Kvasir PolypNet - Polyp Classification Projcet

Nils Gämperli
Student of Computer Sciences
ZHAW
Zurich, Switzerland
gaempnil@students.zhaw.ch

Jan Dalgali
Student of Computer Sciences
ZHAW
Zurich, Switzerland
dalgajan@students.zhaw.ch

I. Introduction

Colorectal cancer is a major global health concern and is among the leading causes of cancer-related deaths. Early detection and removal of precancerous polyps via colonoscopy significantly reduce mortality. However, the accuracy of polyp detection during colonoscopy depends on the expertise of the endoscopist, and polyps can often be missed, especially smaller or flat ones. [1]

In recent years, deep learning models, particularly Convolutional Neural Networks (CNNs), have demonstrated impressive performance in image classification tasks, including medical imaging. Automated polyp classification tools based on CNNs can serve as valuable clinical decision support systems to assist in accurate and timely diagnosis. [2]

This project focuses on developing a CNN-based classifier to distinguish between normal-cecum images and polyp images using the Kvasir dataset. The main objectives were to implement an efficient CNN architecture using PyTorch, apply data augmentation and preprocessing techniques, optimize the model's hyperparameters, and evaluate the performance on a validation set.

A. Polyp

Polyps are abnormal growths of tissue that often develop within the body, most commonly in internal organs such as the colon, rectum, uterus, or nasal cavity. While many polyps are benign and harmless, certain types carry the risk of becoming cancerous if not detected and treated in time. Typically, polyps do not produce noticeable symptoms and are often discovered incidentally during routine screenings or the evaluation of other medical conditions. However, in some cases—particularly when polyps enlarge or begin to bleed—they can lead to visible symptoms. [3]

In Fig. 1 a image of a tissue with a polyp is shown. For comparison, Fig. 2 shows a healthy tissue.

Fig. 1: Tissue with a Polyp, by [4]

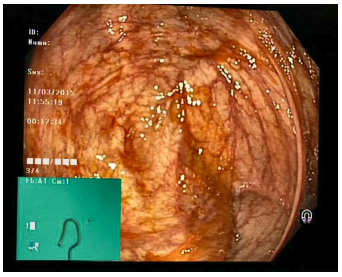

Fig. 2: Healthy tissue, by [4]

II. Materials

A. Dataset

For this project we used the Clahe Preprocessed Medical Imaging Dataset by Heartz Hacker.It is a modified version of the widely known Kvasir Dataset [5], focusing specifically on two image classes: normal-cecum and polyps. The dataset includes a total of 10′000 images, where 9000 are designated for training and 1000 for validation. The dataset was previously enhanced and tailored by Heartz Hacker for medical imaging classification through several preprocessing and structuring steps to improve model performance and implements advanced machine learning methods.

Key enhancements include:

- **Data Augmentation**: The dataset was enriched using common augmentation techniques such as flipping, rotation, and mirroring. These transformations increase data variability and help improve the generalization capabilities of classification models.
- **Incremental Learning Structure**: To support research in incremental learning, the dataset was divided into three separate training sets. This structure allows models to learn progressively from increasing amounts of data, mimicking real-world learning scenarios. This feature of the dataset was not used in our project, as we were not familiar with the concept of incremental learning.
- **CLAHE Preprocessing**: All images were processed using Contrast Limited Adaptive Histogram Equalization (CLAHE), a technique that enhances image contrast. CLAHE is especially beneficial in medical imaging, where subtle differences in texture and intensity are crucial for accurate diagnosis. For more about CLAHE refer to Section III.A.a

[6]

III. Methods

A. Preprocessing

As the given dataset was already highly preprocessed as described in Section II.A, we did not have to perform a lot of preprocessing. To scale all inputs of the dataset to the same input size, we calculated the mean of all images in the dataset. The mean over all images was 676 pixels in width and 650 pixels in height. Then rescaled the images before the input into our Deep Learning Model. With PyTorch transformers, this task can be done really simple, as the following lines show:

```
transform = transforms.Compose([
  transforms.Resize((676, 650))
```

a) *Clahe:*

Contrast-Limited Adaptive Histogram Equalization called CLAHE is a technique for image processing which can be used to increase the contrast of an image. To perform the process, CLAHE works in a small area of an image called a tile call, instead of the whole image. Then the CLAHE function is applied to every tile of the image and each tile increases in contrast.To avoid induced boundaries, the neighboring tiles are merged together. In homogenous areas the contrast can be limited to avoid amplifying a noise, which may be present in the image. In Fig. 3 a before and after is shown how the contrast of the image was increased. [7]

(a) Original Image without AHE

(b) Result Image with AHE

Fig. 3: Process of CLAHE, by [7]

B. Deep Learning

a) *Convolutional Neural Networks:*

Convolutional Neural Networks (CNN) are designed for a grid like imput structure, such as a two dimensional image, where each input represents one pixel of the input image. Images typically form a three dimensional input volume also called channels, where each dimension represents a RGB color.

