

Title

Abstract

Table of Contents

1	Introduction and background	1
1.1	Background	1
1.2	Introduction	1
1.3	Structure of the Report	2
2	Literature review	3
2.1	Microtome and Microscope Selection	3
2.2	Deep Learning in Tissue Sectioning	3
3	Methodology and theory	4
3.1	Computer Vision - Image Segmentation	4
3.1.1	Edge Detection	4
3.1.2	Theresold Segmentation	5
3.2	Deep Learning	6
3.2.1	Convolutional Neural Networks (CNN)	6
3.2.2	Transfer Learning	7
4	Experimental work/ analytical investigation/ design	8
4.1	Data Collection	8
4.2	Data Labeling	9
4.3	Model 1: Original Images with a Simple CNN Network	11
4.4	Image Preprocessing Improvement	13
4.4.1	Edge Detection	13
4.4.2	Threshold Segmentation	16
4.4.3	Another Threshold Segmentation Method: Fingerprint Algorithm	18
4.5	Model 2: Preprocessed Images with a Simple CNN Network	18
4.6	Model 3: Original Images with Transfer Learning	20

4.7 Model Selection Summary	21
5 Presentation of Experimental or Analytical Results/Descriptions of Final Constructed Product	22
5.1 Validating Model Accuracy on a Test Set	22
5.2 Further Improvement of the Model (Changing Input Resolution)	23
5.3 Investigating the Optimal Cutting Angle for the Machine	24
5.4 Model Generalizability	24
6 Discussion and conclusions	27
6.1 Discussion of results	27
6.2 Future work	28
6.2.1 Enhancing Classification Methods	28
6.2.2 Performance Enhancement and Optimization	29
6.2.3 Optimization of the Sectioning Process	30
6.3 Conclusions	30
7 Project management, consideration of sustainability and health and safety	31
7.1 Subsection 5.1	31
7.2 Subsection 5.2	31
References	31

1 Introduction and background

1.1 *Background*

As the fundamental units of life, human research into cells and tissues has never ceased. Biological tissue sections, serving as crucial means for the direct observation of cellular morphology and structure, are essential for biomedical research and clinical diagnosis. A complete and usable tissue section is of great importance to researchers and physicians, as it provides vital information about cell structure, tissue morphology, and pathological changes. Within this, the quality of the section is of paramount importance.

Traditional manual sectioning methods are time-consuming and prone to variability, hence the emergence of automatic microtomes has provided a solution to these issues. For different biological tissues, varying cutting parameters can yield differing results, both positive and negative. Therefore, to enhance the utilization rate of biological sections and increase the production of high-quality specimens, determining optimal cutting parameters for specific tissues remains a goal.

Machine learning and deep learning have achieved significant success in the fields of computer vision and image processing. Machine learning is defined as a series of methods that can automatically detect patterns in data, which are then used to predict future outcomes or make decisions [1]. In this paper, we integrate advanced image analysis and machine learning techniques to identify section quality and then evaluate the quality of tissue samples under different sectioning parameters.

1.2 *Introduction*

Project Overview

This project aims to optimize the cutting parameters of biological tissue microtomes, which are crucial devices in biomedical research and clinical diagnostics. The objective is to enhance the precision and efficiency of tissue sample preparation by determining the optimal slicing conditions. By collecting tissue samples under various cutting parameters and conducting subsequent manual image classification, this study employs deep learning techniques to analyze and predict the most effective cutting parameters. This work not only promises to improve the quality of tissue samples for microscopic examination but also helps to simplify laboratory workflows, thereby advancing biological and medical sciences.

Objectives:

1. Collect a comprehensive dataset of tissue samples sliced under different parameters.
2. Employ artificial image classification to categorize the quality and characteristics of these samples.
3. Develop and train a deep learning model capable of assessing tissue sample quality.
4. Use the model's insights to determine the optimal cutting parameters for the tissue slicer.
5. Validate the model's predictions through empirical testing and refinement.

1.3 *Structure of the Report*

The project is structured into several chapters, each aimed at systematically exploring research background, methodologies, experimental work, results presentation, discussions and conclusions, as well as considerations for project management, sustainability, and health safety:

Introduction and Background - This chapter outlines the project's goals, objectives, and structural arrangements. It briefly introduces the motivation and necessity for the research, as well as the adopted technical protocols and standards.

Literature Review - Provides an in-depth discussion on biological tissue sections, image classification, and the application of deep learning in the preparation of biological samples. This section positions the current study within the context of existing research.

Methods and Theory - Detailed description of experimental methods, theoretical frameworks, and specific plans for data collection and processing.

Experimental Work/Analysis Investigation/Design - Describes in detail the experimental design, implementation, and analytical survey. It elaborates on the strategies and methods adopted to achieve the project's objectives.

Presentation of Experimental or Analysis Results/Final Constructed Product Description - This chapter presents experimental data, analysis results, or final design products, detailing the outcomes of experiments or designs.

Discussion and Conclusion - Analyzes the results, discussing their scientific significance and practical value. This chapter also provides conclusions from the research and suggests potential directions for future studies.

Project Management, Sustainability, and Health Safety Considerations - Discusses project management strategies, sustainability issues, and health safety measures to ensure efficient and safe conduct of the research work.

References - Lists all the referenced literature, supporting the research and providing a foundation for the study.

Assumptions and Technical Specifications

The project is based on several key assumptions and technical protocols, which include:

1. Consistency in the properties of tissue samples across different batches.
2. The reliability and accuracy of the biological tissue microtome and imaging equipment.
3. The adequacy of deep learning models in interpreting complex biological image data.

Technical specifications regarding tissue microtome settings, image classification standards, and deep learning architecture are detailed in the Methods and Theory section.

2 Literature review

This literature review explores the integration of technologies in biological tissue sectioning, with a particular focus on the application of image classification and deep learning in optimizing slicing parameters. It aims to highlight significant advancements, identify gaps in current methodologies, and lay the groundwork for the proposed project.

2.1 Microtome and Microscope Selection

In recent years, the advent of automatic microtomes has significantly simplified the sectioning process and improved the quality of sections.

Zimmermann, in the article "Improved reproducibility in preparing precision-cut liver tissue slices," advocates for the use of the new Leica vibratome to enhance the accuracy and reproducibility of tissue sections from rats, mice, and human tissues [2].

In this experiment, the HM355S microtome provided by Epredia is used for sectioning. This machine is a popular device for biological tissue sectioning research, and many experiments and papers have utilized this equipment for sectioning.

Elzbieta Klimuszko has used the HM355S microtome for sectioning teeth to investigate the calcium and magnesium content in dental enamel [3].

Andelko Hrzenjak also used the HM355S microtome for sectioning pathological endometrial tissues to study the mechanisms of endometrial carcinoma development [4].

Similarly, the choice of microscope is crucial. In this experiment, the VHX7000 microscope from Keyence is used for image acquisition. It is capable of capturing images of biological tissue sections (e.g., mouse prostate cells [5]), as well as inorganic materials (such as ceramics [6], glass [7]).

The experiments will employ the HM355s microtome and VHX7000 microscope for sectioning and image acquisition. This setup ensures that both equipment selection and technological application are optimally aligned to enhance the precision and efficiency of the tissue sectioning process, supporting the overall goals of the research project.

2.2 Deep Learning in Tissue Sectioning

The application of deep learning technologies in the biomedical field has achieved significant advancements. Deep learning models excel in tasks such as image classification, object detection, and segmentation, providing powerful tools for research and diagnostics in biomedical laboratories.

Lorena Guachi-Guachi proposed a method utilizing CNN networks to identify and refine tissue sections. This approach represents an innovative application of deep learning that can enhance the precision of tissue preparation and analysis [8].

In the book *Biomedical Texture Analysis*, Vincent Andrearzyk introduced a CNN architecture specifically designed for texture analysis, which significantly improves the accuracy of classifying biological

tissues compared to traditional architectures [9]. This development demonstrates the potential of deep learning to enhance the detailed analysis of tissue characteristics, which is crucial for accurate diagnostics and research.

Yan Xu suggested that features extracted from CNNs trained on the large natural image database, ImageNet, can be transferred to histopathological images of tissues. This provides a viable approach for implementing transfer learning, which can greatly enhance the efficiency of tissue image classification and analysis [10].

Based on the literature, deep learning technology holds broad prospects for application in image classification and analysis of tissue sections. By leveraging deep learning models, efficient identification and classification of tissue samples can be achieved, providing strong support for optimizing sectioning parameters.

This section underscores the transformative impact of deep learning on the field of tissue sectioning, promising significant improvements in the accuracy and utility of histological analyses.

3 Methodology and theory

3.1 Computer Vision - Image Segmentation

For the acquired image data, appropriate image preprocessing can be applied. Under the premise of maintaining the integrity and quality of images, certain processing can be implemented to highlight the features intended for computer recognition and, to some extent, remove irrelevant features and noise. This enhances the accuracy of subsequent deep learning models.

Image segmentation is a critical step in image processing, aiming to divide the image into several meaningful regions for further analysis and processing. In models focusing on the yield rate of biological tissues, it is necessary to segment the biological sections into biological tissue and paraffin areas, emphasizing the biological tissue parts.

Common image segmentation algorithms include edge detection and threshold segmentation.

3.1.1 Edge Detection

For biological tissue sections, a crucial indicator of quality is the clarity of the section's edges. The integrity and continuity of the slice edges can reflect whether there are quality issues with the sample.

There are numerous algorithms for edge detection, such as Sobel, Laplacian, and Canny operators [11].

The **Sobel operator** is a first-order differential operator that can be used to detect image edges [12]. Suppose there is a one-dimensional image $f(x)$, the relationship between its intensity and the pixel coordinate x can be represented as shown in Figure 1. It can be observed in Figure 3.1 that the slope is the largest around $x=2.2$, indicating that there is a sudden change in image intensity (an edge exists) near this point. Taking its derivative gives the first-order derivative $f'(x)$, as shown in Figure 3.2, where

the absolute value of the derivative is the largest. The Sobel operator uses this characteristic to detect edges.

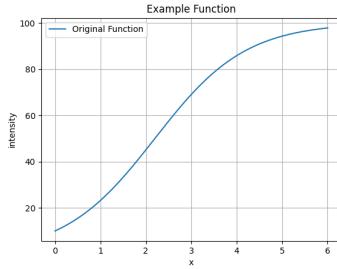


Figure 3.1 $f(x)$

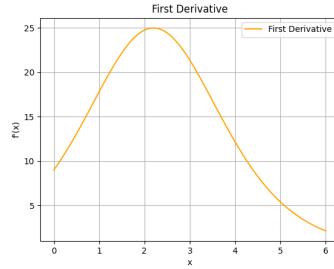


Figure 3.2 $f'(x)$

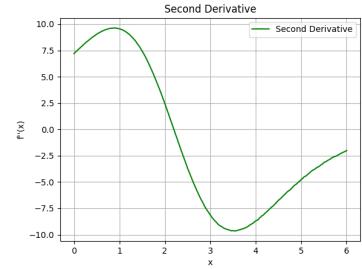


Figure 3.3 $f''(x)$

The **Laplacian operator** is a second-order differential operator that performs well in edge detection of images. It is derived by taking the derivative of the Sobel operator once more. In 2D images, the Laplacian operator is defined as follows:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (3.1)$$

As shown in the figure above, taking the derivative of the first-order derivative results in the second-order derivative $f''(x)$, as shown in Figure 3.3. It can be seen that around $x=2.2$, the second-order derivative is 0, which indicates that when the value of the Laplacian operator $\nabla^2 f$ is 0, there is a sudden change in image intensity, indicating the presence of an edge.

