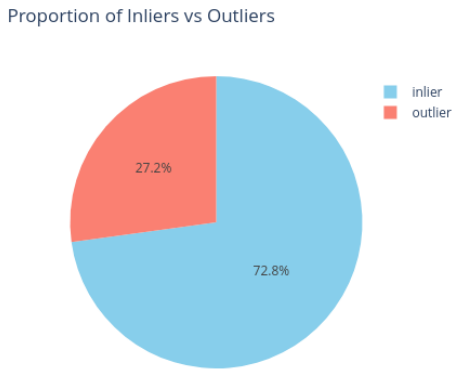

Figure 3: Ratio outliers to inliers

2.1.2. Format

Before implementing our image preprocessing steps, we discovered that the images consist of eight layers of 32-bit floating-point matrices. Many of the images have varying dimensions, and some layers contain mainly noise. After performing an edge analysis on all the scans using the 'sobel' function from 'skimage', we concluded that most of the useful data is found in layers 1-4. Taking the mean of these layers simplifies the input data for the models and helps filter out much of the noise.

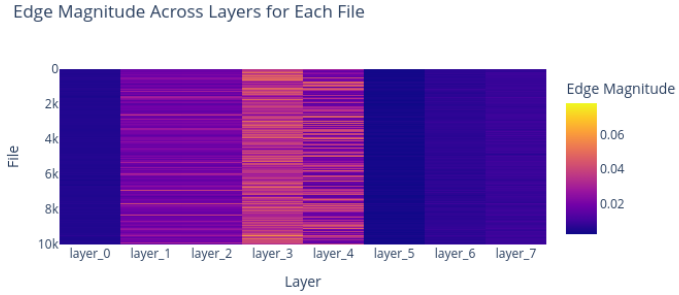

Figure 4: Edge data for each layer

2.2. Models

Given the low resolution and limited feature complexity of the dataset, we adapted our models from existing implementations designed for the MNIST handwritten digit dataset.

2.2.1. Convolutional Neural Net (CNN)

Due to the poor balancing of the dataset as discussed in section 2.1 the first thing we researched was designing a CNN for single class recognition. We started experimenting with a model based on a paper from Oza and Patel (2019) that proposed a solution to one-class classification problems. Since the amount of outlier data was limited, this seemed like a promising start for our research. However, this approach turned out to be too complex for the data. It involved feature extraction using a pre-trained VGG16 model which did not produce any meaningful features, likely due to our dataset consisting of simple greyscale images. This approach also suggested generating pseudo negative data to represent the minority class based on these extracted features, which, given the poor feature extraction, did not provide any meaningful results.

The poor results of our first implementation made us explore a simpler approach to the model. The feature extraction of the data was vastly simplified in this model and consists of 3 simple steps. Firstly, the images all get the same dimensions(60 by 80). Secondly a mean is calculated from the second, third and fourth layers of the '.tiff' image files. Finally, the images are standardized by making use of the tensorflow.keras.image library.
The second important part of preprocessing is handling the unbalanced dataset. In this version of the model the minority class has been enriched by new data generated by SMOTE data generation. All outlier data is used to generate the SMOTE images, but we made sure that any test data was removed from the training set after this data generation.

The proposed Convolutional Neural Network (CNN) architecture is designed for binary classification(outlier or non-outlier), comprising a series of convolutional, pooling, and fully connected layers. The input layer accepts images of dimensions (60,80,1)(width,height,channels), where *width*, *height*, and *channels* correspond to the spatial and color dimensions of the input data. Since the images are greyscale the color dimension is 1. The first convolutional layer applies 32 filters of size 3×33×3 with ReLU activation, followed by a 2×22×2 max-pooling layer for spatial downsampling. A second convolutional layer with 64 filters of size 3×33×3, also with ReLU activation, is appended, followed by another 2×22×2 max-pooling layer. The output is then flattened into a one-dimensional vector, which is passed through a fully connected layer with 64 neurons and ReLU activation.
Finally, the network concludes with a single-neuron output layer employing a sigmoid activation function, suitable for binary classification. The model is trained using the Adam optimizer with a binary cross-entropy loss function.