Convolution:

A core component of CNNs is the convolution operation, which applies a grid of weights to different regions of the input image. This localized, sliding-window approach allows the network to detect patterns like edges, textures, or other features that depend on spatial arrangement. Because CNNs rely on spatial locality, they are particularly well-suited for analyzing visual data. The following image describes a convolutional operation. On the top left our input image is represented as a grid of numbers, where each number acts for a grey scale color. Over the image a grid structure like element, called a kernel slides and performs a convolution operation, where the value of the given pixel in the image is multiplied with the corresponding weight of the kernel. This weight of the kernel is also trained and adapted during the back propagation algorithm. This has the conclusion, that some pixels have a higher impact than others. [8]

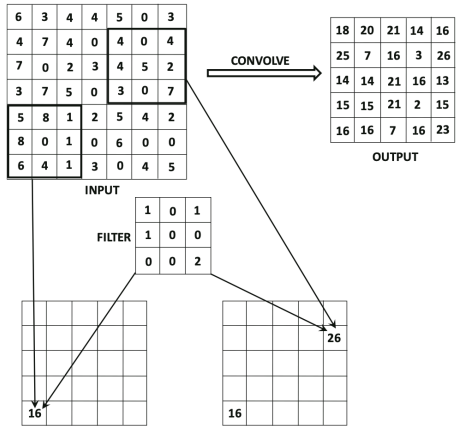

Fig. 4: Convolution by [8]

Fully Connected Layers:

The output of the convolutional operations is subsequently inputted into the initial of at least two fully connected layers (FCL), wherein the dimension of the first layer of the FCL corresponds to the output dimension of the convolutional output. Th4e output of the convolutional operations is then inputted into the first of at least two FCL, where the first layer of the FCL has the size of the output of the convolutional output. Two FCL are necessary to give the Network the ability to regress a non linearity, whereas one layer could only predict linear functions. [9]

The primary function of the FCL is to perform classification. The output layer of the FCL comprises a number of neurons corresponding to the number of classes in the classification task. In Fig. 5 an example of Fully Connected Layers is shown. Each circle represents a Neuron and the lines a connection, also called a weight. Each circle in the diagram represents a neuron, and the lines connecting them represent a weight between two neurons. These weights are subsequently trained using the backpropagation algorithm discussed in Section III.B.b

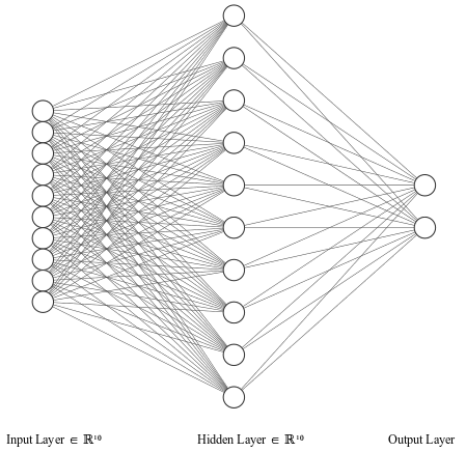

Fig. 5: Fully Connected Layers, visualized with: [10]

b) *Backpropagation Algorithm:*

The backpropagation algorithm is used in both convolutional and fully connected layers to adapt each weight. Within these layers, the weights are adjusted in response to the gradient to execute a process known as gradient descent. Mathematically, this can be represented as: $w_i(t+1) = w_i + \Delta_E w_i(t)$ where $\Delta_E w_i = -\frac{\partial E}{\partial w_i}$. ∂E represents the error at the network's output.In our context, this would signify the misclassification of an image, i.e., classifying an image as containing a polyp when none exists. [9]

C. Model Selection

For our classification task with the described dataset in Section II.A, we chose the following architecture for a CNN:

- The network begins with three convolutional blocks. Each block consists of a convolutional layer with a kernel size of 3×3 and padding of 1, followed by batch normalization, a ReLU activation function, and a 2×2 max-pooling layer to reduce the spatial dimensions. The number of filters starts at 32 and doubles with each subsequent block (32 → 64 → 128), allowing the network to learn increasingly abstract features at deeper layers.
- After the final convolutional block, we apply a global average pooling layer (AdaptiveAvgPool2d) to reduce the spatial dimensions of the feature maps to 1×1. This reduces the number of parameters and helps prevent overfitting while retaining the learned feature representations.
- The resulting 1-dimensional vector is then passed through a feedforward network consisting of three fully connected layers: the first with 64 units, which is the same as the output of the convolution layer, the second with 32 units, and the final output layer with a size corresponding to the number of target classes (in our case, 2). ReLU activation functions are used between the layers to introduce non-linearity.

[9]

In Fig. 6 a visualization of our used Network is shown. The selected architectural design should be a balance between simplicity and complexity, ensuring that it can effectively extract all the features of the image and accurately classify them. We performed additional tests using a more sophisticated CNN architecture. Given the unsatisfactory outcomes of the initial simple CNN, we subsequently increased the complexity of the model and reached the above described architecture.

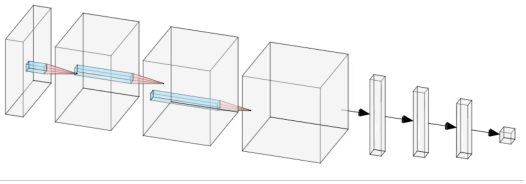

Fig. 6: Chosen CNN architecture, visualized with [10]

D. Model training

To train our convolutional neural network (CNN) for polyp classification, we implemented a robust and efficient pipeline using PyTorch. The training process involved the following steps:

1) Data Preparation and Loading
- loading the images from the file system into the memory
- resized to a consistent shape (676, 650)
- the dataset described in Section II.A already consits of a training set and a validation set. Therefore, we did not had to perform a train-validation-split.

2) Training configuration
- loss function: Cross Entropy Loss
- criterion: Adam, which combines training with momentum and RMSProp
- Training was done on GPU using mixed precision (`torch.amp`) to reduce memory usage and speed up training.
- Training Loop pseudo code:

```
For each (image, label) in dataset:
  move images and labels to GPU
  Zero the gradient
  outputs ← model(images)
  loss ← loss_function(outputs, labels)
  loss.backward() #perform
backpropagation
  optimizer.step #update model weights
```

3) Hyper parameter Tuning

We used Optuna [11], a hyper-parameter optimization framework, to tune key parameters:

- Learning rate (between 0.0015 and 0.0020)
- Number of epochs (9 or 10)
- Each trial trained the model with a different configuration. Optuna monitored the validation accuracy and pruned unpromising trials early to save resources.
- All training logs were saved to disk, and the best-performing configuration was printed at the end.

IV. Results

A. Visualizations

To further assess the model's performance and interpretability, we generated several visualizations, including confusion ma-

trices, ROC curves, and feature maps from the convolutional layers. These visualizations provide insights into the model's decision-making process and learned representations.

The confusion matrix, shown in Fig. 7, highlights the distribution of true positives, true negatives, false positives, and false negatives. Notably, the model exhibited fewer false positive than false negative (missed polyps), which is not preferable in a medical context where missing a polyp could have severe consequences.

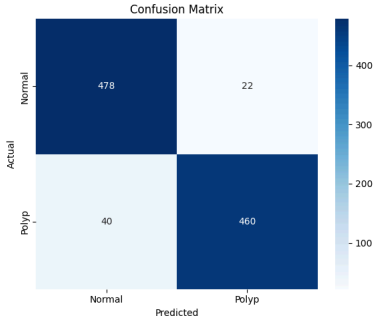

Fig. 7: Confusion matrix for the Kvasir PolypNet model on the validation set.

The ROC curve, depicted in Fig. 8, illustrates the trade-off between true positive rate and false positive rate, reinforcing the high ROC-AUC score of 0.97.

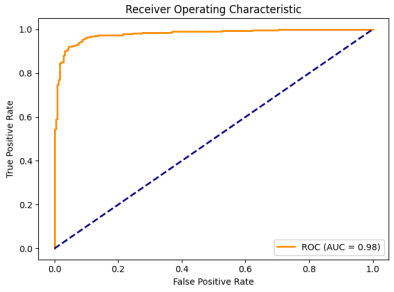

Fig. 8: ROC curve for the Kvasir PolypNet model, with an AUC of 0.97.

