



Final Report

**Research on Deep Learning Models: Determining
the Optimal Cutting Parameters for a Microtome**
710082147

3rd Year Individual Project

I certify that all material in this thesis that is not my own work has been identified and that no material has been included for which a degree has previously been conferred on me.

Signed.....Jiaqi Yao.....|

Final Report

[ECM3175/ECM3149]

Title: Research on Deep Learning Models: Determining
the Optimal Cutting Parameters for a Microtome

Date of submission: May 3rd 2024

Student Name: Jiaqi Yao
Programme: Mechanical Engineering
Student number: 710082147
Candidate number: 116251

Supervisor: Dong Wang

Abstract

This research demonstrates the innovative application of deep learning technologies in optimizing the cutting parameters of microtomes used in biomedical research. Our objective is to enhance the precision and efficiency of tissue sample preparation by utilizing Convolutional Neural Networks (CNN) to analyze tissue sections and determine optimal cutting conditions. The study integrates advanced image analysis and machine learning to evaluate the quality of tissue samples under various cutting angles, thereby improving the production of high-quality specimens for microscopic examination.

Our methodology includes collecting a comprehensive dataset of tissue samples, classifying sample quality using artificial image classification, and developing a CNN model to predict the best cutting parameters. The performance of the model was validated through empirical testing, showcasing significant improvements in tissue sample quality and laboratory workflows. This work promises to advance biological and medical sciences by simplifying the tissue sample preparation process and enhancing the quality of tissue sections in scientific research.

The results indicate that deep learning models, especially those pretrained on large datasets, are highly effective in identifying and classifying the quality of tissue sections, thus providing a robust framework for the automatic adjustment of microtome settings. This study not only deepens our understanding of the factors affecting tissue section quality but also proposes a concept for future research on automating biological sample preparation using machine learning.

Keywords: Deep Learning Image Analysis Convolutional Neural Networks
Tissue Microtome

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 | 2 |
| 2.1 | Microtome and Microscope | 3 |
| 2.2 | Medical Image Classification - Based on Convolutional Neural Networks | 3 |
| 2.3 | Medical Image Classification - Based on Transfer Learning | 4 |
| 2.4 | Deep Learning in Tissue Sectioning | 5 |
| 3 | Methodology and theory | 5 |
| 3.1 | Experimental Workflow | 5 |
| 3.2 | Image Processing Methods | 6 |
| 3.2.1 | Edge Detection for Tissue Slices | 7 |
| 3.2.2 | Theresold Segmentation for Tissue Slices | 8 |
| 3.3 | Simple Convolutional Neural Network Framework | 8 |
| 3.4 | Model Selection of Transfer Learning Methods | 9 |
| 4 | Experimental work/ analytical investigation/ design | 10 |
| 4.1 | Data Collection | 10 |
| 4.2 | Data Labeling | 11 |
| 4.3 | Model 1: Original Images with a Simple CNN Network | 12 |
| 4.4 | Image Preprocessing Improvement | 13 |
| 4.4.1 | Edge Detection | 13 |
| 4.4.2 | Threshold Segmentation | 16 |
| 4.4.3 | Performing Threshold Segmentation Using Fingerprint Algorithm | 17 |

| | |
|--|----|
| 4.4.4 Summary | 17 |
| 4.5 Model 2: Preprocessed Images with a Simple CNN Network | 18 |
| 4.5.1 Summary | 18 |
| 4.6 Model 3: Original Images with Transfer Learning | 19 |
| 4.6.1 Summary | 20 |
| 4.7 Model Selection Summary | 20 |
| 5 Presentation of Experimental or Analytical Results/Descriptions of Final Constructed Product | 21 |
| 5.1 Testing Model Accuracy on a Test Set | 21 |
| 5.2 Improvement of the Model (Changing Input Resolution) | 21 |
| 5.3 Determining the Optimal Cutting Angle for the Machine | 23 |
| 5.4 Model Generalizability | 23 |
| 6 Discussion and conclusions | 25 |
| 6.1 Discussion of results | 25 |
| 6.2 Future work | 26 |
| 6.2.1 Enhancing Classification Methods | 26 |
| 6.2.2 Exploring the Impact of Other Parameters on Cutting Quality | 26 |
| 6.2.3 Performance Enhancement and Optimization | 27 |
| 6.2.4 Optimization of the Sectioning Process | 27 |
| 6.3 Conclusion | 28 |
| 7 Project management, consideration of sustainability and health and safety | 29 |
| 7.1 Project management | 29 |
| 7.2 Health and Safety | 29 |
| 7.3 Sustainability | 30 |
| References | 30 |

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 report is organized into multiple sections, each focusing on specific aspects of the research on optimizing biopsy parameters using deep learning:

Introduction and Background - This initial section outlines the project's goals and framework, provides the motivation for the research, and describes the technical protocols and standards employed.