2.2.2. *Self Organizing Map (SOM)*

Clustering-based methods are effective for anomaly detection, as they allow for modelling the structure of normal data and identifying deviations. In this study, a Self-Organizing Map (SOM) was trained exclusively on inliers to learn the topological structure of normal data in the feature space. This approach enables the detection of outliers as data points that significantly deviate from the learned patterns.

The SOM was implemented with a grid size of 100×100 and an input dimension equal to the number of pixels in an image. Training was carried out using a Gaussian neighbourhood function and a learning rate that decayed linearly over 1000 iterations.

A threshold for outlier detection was derived from the BMU distances of a separate validation set. Data points with BMU distances that exceed this threshold are classified as outliers.
The hyperparameter for the grid size is optimized using Optuna.

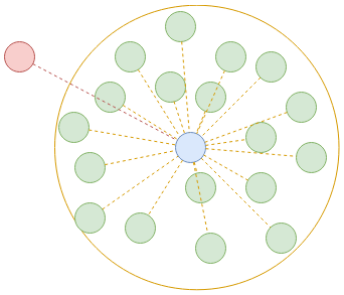

Figure 5: Visualization idea

2.2.3. *Variational Auto-Encoder (VAE)*

We further explored outlier classification strategies and found Variational Auto-Encoders (VAEs) to be an interesting solution. Although the approach of Burgess et al. Burgess et al. (2024) seemed similar to what we wanted to achieve, our problem seemed simpler. Since our cells are primarily round or oval, we went with a standard VAE rather than a more complex orientation-invariant version.
VAEs are generative models that learn a probabilistic representation of data, making them suitable for anomaly detection. To identify anomalies, we used an error threshold similar to the SOM model 2.2.2.

Our implementation builds upon the architecture introduced by Kingma and Welling Kingma and Welling (2022), initially developed for the MNIST dataset. We adapted their design to handle the higher dimensionality of our cytometry data.
The final model consists of five linear transformation layers in both the encoder and the decoder, with Swish Ramachandran et al. (2017) activation functions B.14. Swish was selected to enhance the reconstruction quality by enabling smoother gradient flow during backpropagation. This property allows the network to learn more detailed representations and achieve lower reconstruction error compared to traditional activation functions such as ReLU. The encoder progressively reduces the dimensionality, halving the hidden dimension at each layer until it reaches the latent space 1.
To avoid numerical issues caused by negative variances, we added a softplus layer for the latent space variance, ensuring all variances remain positive, and a sigmoid layer at the decoder's output to constrain values within a valid range.
To train the VAE, we used the AdamW optimizer Loshchilov and Hutter (2019) to minimize the combined reconstruction and KL divergence loss. AdamW extends the Adam optimizer by decoupling weight decay from gradient updates, which improves stability and generalization during training. This helped balance reconstruction accuracy with regularization to avoid overfitting.

input dim	4800
hidden dim	4000
latent dim	320

Table 1: VAE dimensions

After reaching a working implementation we tried optimising the model using Optuna. The following parameters were optimized:

- **Batch size**: Chosen from $[\frac{training}{10}, \frac{training}{5}, \frac{training}{2}]$
- **Hidden dimension**: Tuned within [200,4000] with a step size of 200.
- **Latent dimension**: Optimized in the range [10,400] with a step size of 10.
- **Epochs**: Explored in [50,500] with a step size of 50.

The objective function minimized the validation loss, and Optuna's pruning mechanism was used to terminate the underperforming trials early. This approach streamlined the optimization process and improved both performance and generalization.

Our implemented model is able to effectively reproduce inlier samples. The images of the reconstructions6 clearly provide a visual distinction between the outliers and inliers.

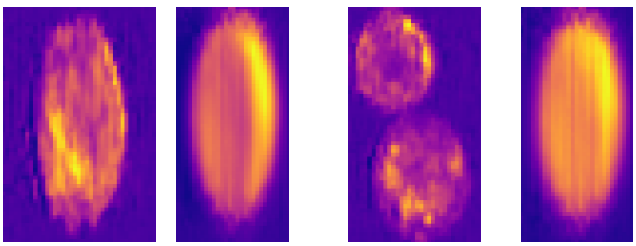

Figure 6: original sample and reconstructions of inlier(left) and outlier(right)

Initially, binary cross-entropy (BCE) between the sample and reconstruction was used as the error metric for classifying samples from the VAE, but it performed poorly. This is likely because the primary difference in reconstructions lies in the shape rather than in the exact pixel values, as the VAE introduces noise. To better capture structural differences, the Structural Similarity Index (SSIM) from skimage was used, leading to improved results.