To understand the model's learned representations, we visualized feature maps from the three convolutional layers of the Kvasir PolypNet model, as shown in Fig. 9, Fig. 10, Fig. 11, and Fig. 12. These visualizations were generated using a sample polyp image from the validation set, allowing us to observe how the model processes the input at different stages of the network.

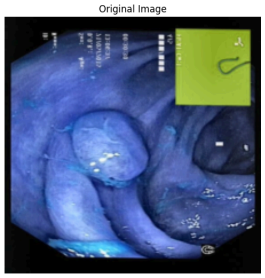

Fig. 9: Original polyp image used for generating feature maps.

The feature maps from the first convolutional layer, shown in Fig. 10, primarily capture low-level features such as edges, textures, and basic patterns. For example, some filters highlight the boundaries of the polyp and the surrounding tissue, while others emphasize textural differences in the image.

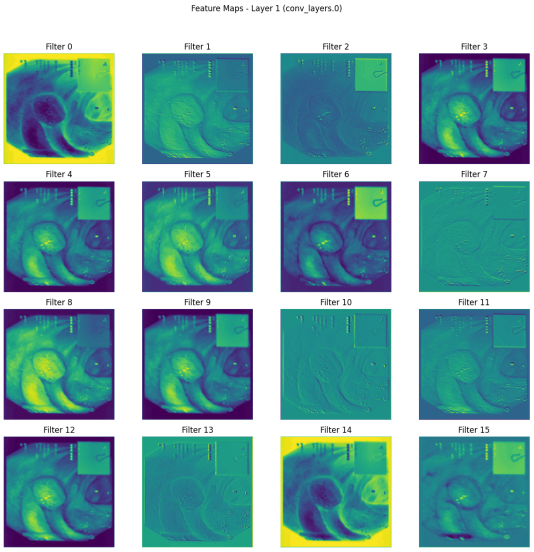

Fig. 10: Feature maps from the first convolutional layer of the Kvasir PolypNet model, showing low-level features like edges and textures.

In the second convolutional layer, depicted in Fig. 11, the feature maps begin to capture more complex patterns by combining the low-level features from the first layer. These maps show activations that correspond to larger structures, such as the overall shape of the polyp and its interaction with the surrounding tissue.

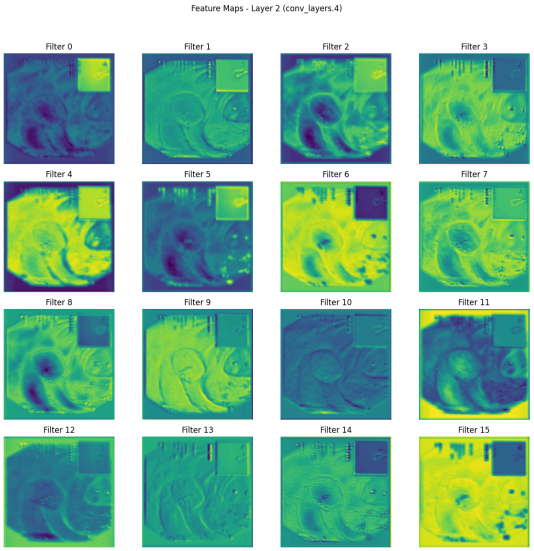

Fig. 11: Feature maps from the second convolutional layer of the Kvasir PolypNet model, showing intermediate-level features like combined edges and shapes.

The third convolutional layer, shown in Fig. 12, captures high-level features that are more abstract and specific to the polyp

class. These feature maps highlight regions that correspond to the polyp's shape and structure, demonstrating the model's ability to focus on clinically relevant areas for classification.

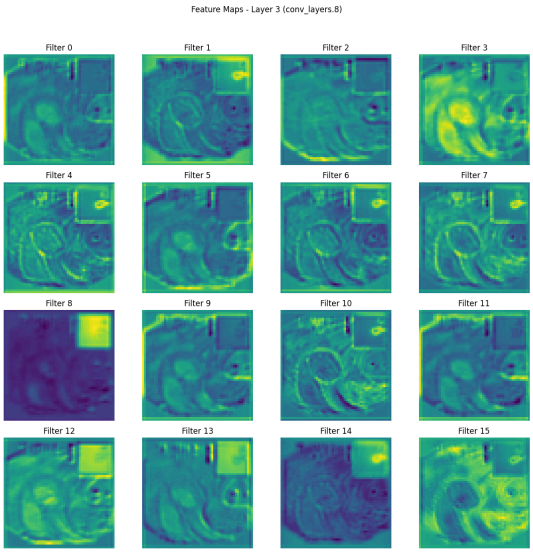

Fig. 12: Feature maps from the third convolutional layer of the Kvasir PolypNet model, showing high-level features like polyp shapes and structures.