Canny Operator is a multi-stage differential operator that enhances the edge detection process by incorporating noise suppression, building on the initial computations similar to those used by the Sobel operator. Introduced by John F. Canny in 1986 [13], the Canny operator refines the results obtained from Sobel operator calculations through additional steps such as non-maximum suppression and hysteresis thresholding. These steps set thresholds to eliminate false edges from the image, resulting in more accurate edge detection.

In the chapter on *Experimental Work/Analytical Investigation/Design*, experiments will be conducted on the collected image data using these three edge detection algorithms—Sobel, Laplacian, and Canny—to compare their effectiveness. This comparative analysis will help in identifying the most suitable method for edge detection in the context of tissue sectioning, where the clarity and precision of edges are vital for quality assessment. The results will guide the selection of the optimal algorithm to be integrated into the image processing pipeline, enhancing the capability of the system to accurately segment and analyze biological tissue sections.

3.1.2 *Threshold Segmentation*

Apart from edge detection, another method employed is threshold segmentation. This technique divides the image pixels into two categories: those above a certain threshold and those below it. It is

particularly useful in situations where there is a significant grayscale difference between the target and the background in the image.

For specimens, a straightforward approach is to contrast the colors of the paraffin area and the biological tissue area (which is stained during preparation), and then separate them using threshold segmentation. Assuming the biological tissue is yellow and the paraffin is white, setting a threshold could isolate the white parts of the image, leaving behind the biological tissue.

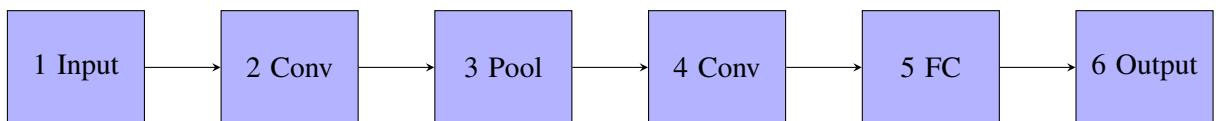
Additionally, there are more sophisticated methods of threshold segmentation, such as the Otsu method used for fingerprint extraction. Implementing this method can significantly enhance the segmentation of biological tissues. Yue Yaru and Zhu Jialin in "Algorithm of fingerprint extraction and implementation based on OpenCV" have proposed an improved Otsu-based fingerprint extraction algorithm using OpenCV. This algorithm excels particularly under conditions of uneven illumination and blurred images, providing accurate, simple, and fast fingerprint extraction [14].

Comparisons and experiments related to these segmentation techniques will be conducted in the *Experimental Work/Analytical Investigation/Design* chapter.

3.2 Deep Learning

3.2.1 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are a type of deep learning model that are particularly effective at processing image data. They automatically learn spatial hierarchies of features through a series of convolutional layers without the need for manual feature extraction. A typical CNN model includes layers such as convolutional layers, pooling layers, and fully connected layers [15]. The architecture of a CNN is illustrated below:



In this model:

Convolutional layers (Conv): These are the core layers of a CNN, responsible for feature extraction from images.

Pooling layers (Pool): These serve to reduce the dimensionality of the feature maps, thereby decreasing the computational load.

Fully connected layers (FC): These integrate the features extracted by the convolutional and pooling layers for classification or regression analysis, eventually leading to the output.

The typical method for training a CNN involves several key processes:

1. **Forward Propagation:** Input data passes through each layer of the network until it reaches the output layer.

2. **Loss Computation:** The network's output is compared to the actual labels using a loss function, such as cross-entropy loss, to calculate the difference.
3. **Backpropagation:** The gradient of the loss function with respect to the network weights is computed.
4. **Weight Update:** The network weights are updated using an optimization algorithm such as gradient descent or its variants like Adam or RMSprop, with the aim of minimizing the loss function.

Once trained, the CNN can be employed to predict labels for new, unseen images. The distinctive feature of CNNs is their ability to automatically and efficiently learn features at different levels of abstraction, making them highly effective for tasks involving complex image data, such as medical image analysis, where accuracy and detail are paramount.

3.2.2 Transfer Learning

Indeed, for complex image tasks, constructing a simple CNN network is often insufficient. In such cases, transfer learning becomes essential. Transfer learning is a machine learning method that accelerates the training process by transferring knowledge from a pre-trained model to a new task. The core idea of transfer learning is to leverage knowledge from the source domain to aid learning in the target domain.

For CNN models, there are several approaches to transfer learning, such as fine-tuning and feature extraction:

Fine-tuning involves adjusting the parameters of a pre-trained model to adapt it to a new task. This often includes retraining some of the convolutional layers along with the fully connected layers on the new data, which allows the model to fine-tune the features to the specific characteristics of the new dataset.

Feature extraction involves using a pre-trained model as a fixed feature extractor, where only the fully connected layers are trained on the new data. In this approach, the convolutional layers retain their learned weights and act solely to extract features, which are then used by the newly trained classifier layers to perform tasks specific to the new dataset.

Commonly used pre-trained models include VGG, Inception, and others. These models have been extensively trained on large datasets like ImageNet, where the weights of various layers in the model have been optimized and can be effectively used for transfer learning.

Table 2 shows the number of parameters in models such as VGG16, VGG19 [16], InceptionV3 [17], Xception [18], etc. These models have a large number of parameters, allowing them to accurately extract features from complex images. The capability to leverage these well-trained models enables researchers and practitioners to achieve high performance on specific tasks without the need to train an entire network from scratch, saving both time and resources while maintaining high accuracy.

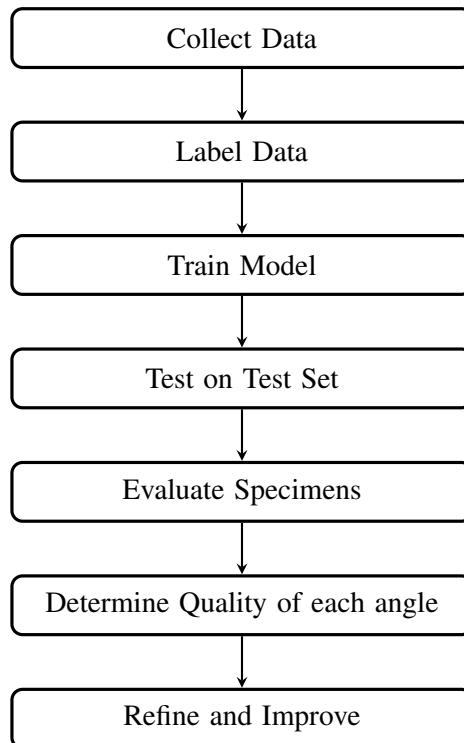
Table 3.1 Comparison of CNN Models

Model	VGG16	VGG19	InceptionV3	Xception
Number of Parameters	138,357,544	143,667,240	23,851,784	22,910,480

4 Experimental work/ analytical investigation/ design

Experimental workflow

The experiment workflow is outlined in the diagram below, detailing the sequential steps from data collection to iterative improvement of the model.



4.1 Data Collection

The first essential step for deep learning is data collection. In this experiment, pre-prepared paraffin-embedded tissue sections (fish ovary tissues) were used. These sections were placed on an HM355s automatic microtome, and slicing operations were performed according to different cutting angles as specified in the microtome's manual. The cutting data was recorded meticulously.

The schematic diagram of the microtome (taking a tooth as an example) is shown in Figure 4.2 [19].

In the experiment, the microtome was configured with the following parameters: the mode was set to continuous, the feed rate was 5.0, the trimming value was 25, the speed was set at 32, the water flow rate was 7.5, and the water temperature was approximately 36 degrees Celsius. The cutting angle was adjusted between 8 to 12 degrees.

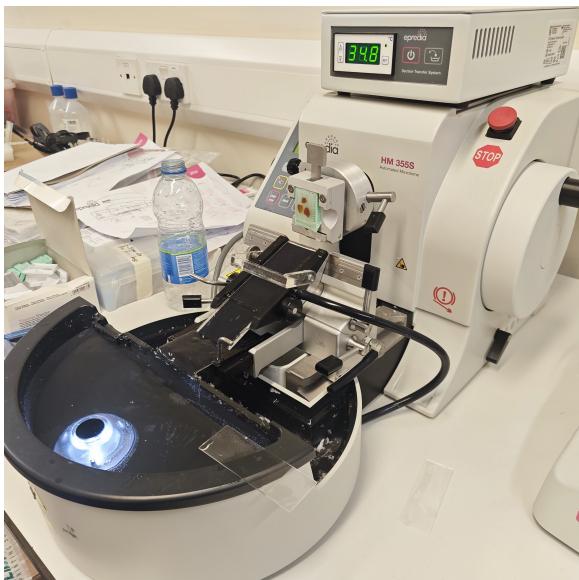


Figure 4.1 Microtome

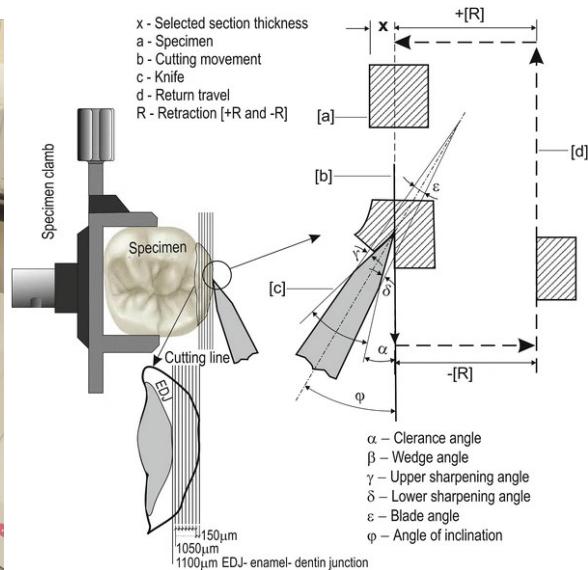


Figure 4.2 Working principle of the microtome

The biological tissues used for sectioning are shown in Figure 4.3. After sectioning, the different types of tissue sections were placed on slides as depicted in Figure 4.4. Once dried, these slides were transferred under a VHX7000 microscope for imaging. Each sample was photographed under the microscope to capture electronic image data, as shown in Figure 4.5.



Figure 4.3 Biological tis-
sues



Figure 4.4 Collecting samples

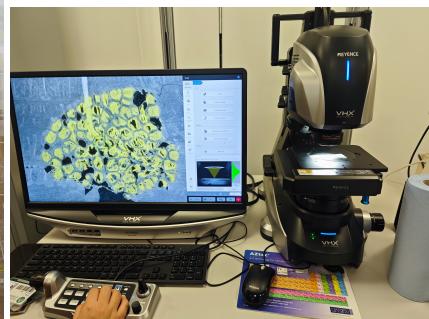


Figure 4.5 Microscope

Based on this, several hundred images were obtained, each with a resolution of 2880*2160. An example of the samples is shown in Figure 4.6.