3. Results

In this section, we will show the results of our models. We show the performance of the 3 individual models as well as the majority-vote model. All tests of our models used the exact same test set, but the division of the training data(train and validation split, if outlier data were actually used for training, ...) can differ from model to model. The data split is 20% test data and 80% training data. The train-test split was performed prior to any further data processing to prevent any data leakage into the models.

3.1. CNN results

The CNN has a very good result with an accuracy of 95.09%. Of the 4.91% of the wrongly predicted samples, only 0.65% were false negatives. For reducing a given set to only outliers and possibly a few inliers for manual classification, this model proves to be very good. Another noteworthy result achieved after testing is that both the training and evaluation time of the CNN turned out to be quite a bit faster than the VAE and the SOM.

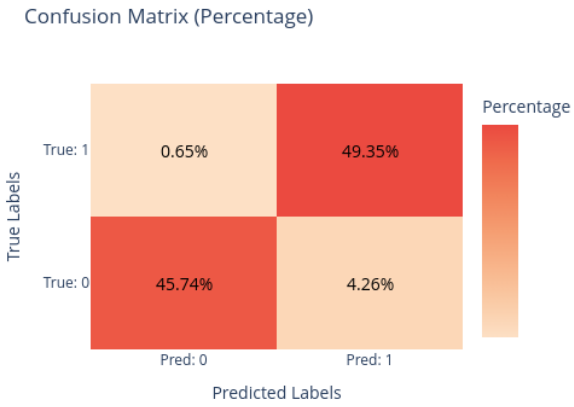

Figure 7: Confusion matrix for the CNN model

3.2. SOM results

Our clustering approach showed moderate accuracy at ≈ 82% but misclassified a significant number of samples (Figure 8). The model metrics indicate that when the SOM identifies a sample as an outlier, it is quite likely to be correct. However, the SOM fails to detect many true outliers.

Further analysis revealed that the SOM performs well in identifying outliers caused by multiple cells, but struggles with partial scans. To address this limitation, we think that using metaclustering techniques, similar to those used in FlowSOM Van Gassen et al. (2015), could reduce the noise in the samples and better define the regions containing inliers.

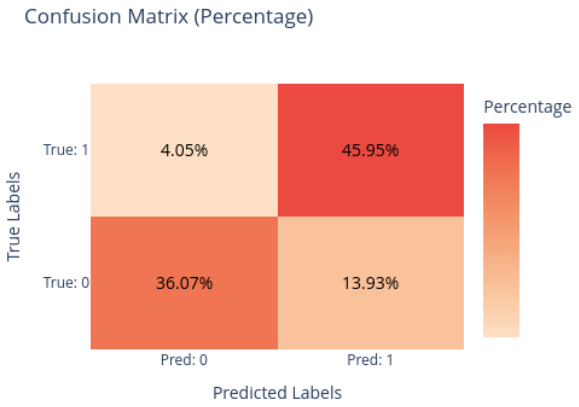

Figure 8: Confusion matrix for the SOM model

3.3. VAE results

The performance of the VAE was somewhat underwhelming but not problematic. The model correctly classified 81% of the samples, although its sensitivity to anomalies was limited. While the VAE outperformed the SOM in the detection of outliers, as shown in Table C.2, its confidence in detected anomalies was lower.