Literature Review - An extensive review of the relevant literature on biological tissue sections, image classification, and deep learning applications in biological sample preparation. This section contextualizes the current study within existing research.

Methods and Theory - A comprehensive description of the experimental methods, theoretical frameworks, and plans for data collection and processing.

Experimental Work/Analysis Investigation/Design - Details the experimental design, implementation, and analytical survey, explaining the strategies and methods used to meet the project objectives.

Presentation of Experimental or Analysis Results/Final Constructed Product Description - This section documents the experimental data, analysis results, or descriptions of the final design products, elaborating on the outcomes.

Discussion and Conclusion - Discusses the results in terms of their scientific significance and practical implications, draws conclusions, and suggests future research directions.

Project Management, Sustainability, and Health Safety Considerations - Covers project management strategies, addresses sustainability and health safety issues to ensure the research is conducted efficiently and safely.

References - Compiles all literature referenced throughout the research, supporting the study's foundation.

Assumptions and Technical Specifications - The project relies on several assumptions:

1. Uniform properties of tissue samples across different batches.
2. Reliability and accuracy of the biological tissue microtome and imaging equipment.
3. Effectiveness of deep learning models in interpreting complex biological image data.

Technical details regarding tissue microtome settings, image classification standards, and deep learning architecture are thoroughly described 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*

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].

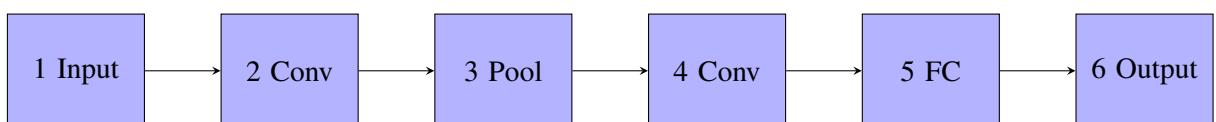
Andelko Hrzenjak also used the HM355S microtome for sectioning pathological endometrial tissues to study the mechanisms of endometrial carcinoma development [4].

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 *Medical Image Classification - Based on Convolutional Neural Networks*

For the problem of medical image classification, deep learning models based on convolutional neural networks have been proven to be one of the most effective methods[8]. 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 [9]. A typical CNN architecture is shown below :[10]



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:[11]

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[12].

2 . 3 *Medical Image Classification - Based on Transfer Learning*

For CNN models, an evident method to enhance deep learning performance is through transfer learning. Sudhakar points out that in facing complex classification issues such as malware classification, deep learning models with a substantial number of parameters significantly outperform traditional machine learning models. However, due to the extensive data and computational resources required by deep learning models, transfer learning emerges as a crucial solution [13].

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[14].

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[15].

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[16].

In Kora's study[17], transfer learning eliminates the need to train models from scratch and requires less data while ensuring the accuracy of medical slice identification. This method has broad application prospects in the problem of medical image classification.

2.4 *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 [18].

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 [19]. 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 [20]. This provides a viable approach for implementing transfer learning, which can greatly enhance the efficiency of tissue image classification and analysis.