4.2 Data Labeling

For this experiment, the dataset is re-labeled based on the quality of the tissue sections. Overall, the quality of the biological tissues is categorized into two primary classes: normal and bad. Further analysis of the collected data revealed common flaws - the presence of vertical or horizontal white creases on the sections, which clearly indicate unusable slices. Given the distinct nature of these flaws,

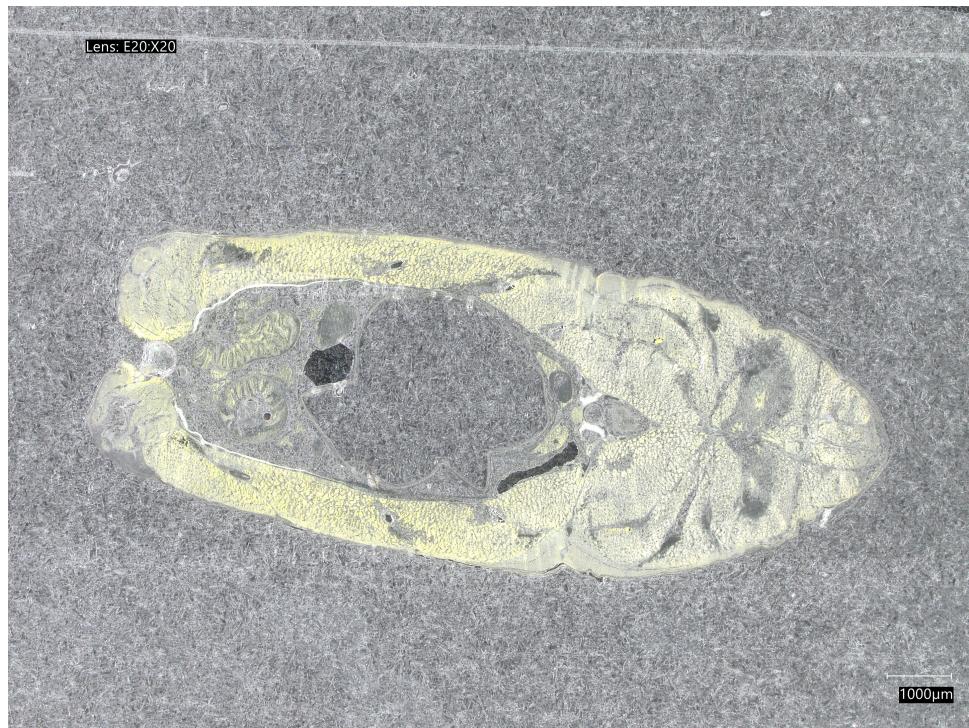


Figure 4.6 Sample with a cutting angle of 9.5 degrees

they are classified into two additional specific categories: **horizontal line**(Figure 4.7) and **vertical line**(Figure 4.8).

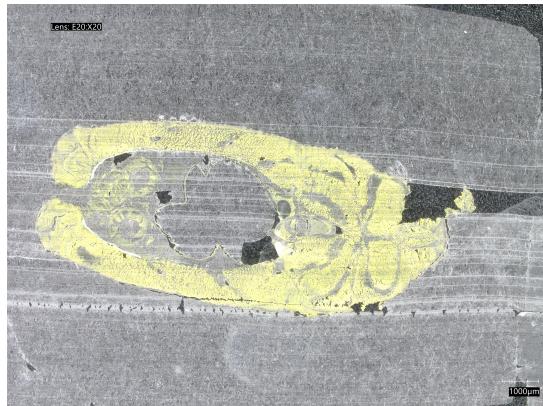


Figure 4.7 horizontal line

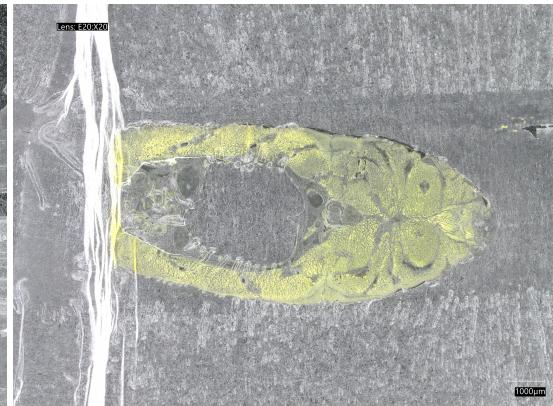


Figure 4.8 vertical line

Additionally, some images were noted to have a significant rotational angle at the time of sampling. These instances are categorized separately as **slope**(Figure 4.9). Finally, any images that do not fit into the aforementioned categories but still show irregularities(Excessive changes in brightness) are labeled as **other**(Figure 4.10).

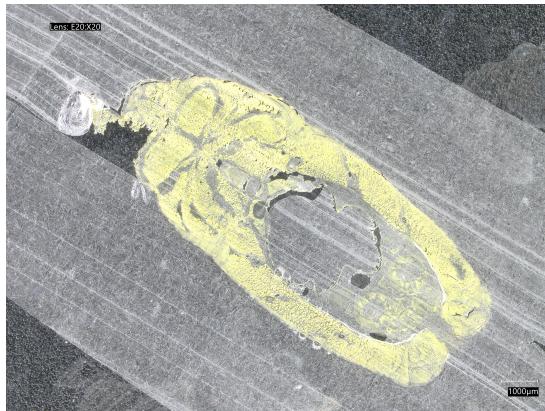


Figure 4.9 slope

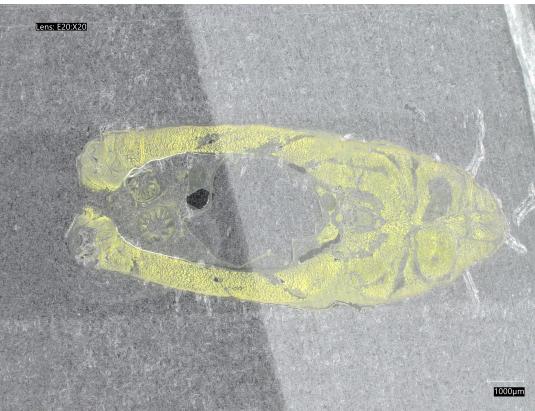


Figure 4.10 other

An example of a normal slice that meets observational requirements is shown in Figure 4.11.



Figure 4.11 normal

For each image, we need to label it as one of the above five categories. This will serve as our dataset for training the model.

4.3 *Model 1: Original Images with a Simple CNN Network*

For a new dataset, where the appropriate complexity of the model for the given image complexity is uncertain, a basic CNN architecture is initially employed to gauge the characteristics of the dataset and the complexity of the images.

Table 4.1 Configuration of the simple CNN model

Layer Type	Configuration 1a	Configuration 1b	Configuration 1c
Input Layer	-	-	-
Conv Layer 1	Conv3-32 (relu)	Conv3-16 (relu)	Conv3-32 (relu)
Pooling Layer 1	MaxPooling	MaxPooling	MaxPooling
Conv Layer 2	Conv3-32 (relu)	Conv3-32 (relu)	Conv3-32 (relu)
Pooling Layer 2	MaxPooling	MaxPooling	MaxPooling
Conv Layer 3	Conv3-32 (relu)	Conv3-64 (relu)	Conv3-32 (relu)
Pooling Layer 3	MaxPooling	MaxPooling	MaxPooling
Flattening Layer	Flatten()	Flatten()	Flatten()
FC(Full connect)	Dense(128, relu)	Dense(128, relu)	Dense(256, relu)
Output Layer	-	-	-

The configurations of the simple CNN models used are outlined in Table 4.1. These initial models, labeled as Configuration 1a, 1b, and 1c, vary in terms of the number of neurons in the convolutional layers and the size of the neurons in the fully connected layers. Configuration 1a and 1b differ by the number of neurons in the convolutional layers, whereas Configuration 1c differs from Configuration 1a in the size of the neurons in the fully connected layer.

The preprocessing steps involve splitting the dataset into training (80%) and testing (20%) sets. In the input layer, the image dimensions are halved (from 2880x2160 to 1440x1080), and the data is normalized.

During training, the Adam optimizer and cross-entropy loss function are used, with early stopping implemented to avoid overfitting.

The graphs below display the accuracy and loss of Models 1a, 1b, and 1c over the training epochs.

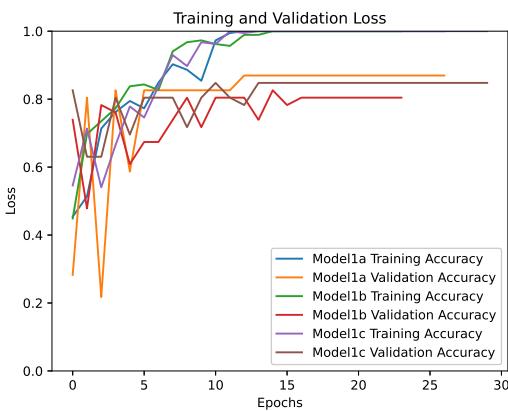


Figure 4.12 Accuracy of Model 1

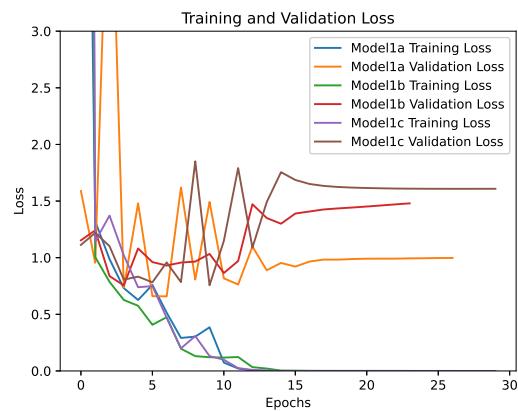


Figure 4.13 Loss of Model 1

From the charts, it is observed that Models 1a, 1b, and 1c exhibit a gradual increase in training accuracy,

stabilizing over time, while training losses decrease, approaching zero. This indicates that the models are learning from the training data relatively well. However, for the validation set, the accuracy of all three models stabilizes within the range of 80% to 85%, and the validation loss is comparatively high in some cases, especially with Model 1a, where it approaches 2.5 and shows significant fluctuations. This suggests a degree of overfitting, where the models perform better on the training data than on unseen data. Notably, Model 1c shows the best performance in terms of validation loss, indicating that its structure or parameter adjustments may be more effective at improving generalization.

The likely cause of overfitting here could be the insufficient complexity of the models relative to the complexity of the dataset, indicating that the models may not be effectively extracting features from the data. Although the models achieve high accuracy and low losses on the training set, their generalization capabilities on the validation set need enhancement.

To improve model accuracy, considering preprocessing of the images and assisting the model with feature extraction manually might be beneficial, helping the model better generalize to new data.

4.4 *Image Preprocessing Improvement*

In cases where model performance is suboptimal, it may be due to the complexity of the images which hampers the model's ability to extract significant features effectively. Therefore, image preprocessing techniques such as edge detection and threshold segmentation are considered to highlight desired features for recognition by the model and to reduce irrelevant features and noise, thereby improving the accuracy of subsequent deep learning models.