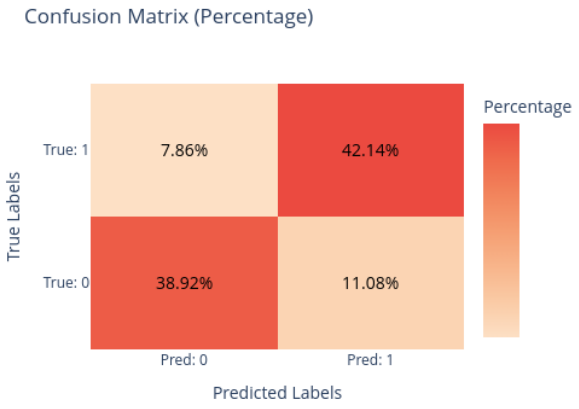

Figure 9: Confusion matrix for the VAE model

We initially thought that partial scans might be impacting performance, as some of them are challenging to classify manually (Figure **??**). However, this hypothesis was not supported by the data in Table C.3, which shows no strong

evidence that the model struggles with any specific type of outlier. Interestingly, multiple cell scans were misclassified slightly more often than partial scans.

We propose that using a more complex VAE with additional layers could enhance the network's ability to capture the nuanced features of inlier samples. Improved feature extraction may enable the model to reconstruct inliers more accurately, thereby reducing the reconstruction error for inliers and making anomalies more distinguishable.

3.4. Majority vote results

The majority vote model's performance aligns closely with the SOM and VAE, as these two models limit its accuracy. Although it slightly outperforms the SOM in detecting outliers (Table C.2), it struggles with prediction certainty, resulting in lower accuracy overall (Figure 9).

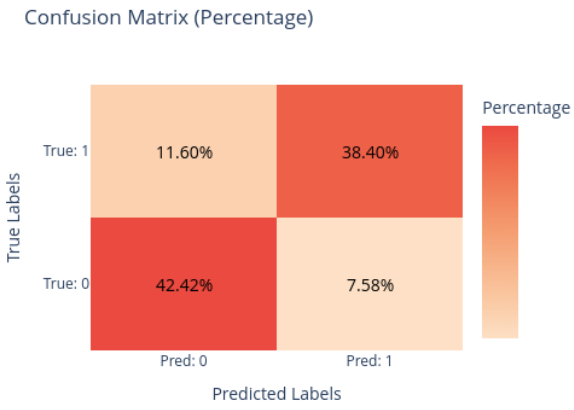

Figure 10: Confusion matrix for the Majority Vote model

4. Discussion

Our individual models have various levels of success.
The convolutional neural network is by far the best model of the three. The variational auto-encoder and self-organizing map both achieve a similar accuracy but are out-performed by the CNN. This large gap in performance between the models also explains the performance of the majority vote model. The majority vote model shows slightly worse metrics than the SOM and VAE.

The performance of our majority vote model is due to the VAE and SOM overshadowing the performance of the CNN which limits the models performance to one similar to the VAE and the SOM. We do believe that some further research and tinkering could achieve a better result for both the VAE and the SOM.

Exploring alternative approaches to classification could also address these limitations. For instance, classifying specific types of outliers individually, rather than using a binary inlier/outlier distinction, might enable the models to better differentiate between various outlier types. This more nuanced classification could improve the accuracy of identifying outliers, as well as provide insights into the nature of the deviations.

Future work could also investigate the integration of complementary data sources or modalities beyond image data, potentially enhancing the robustness of outlier detection in cytometry applications.

5. Conclusion

Our results may seem a little lacklustre but we do believe that a majority vote model shows promise. The VAE and SOM could be investigated further and could produce better results with some tweaking. On the other hand the three models that we implemented were all designed to process image data of the cells. Implementing a majority vote model that consists of models that make predictions on different metrics could combine the strengths of these different metrics.

What we can conclude is that detecting outliers in cytometry data solely on the basis of the images is a viable option and actually works very well. Future work could focus on refining the VAE and SOM and integrating them with the CNN in a more balanced majority vote framework to develop a powerful and reliable solution for outlier detection.

Our research highlights the viability of machine learning for this application but also underscores the need for further refinement to achieve optimal results. Leveraging broader advances in machine learning research could accelerate progress in addressing the unique challenges of outlier detection in cytometry.

Disclaimers

- ChatGPT was used to check the grammar of this paper.
- Copilot was on in our ide for basic code completion.
- The models are trained on the joltik cluster of the UGent hpc.
- The data set was a proprietary labeled dataset containing the ‘.tiff‘ files.