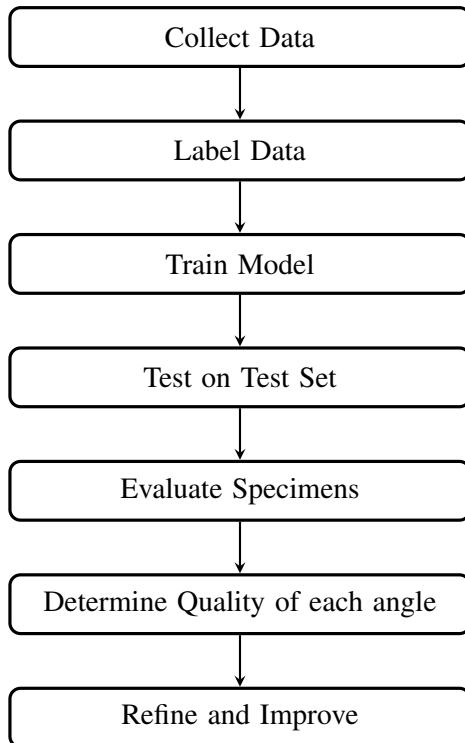
Based on the literature, deep learning technology holds broad prospects for application in image classification and analysis of tissue sections. By leveraging deep learning models, efficient identification and classification of tissue samples can be achieved, providing strong support for optimizing sectioning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

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



Data Collection: Utilize a microtome to slice tissue sections that have been pre-prepared and paraffin-embedded in a biological laboratory following staining procedures. Operate the microtome according to its manual to obtain images of the sections and record the cutting parameters.

Data Labeling: Label the collected image data based on the quality of the sections and their suitability for biological observation and analysis.

Model Training: Experiment with various structures and forms of deep learning convolutional models to train the labeled data. Monitor accuracy and loss functions during the training process to assess model performance.

Testing with Test Set: Evaluate the trained model using a test set to assess its generalization capability.

Specimen Evaluation: Assess an additional prepared test set to observe the practical application effects of the model, calculating the model's prediction accuracy.

Determination of Cutting Yield: Use pre-prepared image data of sections cut at different angles to evaluate with the model, deriving the yield of good cuts at various angles to determine the optimal cutting angle.

Improvements and Enhancements: Based on the evaluation results, adjust and enhance the model to improve its accuracy and generalization capabilities.

3.2 *Image Processing Methods*

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.

3.2.1 Edge Detection for Tissue Slices

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 [21].

The **Sobel operator** is a first-order differential operator that can be used to detect image edges [22]. 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.

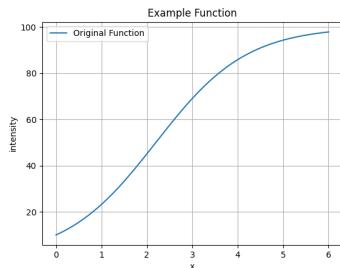


Figure 3.1 $f(x)$

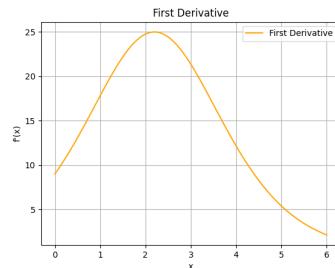


Figure 3.2 $f'(x)$

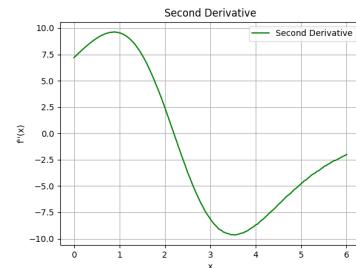


Figure 3.3 $f''(x)$

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 [23], 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 section 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.

3.2.2 *Theresold Segmentation for Tissue Slices*

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 [24].

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

3.3 *Simple Convolutional Neural Network Framework*

Based on the design of convolutional neural network architectures discussed in the literature review, a simple convolutional neural network framework is suitable for classifying images of biological tissue sections in this project. We have constructed three simple convolutional neural network models, labeled as configurations a, b, and c.

Each of these models features a distinct combination of convolutional layers and neuron numbers in both the convolutional and fully connected layers, as summarized in the table below:

Table 3.1 Configuration of the simple CNN model

| Layer Type | Configuration a | Configuration b | Configuration c |
|------------------|------------------|------------------|------------------|
| 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 | - | - | - |

Model a consists of three convolutional layers with 32 neurons each and a fully connected layer with 128 neurons. Model b features three convolutional layers with a progressively increasing number of neurons (16, 32, and 64), and a fully connected layer with 128 neurons. Model c also includes three convolutional layers, each with 32 neurons, but differs in its fully connected layer, which contains 256 neurons.

The primary differences among these models lie in the number of neurons in the convolutional and fully connected layers. In the "Experimental Work/Analytical Investigation/Design" section, these models will be subjected to experiments and performance evaluations to compare their effectiveness.