4.4.1 *Edge Detection*

As mentioned in section 3.1.1, the principle of edge detection involves identifying changes in pixel intensity (gradients) to determine edges within an image. Assuming the original image is displayed in Figure 4.6.

Before proceeding with edge detection, an initial preprocessing step—Gaussian blur—is applied. The rationale behind Gaussian blurring is that it helps reduce noise in the image, smoothens the gradient transitions, and decreases the likelihood of detecting false edges, thus enhancing the accuracy of edge detection[20]. We experiment with Gaussian kernels of sizes 21, 41, 61, and 81, which correspond to 1%, 2%, 3%, and 4% of the image width, respectively.

The images post-Gaussian blurring are displayed below. To better demonstrate the impact of the Gaussian kernel size on edge detection, the Sobel operator is used post-blurring to compute edges and increase brightness by 50 units for visibility.

From Figure 4.14 to Figure 4.20, it can be observed that as the Gaussian blur kernel size increases, the image details become progressively more blurred, and the edges also become more indistinct. From Figure 4.15 to Figure 4.21, the effectiveness of edge detection diminishes as the kernel size increases, with the edges becoming less prominent. Considering the clarity of image edges against background noise, a Gaussian kernel size of 61 is selected.

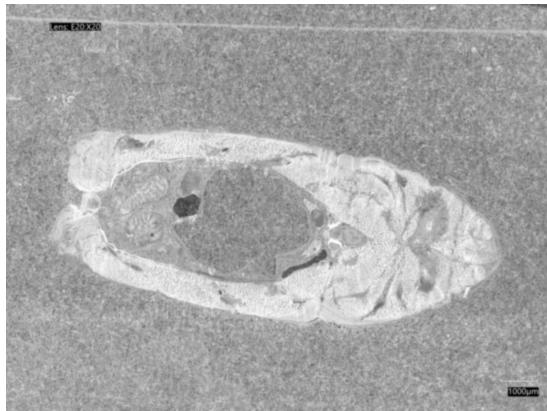


Figure 4.14 blurred k=21

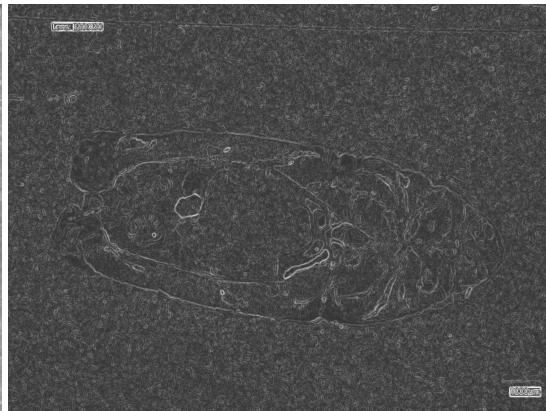


Figure 4.15 sobel k=21

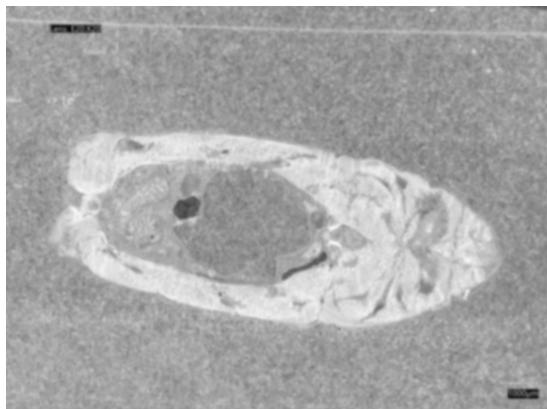


Figure 4.16 blurred k=41



Figure 4.17 sobel k=41

Following the application of Gaussian blur (k=61), the results using the Laplacian operator via Python's OpenCV library are depicted below:

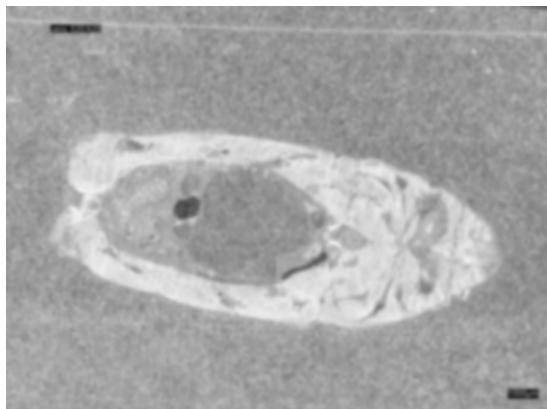


Figure 4.18 blurred k=61

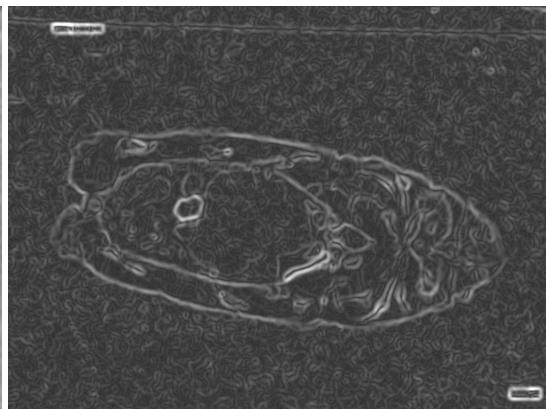


Figure 4.19 sobel k=61

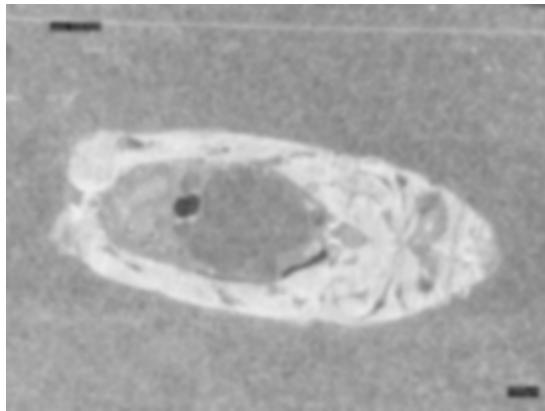


Figure 4.20 blurred k=81

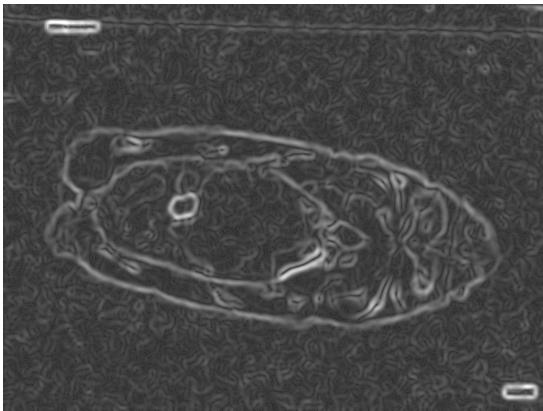


Figure 4.21 sobel k=81



Figure 4.22 Result of applying the Laplacian operator

As previously mentioned, the Canny algorithm is more sophisticated compared to the Sobel algorithm, incorporating steps such as thresholding and non-maximum suppression. The Canny method uses two thresholds, a low and a high. An image gradient greater than the high threshold is marked as an edge, while a gradient below the low threshold is not considered an edge. Gradients that are between the two thresholds are only considered edges if they are connected to high-threshold edges, effectively reducing noise and resulting in more accurate edge detection.

Typically, the ratio between the high and low thresholds is between 2:1 and 3:1. For this experiment, a ratio of 2.5:1 is selected, and the impact of different thresholds on edge detection is explored.

The chosen low thresholds are 2, 4, and 6, with corresponding high thresholds of 5, 10, and 15, respectively. The results of the Canny algorithm are shown below:



Figure 4.23 Canny thresholds 2 and 5

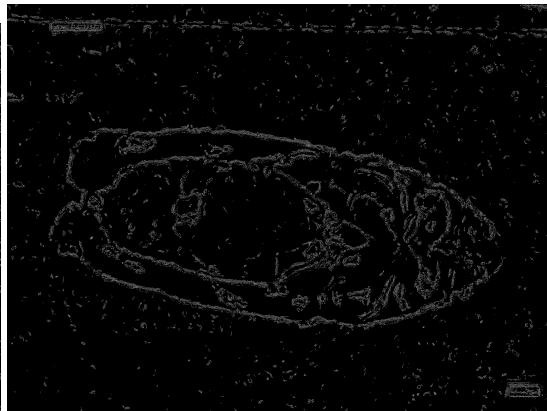


Figure 4.24 Canny thresholds 4 and 10



Figure 4.25 Canny thresholds 6 and 15

Among the three Canny results, Figure 4.24 exhibits the best performance, managing to retain most of the edge details while effectively eliminating most of the noise. Consequently, the thresholds of 4 and 10 are chosen for the Canny algorithm.

Summary

Comparing the results of the Sobel, Laplacian, and Canny algorithms, the Sobel algorithm shows average performance, with significant edge detection but limited noise reduction. The Laplacian algorithm performs the worst, with edges becoming nearly invisible, likely due to its high sensitivity to noise. The Canny algorithm delivers the best results, maintaining edge details while effectively removing most noise. Therefore, the Canny algorithm is selected as the method for image preprocessing, enhancing the model's ability to focus on relevant features for further analysis.

4.4.2 *Threshold Segmentation*

Considering the distinct colors of the biological tissue samples (yellow) and paraffin (white) in the specimens, threshold segmentation offers a straightforward method to distinguish between these two

components by isolating the white regions of the image, leaving the biological tissue intact. This procedure involves enhancing the image contrast and saturation to better highlight the yellow color of the biological tissues, as demonstrated in Figure 4.26. The processing steps are executed using Python's OpenCV library.

Initially, each pixel in the image is evaluated, and pixels within approximately a radius of 15 (about 1% of the image width) surrounding yellow pixels are preserved. Other colors are removed, as shown in Figure 4.27. However, this method has shown limitations due to the dispersal of tissue fragments during the sectioning process, which can appear scattered throughout the specimen and interfere with the detection of yellow pixels.