References

Burgess, J., Nirschl, J., Zanellati, M.C., Lozano, A., Cohen, S., Yeung-Levy, S., 2024. Orientation-invariant autoencoders learn robust representations for shape profiling of cells and organelles. Nature Communications 15. URL: https://api.semanticscholar.org/CorpusID:267397743.

Kingma, D.P., Welling, M., 2022. Auto-encoding variational bayes. URL: https://arxiv.org/abs/1312.6114, arXiv:1312.6114.

Loshchilov, I., Hutter, F., 2019. Decoupled weight decay regularization. URL: https://arxiv.org/abs/1711.05101, arXiv:1711.05101.

Oza, P., Patel, V.M., 2019. One-class convolutional neural network. VIU lab .

Ramachandran, P., Zoph, B., Le, Q.V., 2017. Swish: a self-gated activation function. arXiv: Neural and Evolutionary Computing URL: https://api.semanticscholar.org/CorpusID:196158220.

Van Gassen, S., Callebaut, B., Van Helden, M.J., Lambrecht, B.N., Demeester, P., Dhaene, T., Saeys, Y., 2015. Flowsom: Using self-organizing maps for visualization and interpretation of cytometry data. Cytometry Part A 87, 636–645. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cyto.a.22625, doi:https://doi.org/10.1002/cyto.a.22625, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cyto.a.22625.