3.4 *Model Selection of Transfer Learning Methods*

In transfer learning, commonly used pre-trained models include VGG16, VGG19, 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[25].

Table 3.2 displays the number of parameters for models such as the VGG series (VGG16, VGG19) designed by the Visual Geometry Group at the University of Oxford [26], as well as the modular deep learning models InceptionV3 [27][28] developed by Google. These models possess a vast quantity of parameters, enabling them to accurately extract features from complex images. Utilizing the capabilities of these well-trained models allows researchers and practitioners to achieve high performance on specific tasks without the need to train the entire network from scratch, thereby saving time and resources while maintaining high accuracy[29].

Table 3.2 Comparison of Transfer Learning Models

| Model | VGG16 | VGG19 | InceptionV3 |
|-----------------------------|-------------|-------------|-------------|
| Number of Parameters | 138,357,544 | 143,667,240 | 23,851,784 |

The primary differences between these models are as follows:

- VGG16 (Model 3a) and VGG19 (Model 3b) are similar, but VGG19 includes three additional convolutional layers, potentially offering better feature extraction capabilities.
- InceptionV3 (Model 3c) incorporates Inception modules, enabling it to capture a broader range of features across multiple scales, providing a more complex and possibly more effective feature extraction mechanism.

Considering that these models are all trained on the ImageNet dataset, and the features of the biological tissue slice dataset are quite different from the ImageNet dataset. As for how to determine the best model, it needs to be verified through experiments. In the section of "Experimental Work/Analytical Investigation/Design", these models will be experimented and performance evaluated, comparing their effects.

4 Experimental work/ analytical investigation/ design

4.1 Data Collection

In this experiment, pre-prepared and stained paraffin-embedded tissue sections from a biological laboratory were utilized (Figure 4.3). Slicing was performed using an HM355s automatic microtome, following the operating manual at various cutting angles, and cutting parameters were recorded. After slicing, the tissue samples were placed on a flat surface and dried (Figure 4.4), followed by imaging under a VHX7000 microscope to capture electronic image data (Figure 4.5).

The microtome is shown in Figure 4.1, and its working principle (using a tooth as an example) is illustrated in Figure 4.2 [30]. During the cutting process, the original sample is securely fixed for slicing, and precise cuts are made by a blade. After cutting, the sections are extended and flattened by a water flow on a designated surface, then moved to a storage area. After being retrieved and dried, the sections are placed on the microscope stage where the microscope automatically focuses and captures the image data.

The cutting parameters are as follows: the mode is set to continuous, the feed rate is 5.0, the trimming value is 25, the speed setting is 32, the water flow rate is 7.5, and the water temperature is approximately 36 degrees Celsius. The cutting angles range from 8 to 12 degrees. The magnification of the microscope is 20x20, and the resolution is 2880x2160.