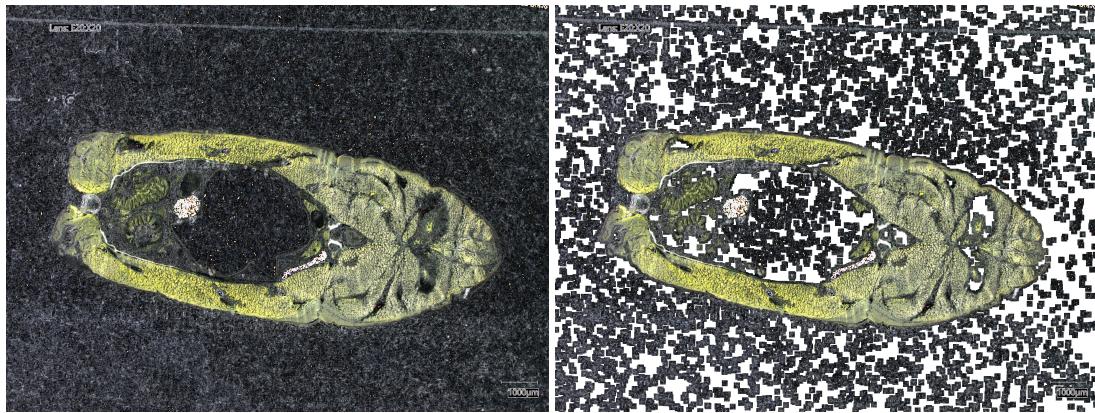


Figure 4.26 Enhanced image for better color differentiation

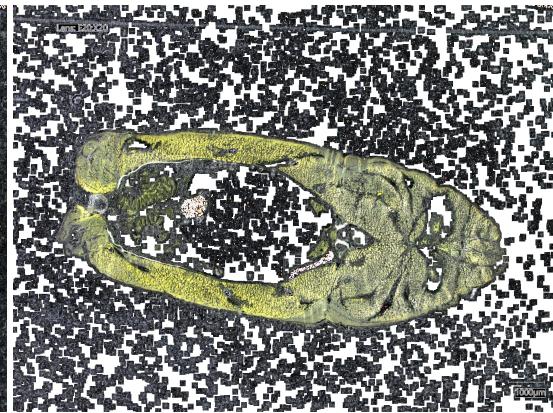


Figure 4.27 Segmentation focused on yellow pixels

To refine the segmentation, further processing is required to eliminate black blocks appearing in the image. This is achieved by applying a mask inversion to turn these black blocks into white, thereby enhancing the separation of biological tissue from the paraffin base. The results are displayed in Figure 4.28.



Figure 4.28 Final image after removing black blocks

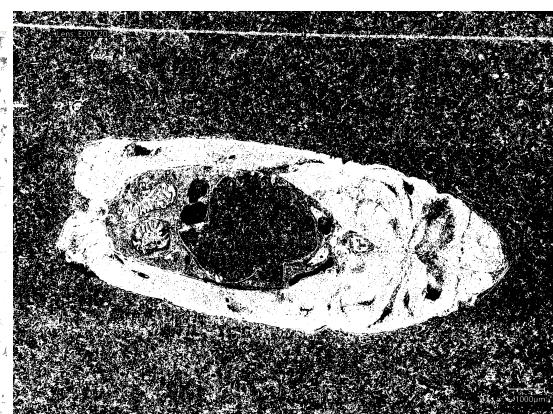


Figure 4.29 Result of segmentation using the fingerprint algorithm

This approach demonstrates the utility of combining color enhancement and thresholding techniques

to effectively segment biological tissue from paraffin in microscopic images. The challenge lies in accurately distinguishing tissue fragments from background noise and other non-tissue elements. This method can be particularly effective for automated image analysis in histopathology, where accurate tissue segmentation is crucial for research.

4.4.3 Another Threshold Segmentation Method: Fingerprint Algorithm

During the literature review, an article was found that described an improved segmentation method based on the Otsu algorithm, specifically adapted for fingerprint segmentation. Considering that both tissue sections and fingerprints are biological tissues with complex patterns and textures, it was hypothesized that this algorithm might also be effective for segmenting tissue sections. The results of applying this method are illustrated in Figure 4.29.

The fingerprint algorithm, which is an adaptation of the Otsu method, is particularly effective in differentiating areas of high and low density, which is ideal for applications like fingerprint recognition where contrast between ridges and valleys is critical. The use of this algorithm in the context of biological tissue segmentation may offer a robust way to delineate regions of different cellular densities or structures within a sample.

Summary

Based on the image preprocessing techniques discussed, both edge detection and threshold segmentation have shown promising results in highlighting the features of biological tissues and eliminating interference from paraffin. These preprocessing steps have significantly enhanced the visibility of essential features while minimizing noise and irrelevant information, which is critical for accurate analysis in histopathology.

To leverage these improvements, three datasets can be established:

- Images processed through **edge detection**.
- Images processed through **threshold segmentation**.
- Images processed using **the fingerprint algorithm**.

These datasets will serve as training sets for the upcoming model training phase. Utilizing diverse preprocessing approaches not only enhances model robustness by providing varied representations of the data but also helps in exploring which image preprocessing technique best assists the model in learning relevant features effectively.

4.5 Model 2: Preprocessed Images with a Simple CNN Network

In this section, we adapt Model 1c, the best-performing model from the previous experiments, to utilize preprocessed images. The architecture remains unchanged; however, the input now consists of images that have undergone various preprocessing techniques:

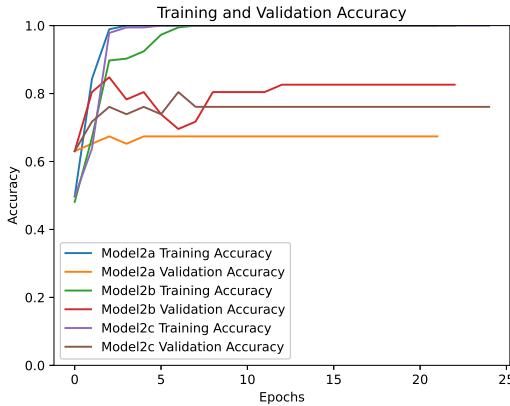


Figure 4.30 Accuracy of Model 2

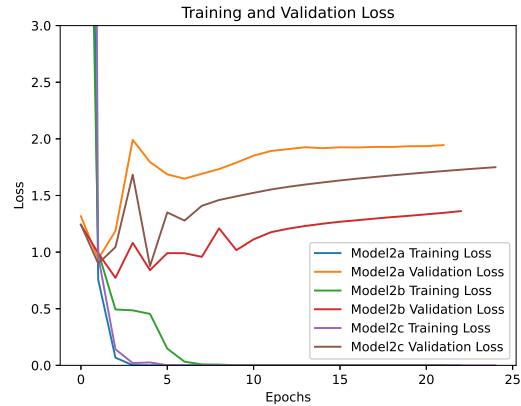


Figure 4.31 Loss of Model 2

- **Model 2a:** Utilizes images processed with Canny edge detection.
- **Model 2b:** Uses images processed through threshold segmentation.
- **Model 2c:** Inputs are images segmented using the fingerprint algorithm.

Each model follows the same architecture as Model 1c, which includes three convolutional layers each with 32 feature maps and 3x3 kernels, max pooling layers, and a fully connected layer with 256 neurons.

Results are displayed in the graphs below (Figure 4.30 and Figure 4.31), showcasing the training and validation accuracy and loss for Models 2a, 2b, and 2c.

Summary

The graphs illustrate that Models 2a and 2c begin to stabilize after approximately 8 training epochs, with training accuracies nearing 100%, while validation accuracies stabilize around 65% and 75%, respectively. Despite high accuracies, both models exhibit relatively high validation losses above 1, suggesting overfitting and limited generalization to unseen data.

Model 2b, however, converges after about 10 training epochs, displaying the highest validation accuracy at approximately 82% and a lower loss fluctuating between 1 and 1.2. This indicates that Model 2b performs better on the validation set, suggesting better adaptability and robustness. This might be due to Model 2b processing color images, which provide richer features from RGB channels, potentially enhancing feature extraction and generalization capabilities.

However, there is a risk that important details could be lost in the preprocessing steps, particularly with Model 2b's threshold segmentation. This can negatively affect the model's performance on specific types of images. An example of this loss of key information is demonstrated below:

In Figure 4.33, we see that the threshold segmentation algorithm used in Model 2b's training set significantly enhances horizontal creases, potentially confusing the model during training.

These findings highlight the challenges in image preprocessing. Aggressive preprocessing can sometimes eliminate crucial information, leading to diminished training outcomes. In future steps, trans-

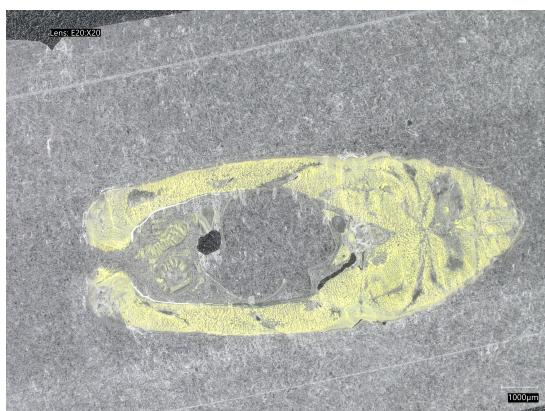


Figure 4.32 Original Image

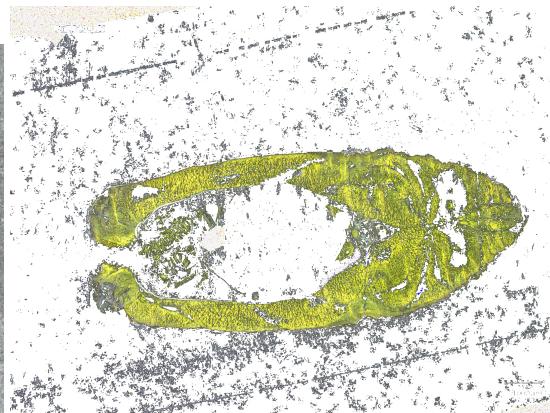


Figure 4.33 Image after Yellow Threshold Segmentation

fer learning might be employed, utilizing pre-trained large-scale deep learning models adapted to our dataset, to improve training effectiveness and address these challenges.

4.6 *Model 3: Original Images with Transfer Learning*

Transfer Learning with Pre-trained Models We are incorporating three well-known models pre-trained on the ImageNet dataset: VGG16, VGG19, and InceptionV3. These models come with pre-trained weights that are highly optimized and are expected to improve feature extraction capabilities significantly when adapted to our specific dataset.

- VGG16 (Model 3a) and VGG19 (Model 3b) are similar, with VGG19 having three additional convolutional layers that could potentially offer better feature extraction capabilities.
- InceptionV3 (Model 3c) incorporates Inception modules that allow it to capture a broader range of features at multiple scales, providing a more complex and possibly more effective feature extraction mechanism.

Adaptations for Transfer Learning

To prevent overfitting and optimize the transfer learning process:

- Early stopping is utilized.
- Learning rates are set low, at 1e-5 for VGG16 and VGG19, and slightly higher at 1e-4 for InceptionV3, given its more complex architecture.
- All models are adapted to accept an input size of 224x224, except for InceptionV3 which uses its default input size of 299x299. This uniform input size helps standardize the data preprocessing step.
- A global average pooling layer is added to each model following the base model layers, followed by a fully connected layer that matches the number of output classes.