Appendix A. data

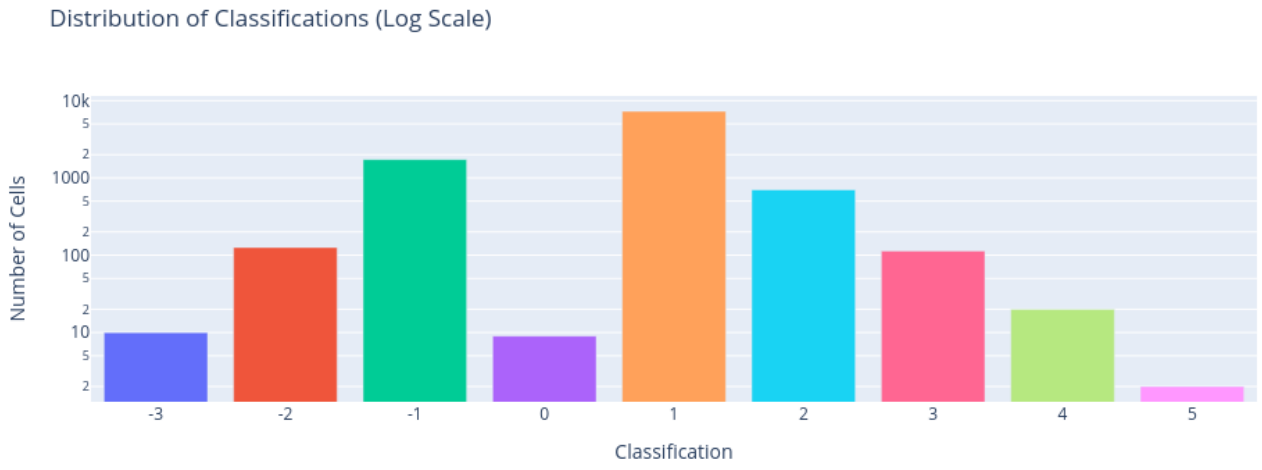

Figure A.11: Amount of samples per class

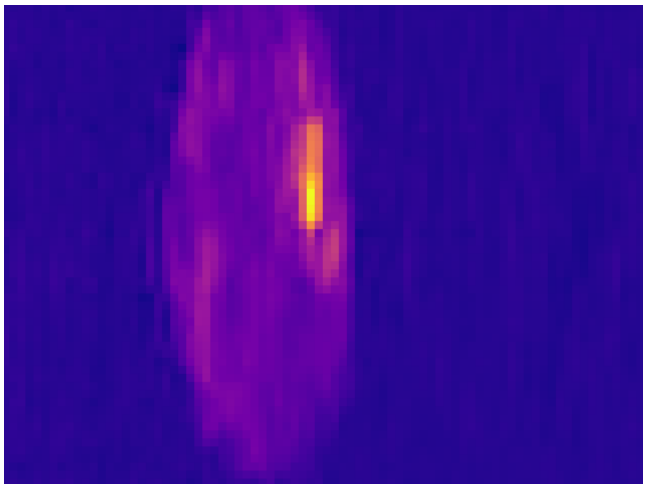

Figure A.12: inlier

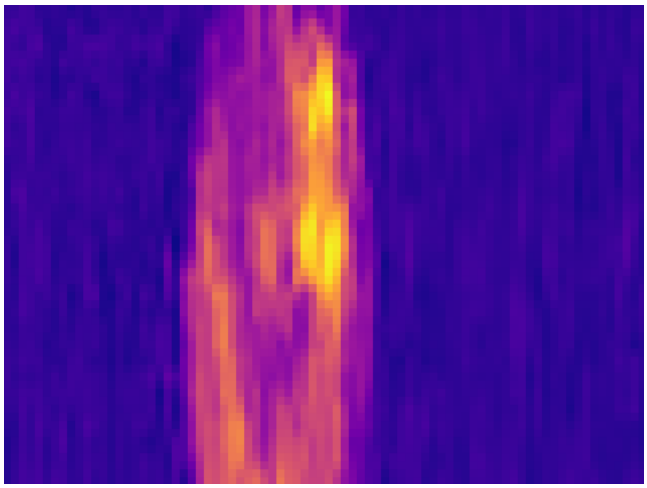

Figure A.13: partial scan

Appendix B. Models

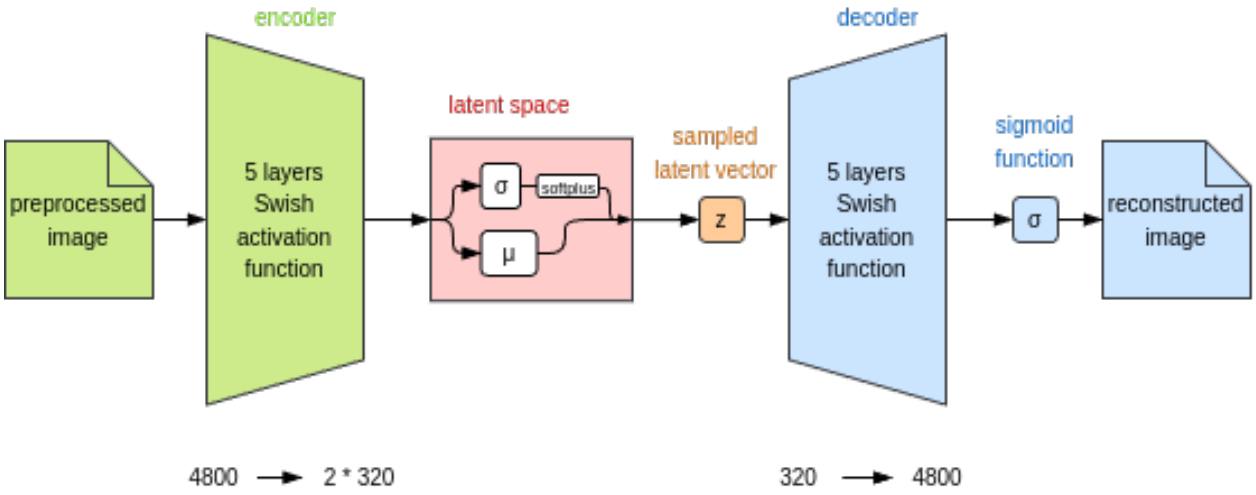

Figure B.14: Architecture of VAE model

Appendix C. Results

	CNN	SOM	VAE	Majority Vote
precision	0.9206	0.8990	0.8314	0.8328
recall	0.9870	0.7213	0.7817	0.7591
accuracy	0.9503	0.8202	0.8116	0,8034
f1 score	0.9526	0.8005	0.8058	0.7943

Table C.2: Metrics for each model

	> 1	< 1
correct	0.7315	0.8029
wrong	0.2685	0.1971

Table C.3: Problem outliers VAE

	> 1	< 1
correct	0.8981	0.6468
wrong	0.1019	0.3532

Table C.4: Problem outliers SOM