Additionally, we generated a Grad-CAM visualization, shown in Fig. 13, to highlight the regions of the input image that most influence the model's prediction. The Grad-CAM heatmap confirms that the model focuses on the polyp region, aligning with the high-level features captured in the third convolutional layer.

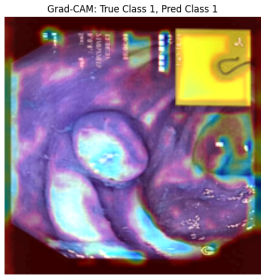

Fig. 13: Grad-CAM visualization for the Kvasir PolypNet model, highlighting the regions most influential for the polyp classification.

V. Discussion

The results of the Kvasir PolypNet project demonstrate that our CNN-based approach is a promising tool for automated polyp classification, aligning with the growing body of evidence supporting deep learning in medical imaging [8]. The high recall for the polyp class (93.0%) is particularly encouraging, as it suggests that the model can effectively identify most precancerous polyps, reducing the risk of missed diagnoses during colonoscopy screenings. This aligns with the primary clinical goal of improving early detection rates for colorectal cancer.

The model's precision (91.8% for polyps) and tendency to produce some false positives (normal-cecum classified as polyps) could be attributed to the inherent complexity of distinguishing subtle tissue variations, even with CLAHE-enhanced images. While false positives are less critical than false negatives in a screening context—since they would likely prompt further examination rather than missed treatment—they could increase procedural costs and patient anxiety. This trade-off suggests that the model is better suited as a decision-support tool for endoscopists rather than a standalone diagnostic system.

The architectural choices, such as the use of three convolutional blocks with increasing filter sizes and global average pooling, struck an effective balance between model complexity and generalization. The feature maps, shown in Fig. 10, Fig. 11, and Fig. 12, confirmed that the network learned hierarchical features relevant to polyp detection. The first layer (Fig. 10) captured low-level features like edges and textures, which are essential for identifying the boundaries of polyps and surrounding tissue. The second layer (Fig. 11) combined these features into more complex patterns, such as the overall shape of the polyp, while the third layer (Fig. 12) focused on high-level features, highlighting the polyp's structure and distinguishing it from normal tissue. The Grad-CAM visualization (Fig. 13) further validated the model's focus on the polyp region, showing that the high-level features in the third layer directly contribute to the final classification decision. However, the initial simpler CNN architecture tested during development underperformed, highlighting the need for sufficient depth to capture the dataset's complexity.

Preprocessing with CLAHE, as provided in the dataset [6], likely played a significant role in enhancing contrast and improving classification performance. This aligns with prior studies demonstrating CLAHE's efficacy in medical imaging [7]. Nonetheless, resizing all images to a uniform 667×650 resolution (as required by the model) may have introduced minor distortions, potentially affecting the model's ability to detect smaller or flatter polyps. Future work could explore multi-resolution inputs or attention mechanisms to address this limitation.

Hyperparameter tuning with Optuna was instrumental in achieving optimal performance, with the selected learning rate (0.0018) and epoch count (10) reflecting a practical compromise between convergence speed and overfitting prevention. The use of mixed-precision training on GPU further enhanced efficiency, making the pipeline scalable for larger datasets or real-time applications.

Limitations of this study include the reliance on a preprocessed dataset with only two classes, which may not fully represent the diversity of colonoscopy images encountered in clinical practice (e.g., varying lighting conditions, polyp sizes, or additional classes like inflammatory tissue). Additionally, the validation set, while sufficient for initial evaluation, is relatively small (1000 images), and testing on an external dataset would strengthen the model's generalizability claims.

Future improvements could involve incorporating transfer learning with pre-trained models (e.g., ResNet or EfficientNet) to leverage larger feature sets, experimenting with ensemble methods, or integrating real-time polyp localization alongside classification. Additionally, exploring attention mechanisms

or transformer-based architectures could enhance the model's ability to focus on subtle polyp features, potentially reducing false positives. These enhancements could further elevate the model's utility in clinical settings.