Observations from Model Training

The training and validation performance of these models is depicted below:

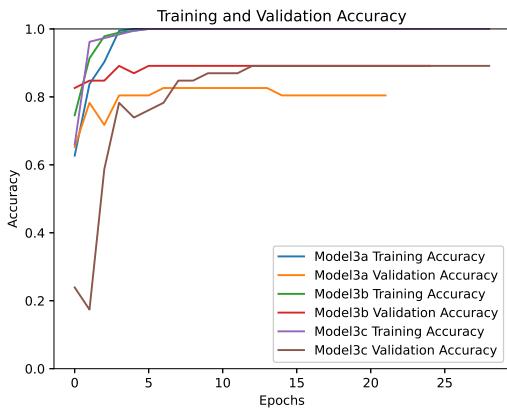


Figure 4.34 Accuracy of Model 3

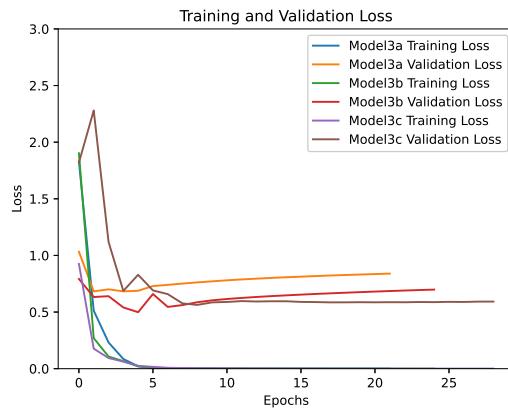


Figure 4.35 Loss of Model 3

Analysis

- Models 3b (VGG19) and 3c (InceptionV3) show significantly higher validation accuracies around 90%, compared to Model 3a (VGG16).
- The loss metrics indicate that Model 3c (InceptionV3) has the lowest validation loss among the three, suggesting it is the most effective in generalizing to unseen data. This underscores InceptionV3's superior capability in capturing complex features.
- The performance gap between Model 3a and Model 3b supports the notion that the additional convolutional layers in VGG19 enhance its ability to process image features more effectively than VGG16.

Summary

A comparative analysis of the VGG16, VGG19, and InceptionV3 models reveals that InceptionV3 yields the best training results, with both training and validation accuracies converging around 1 and 0.9 respectively, and losses converging around 0.6. This indicates that the InceptionV3 model not only trains effectively but also demonstrates superior generalization capabilities.

4.7 Model Selection Summary

When comparing across the model series—Model 1, Model 2, and Model 3—it is evident that Model 3 performs the best, particularly Model 3c. The underlying reason is likely due to Model 3's foundation on deep convolutional networks that are pre-trained on large-scale image datasets, enabling more effective feature extraction and development of a robust feature space.

Notable Attributes of Model 3c (InceptionV3):

- Architectural Design:** InceptionV3 features a modular design incorporating multiple "inception modules," which include multi-scale convolutional layers that operate in parallel within the same

layer. This modular approach allows the network to capture a broad spectrum of features at various scales and depths.

- **Feature Extraction:** Inception modules can adaptively capture appropriate feature representations by processing different scales of features within the same layer. This adaptability makes it exceptionally capable of handling complex image data like biomedical images.
- **Deep Network Handling:** InceptionV3 integrates batch normalization and residual connections, which are critical in training deep networks. These techniques effectively mitigate issues related to vanishing gradients, thus facilitating the training of deeper models without performance degradation.

Given these advantages, Model 3c (InceptionV3) is selected as our final model for further applications and testing. This model stands out not only for its advanced architectural innovations but also for its proven effectiveness in generalizing well to new, unseen data, making it highly suitable for complex tasks such as medical image analysis where accuracy and reliability are paramount.

5 Presentation of Experimental or Analytical Results/Descriptions of Final Constructed Product

In this section, we discuss the testing outcomes of our models and explore areas for further improvement.

5.1 *Validating Model Accuracy on a Test Set*

After training, the models were evaluated on a specially prepared test set to measure their accuracy. The accuracy is defined as the proportion of samples for which the model's predictions match the actual labels.

The table and figure below present the accuracy of the model across different categories:

Table 5.1 Model Accuracy on Test Set

Category	Accuracy(%)
Normal	98.4
Horizontal Line	95.6
Vertical Line	80.0
Slope	96.1
Other	95.2

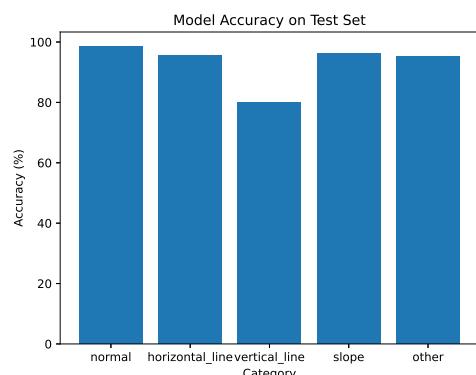


Figure 5.1 Model Accuracy on Test Set

Analysis of Model Performance The model performs exceptionally well for the 'Normal' category with an accuracy of 98.4%, indicating its robust capability in identifying tissue sections without significant defects. Similarly, high accuracy scores are noted for 'Horizontal Line' and 'Other' categories, reflecting the model's effectiveness in recognizing these specific types of imperfections.

However, the accuracy for the "Vertical Line" category is significantly lower, at 80.0%. This indicates that the model's ability to recognize this type of defect needs to be strengthened. This could be due to insufficient training data, which limits the model's learning. It could also be because the features of vertical lines are relatively less obvious, making it difficult for the model to accurately extract features.

5.2 *Further Improvement of the Model (Changing Input Resolution)*

Here we discuss the potential for further improvement of the model.

Rescaling high-resolution images to the default size of 299x299, as required by the InceptionV3 model, can indeed result in the loss of information and detail. This is particularly crucial for images originally at much higher resolutions, such as those captured by the VHX7000 device at 2880x2160. Directly scaling down these images may hinder the model's ability to capture all subtle differences, which is especially detrimental in fields like medical imaging where detail richness is paramount.

One potential solution is to modify the model's input layer to accept larger image sizes. This approach allows the model to process higher resolution images, thus retaining more original information and detail, which could lead to improved performance and higher accuracy. The InceptionV3 architecture, with its multiple convolutions of varying kernel sizes, is particularly well-suited to handle larger images as it can capture features at different scales effectively.

Due to the limitations imposed by the lab's hardware (16GB of VRAM), the images are rescaled to 0.4 times their original size, resulting in dimensions of 1152x864 for this experiment.

Training the New Model (Model 4)

Model 4 is trained with these adjusted image sizes, and its training effectiveness is as follows:

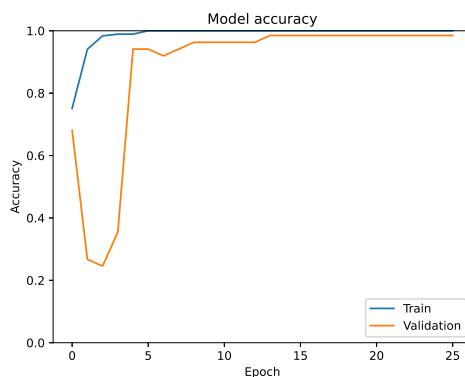


Figure 5.2 Accuracy of Model 4

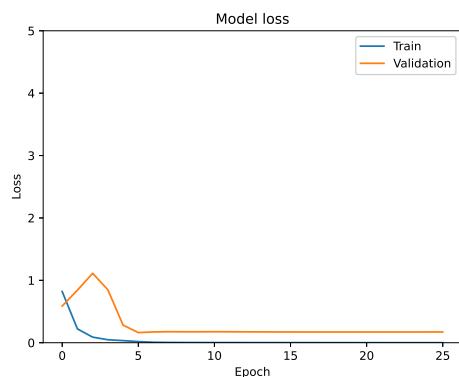


Figure 5.3 Loss of Model 4

Table 5.2 Model 4 accuracy on the test set

	normal	horizontal_line	vertical_line	slope	other
accuracy(%)	98.4	96.7	85.6	96.5	96.5

Observations of training accuracy and loss over time indicate a significant improvement in model performance. Both training and validation accuracies approach 1, with validation losses dropping to around 15%, suggesting strong generalization capabilities. This indicates that the model not only excels on training data but can also generalize effectively to new, unseen data.

Re-Evaluating Accuracy on the Test Set

The updated model is then re-evaluated on the test set, with results as Table 5.2:

Comparing the accuracy before and after changing the resolution, there is a noticeable improvement, though it is not substantial. This modest increase could be attributed to the already high accuracies nearing 1, where further improvements have diminishing returns.

The results affirm the potential benefits of processing higher-resolution images, particularly in settings demanding high fidelity and detail, such as biological tissue analysis and research.

5 . 3 *Investigating the Optimal Cutting Angle for the Machine*

In order to determine the optimal cutting angle for the microtome, images of tissue sections cut at various angles ranging from 8 to 12 degrees, at 0.5-degree increments, were prepared. Each angle category consisted of 100 images, resulting in a total of 9 distinct groups of data. Model 4 was then utilized to assess the quality rate of each group, aiming to identify the angle at which the highest yield of quality sections was achieved.

The table and graph below present the accuracy, defined as the percentage of high-quality cuts, for each cutting angle:

From the data presented in Figure 5.3, it is evident that the optimal cutting angle for achieving the highest yield of quality tissue sections is 10 degrees, which demonstrates an impressive 96.6% accuracy.

Additionally, as illustrated in Figure 5.4, to maintain a section quality rate of at least 80%, the cutting angle should be set between 9 and 10.5 degrees. This range not only ensures a high rate of quality cuts but also offers some flexibility in machine settings to accommodate possible variations in tissue type or condition.

5 . 4 *Model Generalizability*

The experiments thus far have utilized ovarian tissue sections from fish. In practical applications, we may encounter diverse tissue samples, including other organs or specimens from different animals. Therefore, it's crucial to assess the generalizability of our model across various tissue types.

Table 5.3 Normal accuracy on different angles

Angle	Accuracy(%)
8	80
8.5	81.5
9	83.5
9.5	93.3
10	96.6
10.5	88.8
11	84.2
11.5	66.6
12	62.2

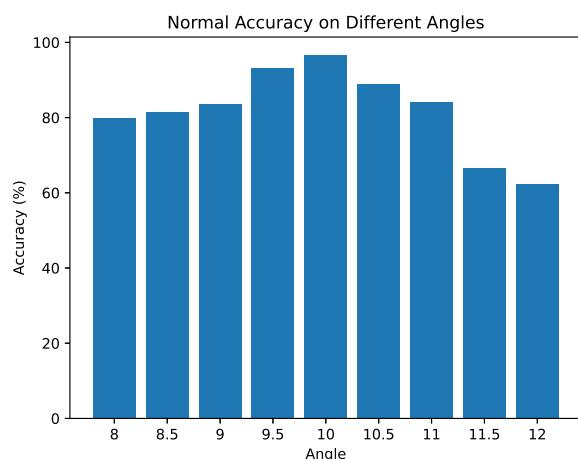


Figure 5.4 Model Accuracy on Different Angle

A new dataset comprising fish lung tissue sections has been prepared, categorized into four classes: good, normal, bad, and other. These categories are demonstrated in the figures below: (From Figure 5.5 to Figure 5.8)

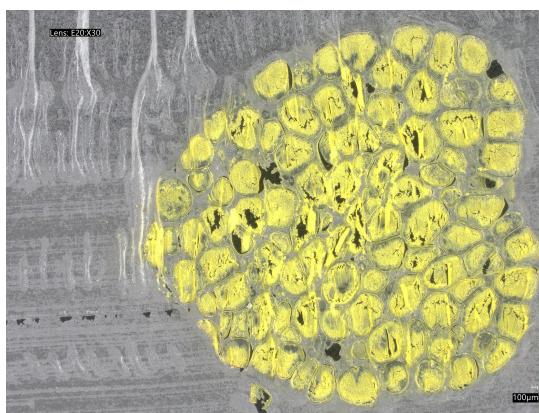


Figure 5.5 Good fish lung

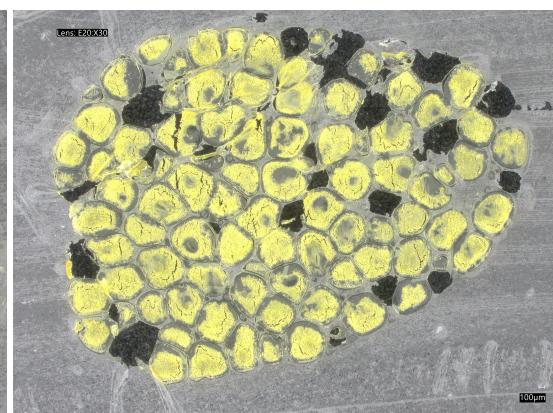


Figure 5.6 Normal fish lung

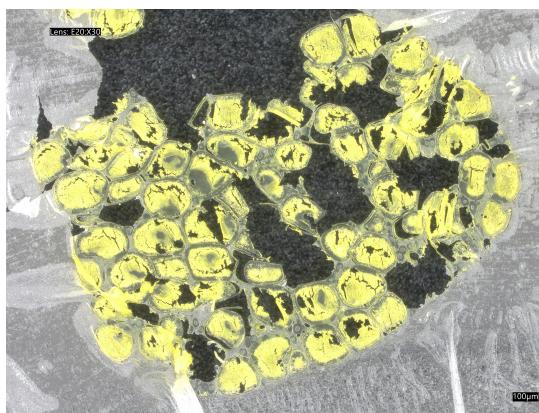


Figure 5.7 Bad fish lung

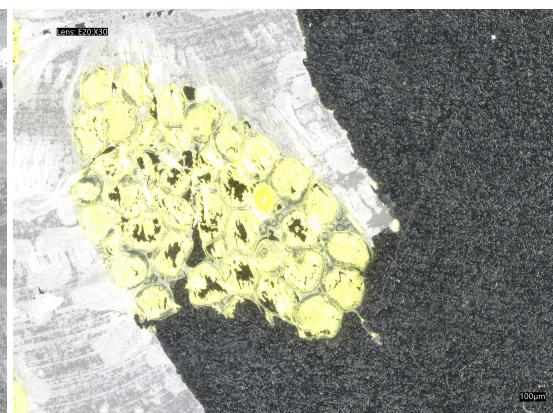


Figure 5.8 Other fish lung

The original model architecture (Model 4) is maintained but retrained with the fish lung images at a resolution of 1152x864. The training accuracy and loss are presented in the Figure 5.9 and Figure 5.10.

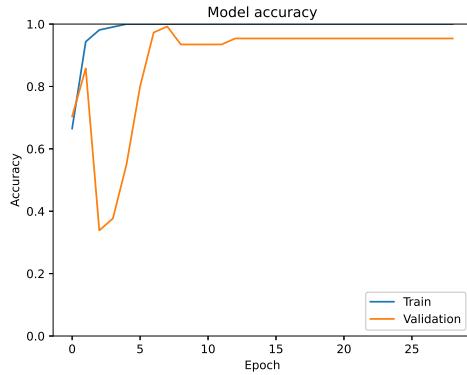


Figure 5.9 Accuracy of Model 5

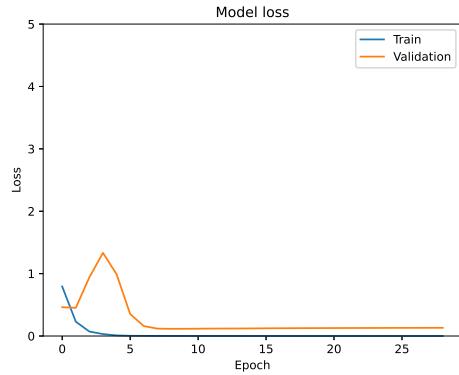


Figure 5.10 Loss of Model 5

The training and validation accuracy rapidly increase and maintain high levels, indicating robust model performance on both datasets. The loss plot shows a rapid decline in training loss towards zero, with validation loss stabilizing after an initial spike—suggesting good fit and generalization.

The model is further tested on a test set, and the results are shown in Figure 5.11.

Table 5.4 Model accuracy on the test set

label	accuracy(%)
bad	94.1
good	98.2
normal	94.7
other	95.0

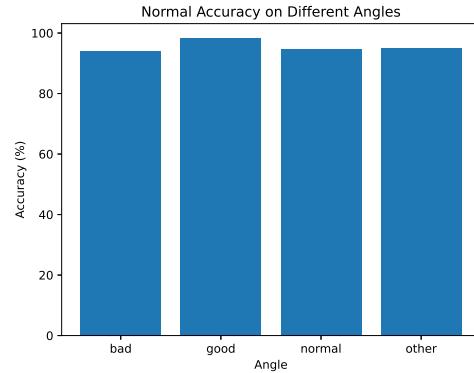


Figure 5.11 Model Accuracy on Test Set

The model demonstrates over 90% accuracy across all labels, indicating its strong performance and substantial generalizability. This suggests that the model can effectively classify different types of tissue sections, potentially making it a versatile tool for various biomedical imaging applications. The robustness of the model across different tissue types underscores its potential in tissue quality assessment and classification tasks.

6 Discussion and conclusions

6.1 *Discussion of results*

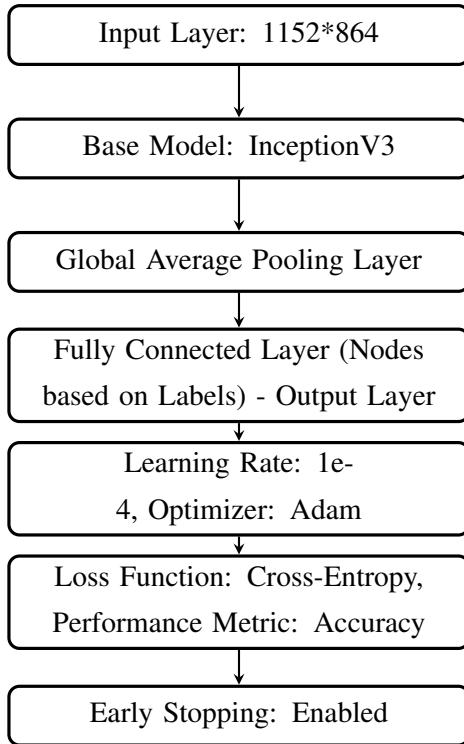
As detailed above, this research aimed to establish a reliable model for classifying tissue section images. Initially, simple CNN models were implemented, but upon observing limited success, the study shifted towards image preprocessing and finally settled on transfer learning with the InceptionV3 model, which produced the best results.

It is worth noting that with the trial of different models, as the model parameters are adjusted or the model architecture becomes more complex (such as InceptionV3), the model's performance significantly improves, i.e., the accuracy of the validation set becomes higher and the loss becomes lower.

Furthermore, comparing model series 1 and 2, we found that using preprocessed images to assist the machine in extracting features is not very effective in the task of image classification. Image processing may lead to the loss of important details and information, thereby affecting the machine's feature extraction, and thus affecting the accuracy and performance of the model.

In Section 5, we tested the model in application. First, we selected an additional test set to test the model's accuracy and found that the model's accuracy on all test sets was greater than 85%. Then we used the model to evaluate different cutting angles and found that if the cutting quality is to be guaranteed at 80%, the cutting angle should be between 9 degrees and 10.5 degrees. Finally, we used another dataset of fish alveolar slice images for secondary verification and found that the model's prediction accuracy for the test set labels was all above 90%, reflecting that the model can be well applied to other datasets.

The Final selected model configuration, illustrated below, highlights the structured approach taken to integrate the InceptionV3 architecture effectively within the training framework.



6.2 Future work

6.2.1 Enhancing Classification Methods

Broadening Classification Categories

The current research has demonstrated promising results with the classification methods employed; however, the scope of classification remains limited to five categories. Enhancing the diversity of these categories could facilitate a deeper understanding of the relationships between cutting angles and sample quality, providing more precise analytical capabilities. More comprehensive classification could enable the prediction of optimal cutting angles based on a wider range of tissue responses, potentially improving outcomes for various tissue types and conditions.

Transitioning to Linear Analytical Methods

With a sufficiently granular set of classification points, it may become feasible to transition from a categorical framework to linear analytical methods. When enough categories are present, these can be approximated as discrete points within a linear relationship, allowing for linear regression to model the relationship between cutting angles and sample quality.

Linear Discriminant Analysis (LDA) is a statistical method used when categorical outcomes can be linearized, as discussed in the context of cutting angles. Employing LDA, especially in determining optimal cutting angles, could provide a more refined approach to correlating tissue quality with cutting parameters. This method could simplify the prediction of cutting parameters and aid in more precisely controlling the tissue sectioning process.

Challenges and Considerations

Transitioning from a classification problem to a linear discriminant analysis framework is challenging. One significant issue is that while binary classification models output probabilities, linear regression inherently seeks to identify relationships between variables (e.g., cutting angles as independent variables and cutting quality as the dependent variable). Moreover, the relationship between cutting angles and tissue quality is likely not a simple linear one, complicating the data the model needs to handle.

Further, training and validating linear models require significantly more data, potentially several orders of magnitude greater than currently used, making data collection a lengthy and challenging process. In addition, training and validating linear regression models demand substantial computational resources. Currently, using the TensorFlow framework with an InceptionV3 model at an input resolution of 1152x864 is pushing the limits of available GPU memory, suggesting that advancements in this area require additional hardware and computational power.

Long-term Goals and Resource Needs

These improvements represent long-term goals needing considerable resources and time. Research into linear discriminant analysis, for instance, involves deeper theoretical exploration and practical testing. For example, Jie Wen's work on "Robust Sparse Linear Discriminant Analysis" introduces sparsity into the LDA framework, making the model more robust and potentially more suited to complex real-world applications[21].

6.2.2 *Performance Enhancement and Optimization*

As this research progresses towards large-scale application, performance optimization emerges as a crucial challenge. This involves not only enhancing algorithm efficiency but also improving the scalability, stability, and deployment capabilities of the model framework, as well as optimizing the underlying programming languages and code.

To optimize the use of computational resources, adopting more efficient computing frameworks and parallel processing algorithms is essential. Utilizing distributed computing resources can significantly reduce model training times and enhance efficiency when processing large datasets. Moreover, considering constraints on energy consumption and computational costs, it's vital to optimize the model's computational architecture and parameter settings to maximize output within limited resources.

The article "Analysis of the Application Efficiency of TensorFlow and PyTorch in Convolutional Neural Network" highlights the differences between TensorFlow and PyTorch in processing convolutional neural networks[22]. TensorFlow exhibits a lower error rate and smaller convergence steps, whereas PyTorch offers faster training speeds.