VI. Conclusion

The Kvasir PolypNet project successfully developed and evaluated a Convolutional Neural Network (CNN) for classifying polyp and normal-cecum images from the Clahe Preprocessed Medical Imaging Dataset. With an accuracy of 93.4%, a polyp recall of 93.0%, and an ROC-AUC of 0.97, the model demonstrates strong potential as a clinical decision-support tool for improving polyp detection during colonoscopies. The integration of CLAHE preprocessing, a carefully designed CNN architecture, and hyperparameter optimization via Optuna contributed to these robust results.

This study underscores the value of deep learning in medical imaging, particularly for tasks requiring high sensitivity, such as polyp identification. While the model excels at detecting polyps, its slight tendency toward false positives suggests it is best used in tandem with expert oversight rather than as an independent diagnostic system. The project also highlights the importance of preprocessing techniques like CLAHE and efficient training pipelines in achieving high performance with moderate computational resources.

Looking forward, expanding the dataset to include more diverse image classes, validating the model on external datasets, and exploring advanced architectures or real-time localization capabilities could further enhance its practical applicability. Ultimately, Kvasir PolypNet represents a step toward automated, reliable polyp classification, with the potential to reduce colorectal cancer mortality through improved early detection.

List of Figures

Fig. 1 Tissue with a Polyp, by [4] 1
Fig. 2 Healthy tissue, by [4] 1
Fig. 3 Process of CLAHE, by [7] 2
Fig. 4 Convolution by [8] 2
Fig. 5 Fully Connected Layers, visualized with: [10] ... 3
Fig. 6 Chosen CNN architecture, visualized with [10] .. 3
Fig. 7 Confusion matrix for the Kvasir PolypNet model on the validation set. 4
Fig. 8 ROC curve for the Kvasir PolypNet model, with an AUC of 0.97. 4
Fig. 9 Original polyp image used for generating feature maps. 4
Fig. 10 Feature maps from the first convolutional layer of the Kvasir PolypNet model, showing low-level features like edges and textures. 4
Fig. 11 Feature maps from the second convolutional layer of the Kvasir PolypNet model, showing intermediate-level features like combined edges and shapes. 4
Fig. 12 Feature maps from the third convolutional layer of the Kvasir PolypNet model, showing high-level features like polyp shapes and structures. 5
Fig. 13 Grad-CAM visualization for the Kvasir PolypNet model, highlighting the regions most influential for the polyp classification. 5

References

[1] "Colorectal Cancer." Apr. 03, 2025. Accessed: Apr. 10, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Colorectal_cancer&oldid=1283722603

[2] N. Juchler, "DSHEAL2025-ImageClassification_CNNs_Pytorch," Mar. 13, 2025. Accessed: Apr. 10, 2025. [Online]. Available: https://moodle.zhaw.ch/pluginfile.php/2072706/mod_resource/content/1/DSHEAL2025-ImageClassification_CNNs_Pytorch.pdf

[3] "What is a Polyp? What Are the Symptoms and Treatment Methods of Polyps? | Anadolu Sağlık Merkezi." Accessed: Apr. 05, 2025. [Online]. Available: https://www.anadolumedicalcenter.com/health-guide/what-is-a-polyp-what-are-the-symptoms-and-treatment-methods-of-polyps

[4] "Clahe Preprocessed Medical Imaging Dataset." Accessed: Apr. 08, 2025. [Online]. Available: https://www.kaggle.com/datasets/heartzhacker/n-clahe

[5] "Kvasir Dataset — kaggle.com."

[6] H. Hacker, "Clahe Preprocessed Medical Imaging Dataset — kaggle.com."

[7] P. Musa, F. A. Rafi, and M. Lamsani, "A Review: Contrast-Limited Adaptive Histogram Equalization (CLAHE) Methods to Help the Application of Face Recognition," in *2018 Third International Conference on Informatics and Computing (ICIC)*, Palembang, Indonesia: IEEE, Oct. 2018, pp. 1–6. doi: 10.1109/IAC.2018.8780492.

[8] C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*. Cham: Springer International Publishing, 2023. doi: 10.1007/978-3-031-29642-0.

[9] C. doc. RNDr. Iveta Mrázová, "Multi-layered Neural Networks: Analysis of Their Properties." Prague, Czech Republic, 2024.

[10] "NN SVG — alexlenail.me."

[11] "Optuna - A hyperparameter optimization framework — optuna.org."