Pascal Fua's "Comparing Python, Go, and C++ on the N-Queens Problem" presents methods to optimize deep learning performance by comparing the efficiency of Python, Go, and C++ in solving the N-Queens problem.[23] It was found that runtime languages have clear advantages in handling loops and data flows, suggesting that compiling tools like Numba, Cython, and Pybind11 can enhance performance in deep learning applications.

6.2.3 *Optimization of the Sectioning Process*

Our research also uncovered that real-time assessment of section quality during the cutting process, followed by adjustments based on those assessments, could significantly improve the quality of tissue sections.

The proposed feedback adjustment process involves installing a camera above the microtome to capture data from the samples being cut. This data is then analyzed in real-time by a pre-trained model, which assesses the quality of the sections. Based on this assessment, the cutting speed and angle parameters of the microtome can be adjusted to improve the quality of subsequent sections, thus ensuring controllable and consistent sample quality.

Implementing this system presents several challenges:

- **Real-Time Image Processing:** A clear camera and an efficient real-time image processing system are needed to capture and process image data swiftly.
- **Powerful Computing Resources:** A pre-trained model and a powerful computer are required to quickly assess images and adjust the microtome's parameters based on the assessment.
- **Effective Control Interface:** An efficient control interface is necessary to ensure that the adjusted parameters are promptly communicated to the microtome.
- **Time Efficiency:** The entire system must operate within the brief intervals between cuts.

A pertinent example can be found in the study "Convolutional neural networks applied to microtomy: Identifying the trimming-end cutting routine on paraffin-embedded tissue blocks"[24]. This research automated the sectioning process by monitoring it with a camera, analyzing the images with a CNN, and adjusting the microtome parameters based on the analysis. This integration of the microtome, camera, and deep learning model provides a feasible solution for real-time assessment and adjustment of cutting parameters during the sectioning process.

6.3 *Conclusions*

This research has significantly advanced our understanding of optimizing biopsy parameters through the application of deep learning techniques to biomedical tissue sectioning devices. By employing sophisticated convolutional neural networks, particularly through the adaptation of the InceptionV3 model via transfer learning, the study has demonstrated a robust framework capable of assessing the quality of tissue sections with high precision. This approach not only improves the accuracy of tissue sample analysis but also introduces a paradigm shift in the operational methodology of tissue sectioning.

By evaluating the quality of sections at different angles, we have discovered the relationship between the cutting angle and the quality of the sections. This provides us with a feasible method to improve the quality of sections in the future. In addition, the application of this model in different types of tissues (including fish ovaries and lung tissues) has confirmed its wide adaptability and potential for promotion, indicating that it is suitable for various tissue sections and research tasks.

However, the study also highlighted the limitations of traditional image preprocessing techniques. Initial attempts to enhance model performance through image preprocessing did not yield significant improvements and, in some cases, potentially obscured critical details necessary for accurate classification. This finding suggests that maintaining the integrity of original image data might be more beneficial than applying aggressive preprocessing techniques.

The research also explored the possibility of real-time assessment and automatic adjustment of sectioning parameters, proposing a feedback system that integrates image capture, real-time analysis, and mechanical adjustment to enhance the quality of sections dynamically. This system points towards the future of automated histology instruments, which could significantly reduce human error and variability in tissue sectioning.

In conclusion, this project not only demonstrates the important role of deep learning in biomedical research and applications, but also provides feasibility for the further improvement of tissue sectioning technology. The combination of these technologies can bring improvements to biological sectioning equipment, provide a possible solution to improve the yield rate of tissue samples and the efficiency of sectioning. This research provides new ideas and methods for future tissue sectioning technology, and is expected to have a profound impact in the field of biomedicine.

7 Project management, consideration of sustainability and health and safety

7.1 *Subsection 5.1*

7.2 *Subsection 5.2*

References

- [1] Murphy, K.P., 2012. Machine learning: a probabilistic perspective. Cambridge: The MIT Press, pp.32.
- [2] Zimmermann, M., Lampe, J., Lange, S., Smirnow, I., Königsrainer, A., Hann-von-Weyhern, C., Fend, F., Gregor, M., Bitzer, M. & Lauer, U.M. (2009). Improved reproducibility in preparing precision-cut liver tissue slices. *Cytotechnology*, 61(3), 145-152. <https://doi.org/10.1007/s10616-009-9246-4>
- [3] Klimuszko, E., Orywal, K., Sierpinska, T. et al. (2018) 'Evaluation of calcium and magnesium contents in tooth enamel without any pathological changes: in vitro preliminary study', *Odontology*, 106(4), pp. 369-376. <https://doi.org/10.1007/s10266-018-0353-6>
- [4] Hrzenjak, A., Moinfar, F., Tavassoli, F.A., Strohmeier, B., Kremser, M.-L., Zatloukal, K. and Denk, H. (2005) 'JAZF1/JJAZ1 gene fusion in endometrial stromal sarcomas: molecular anal-

- ysis by reverse transcriptase-polymerase chain reaction optimized for paraffin-embedded tissue', *The Journal of Molecular Diagnostics*, 7(3), pp. 388-395. [https://doi.org/10.1016/S1525-1578\(10\)60568-5](https://doi.org/10.1016/S1525-1578(10)60568-5)
- [5] Song, L., Mino, M., Yamak, J., Nguyen, V., Lopez, D., Pham, V., Fazelpour, A., Le, V., Fu, D., Tippin, M., Uchio, E. and Zi, X. (2022) 'Flavokawain A reduces tumor-initiating properties and stemness of prostate cancer', *Frontiers in Oncology*, 12. <https://doi.org/10.3389/fonc.2022.943846>
- [6] Azuma, T., Murata, Y., Hokii, Y., Akiyama, S. and Shinozaki, Y. (2023) 'Evaluation of micro-cracks formed by grinding in machinable lithium silicate', *Dental Materials*, 39, p. e10. Abstracts of the Academy of Dental Materials, Annual Meeting, Athens, Greece, 2022. <https://doi.org/10.1016/j.dental.2023.08.022>
- [7] Veer, F.A. (2022) 'Looking at the foundations of structural glass with a digital microscope', in *Structures and Architecture. A Viable Urban Perspective*, 1st ed. CRC Press, pp. 7. eBook ISBN: 9781003023555.
- [8] Guachi-Guachi, L., Ruspi, J., Scarlino, P., Poliziani, A., Ciancia, S., Lunni, D., Baldi, G., Cavazzana, A., Zucca, A., Bellini, M., Pedrazzini, G.A., Ciuti, G., Controzzi, M., Vannozzi, L. and Ricotti, L. (2023) 'Convolutional neural networks applied to microtomy: Identifying the trimming-end cutting routine on paraffin-embedded tissue blocks', *Engineering Applications of Artificial Intelligence*, 126(B), p. 106963. <https://doi.org/10.1016/j.engappai.2023.106963>
- [9] Andrearczyk, V. and Whelan, P.F. (2017) 'Deep Learning in Texture Analysis and Its Application to Tissue Image Classification', in Depeursinge, A., Al-Kadi, O.S. and Mitchell, J.R. (eds.) *Biomedical Texture Analysis*. Academic Press, pp. 95-129. ISBN 9780128121337. <https://doi.org/10.1016/B978-0-12-812133-7.00004-1>
- [10] Xu, Y., Jia, Z., Wang, L.B., Ai, Y., Zhang, F., Lai, M., Chang, E.I.C., 2017. Large scale tissue histopathology image classification, segmentation, and visualization via deep convolutional activation features. *BMC Bioinformatics*, 18(1), p.281. Available at: <https://doi.org/10.1186/s12859-017-1685-x>.
- [11] Culjak, I., Abram, D., Pribanic, T., Dzapo, H., Cifrek, M., 2012. A brief introduction to OpenCV. In: 2012 Proceedings of the 35th International Convention MIPRO, Opatija, Croatia, pp. 1725-1730. Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6240859&isnumber=6240598>.
- [12] Bradski, G. and Kaehler, A., 2008. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc.
- [13] Canny, J., 1986. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), pp.679-698. doi: 10.1109/TPAMI.1986.4767851. Avail-

able at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4767851&isnumber=4767846>.

- [14] Yue, Y. and Zhu, J., 2017. Algorithm of fingerprint extraction and implementation based on OpenCV. In: 2017 2nd International Conference on Image, Vision and Computing (ICIVC), Chengdu, China, pp. 163-167. doi: 10.1109/ICIVC.2017.7984539. Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7984539&isnumber=7984442>.
- [15] Zhou, H. and Sun, Q., 2020. Research on Principle and Application of Convolutional Neural Networks. IOP Conference Series: Earth and Environmental Science, 440(4), p.042055. doi: 10.1088/1755-1315/440/4/042055. Available at: <https://dx.doi.org/10.1088/1755-1315/440/4/042055>.
- [16] Simonyan, K. and Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556. Available at: <https://arxiv.org/abs/1409.1556>.
- [17] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z., 2015. Rethinking the Inception Architecture for Computer Vision. arXiv preprint arXiv:1512.00567. Available at: <https://arxiv.org/abs/1512.00567>.
- [18] Chollet, F., 2016. Xception: Deep Learning with Depthwise Separable Convolutions. arXiv preprint arXiv:1610.02357. Available at: <https://arxiv.org/abs/1610.02357>.
- [19] Klimuszko, E., Orywal, K., Sierpinska, T. et al., 2018. Evaluation of calcium and magnesium contents in tooth enamel without any pathological changes: in vitro preliminary study. Odontology, 106, pp.369-376. doi: 10.1007/s10266-018-0353-6. Available at: <https://doi.org/10.1007/s10266-018-0353-6>.
- [20] Gedraite, E.S. and Hadad, M., 2011. Investigation on the effect of a Gaussian Blur in image filtering and segmentation. In: Proceedings ELMAR-2011, Zadar, Croatia, pp. 393-396. Available at: <https://ieeexplore.ieee.org/abstract/document/6044249>.
- [21] Wen, J. et al., 2019. Robust Sparse Linear Discriminant Analysis. IEEE Transactions on Circuits and Systems for Video Technology, 29(2), pp.390-403. doi: 10.1109/TCSVT.2018.2799214. Available at: <https://doi.org/10.1109/TCSVT.2018.2799214>.
- [22] Novac, O-C., Chirodea, M.C., Novac, C.M., Bizon, N., Oproescu, M., Stan, O.P., Gordan, C.E., 2022. Analysis of the Application Efficiency of TensorFlow and PyTorch in Convolutional Neural Network. Sensors, 22(22):8872. Available at: <https://doi.org/10.3390/s22228872>.
- [23] Fua, P. & Lis, K., 2020. Comparing Python, Go, and C++ on the N-Queens Problem. Available at: <https://arxiv.org/abs/2001.02491>.
- [24] Guachi-Guachi, L., Ruspi, J., Scarlino, P., Poliziani, A., Ciancia, S., Lunni, D., Baldi, G., Cavazzana, A., Zucca, A., Bellini, M., Pedrazzini, G.A., Ciuti, G., Controzzi, M., Vannozzi, L. &

Ricotti, L., 2023. Convolutional neural networks applied to microtomy: Identifying the trimming-end cutting routine on paraffin-embedded tissue blocks. *Engineering Applications of Artificial Intelligence*, 126, p.106963. Available at: <https://www.sciencedirect.com/science/article/pii/S0952197623011478>.