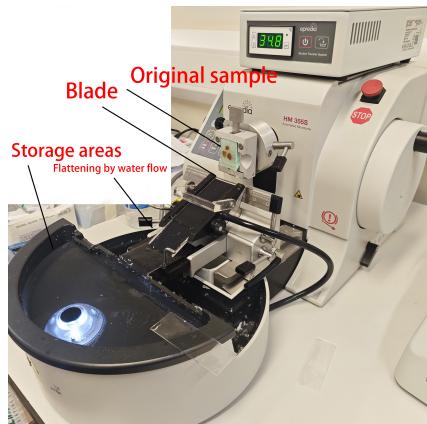


Figure 4.1 Microtome

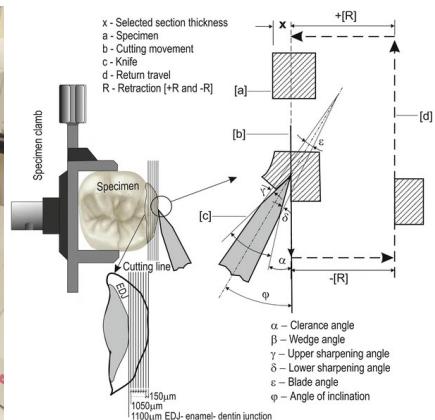


Figure 4.2 Cutting Principles

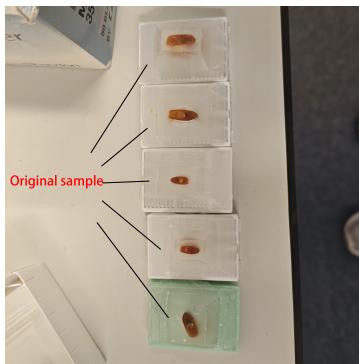


Figure 4.3 Original sam- ple



Figure 4.4 Tissue sec- tions

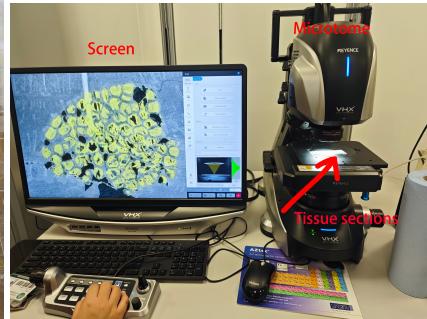


Figure 4.5 Microscope

4.2 Data Labeling

For this experiment, the dataset is 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, they are classified into two additional specific categories: **horizontal line**(Figure 4.6) and **vertical line**(Figure 4.7).

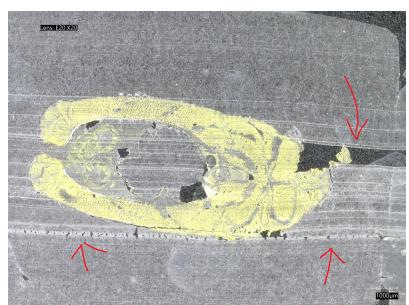


Figure 4.6 horizontal line

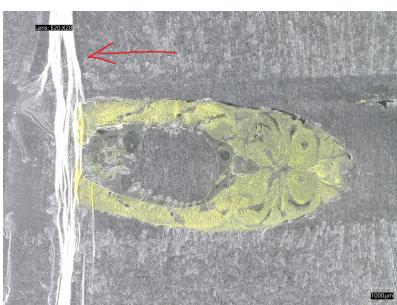


Figure 4.7 vertical line

Additionally, some images were noted to have a significant rotational angle at the time of sampling. These instances are categorized separately as **slope**(Figure 4.8). Finally, any images that do not fit into the aforementioned categories but still show irregularities(Excessive changes in brightness) are labeled as **other**(Figure 4.9).



Figure 4.8 slope

Figure 4.9 other

Figure 4.10 normal

An example of a normal slice that meets observational requirements is shown in Figure 4.10.

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 is uncertain given the complexity of the images, it's advisable to begin with basic CNN architectures to assess the characteristics of the dataset. To do this, the simple CNN models a, b, and c described in the "Methodology and Theory" section will be used for training.

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.

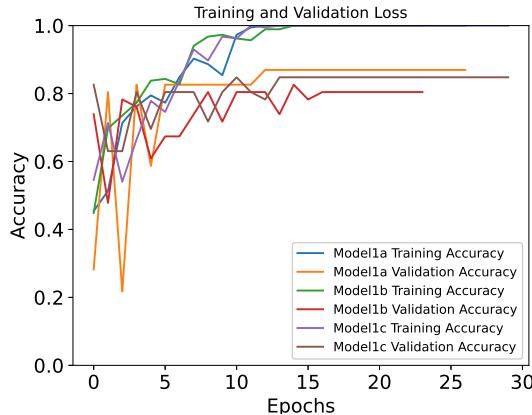


Figure 4.11 Accuracy of Model 1

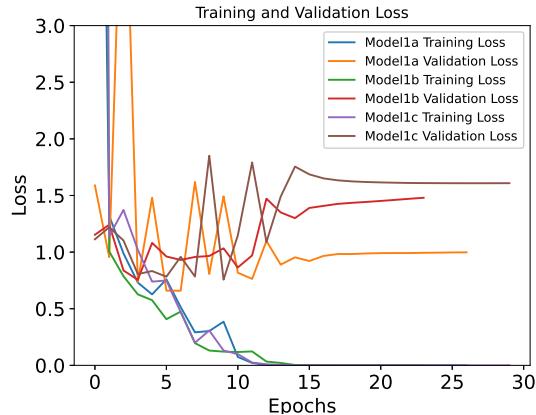


Figure 4.12 Loss of Model 1

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 3.2.1, the principle of edge detection involves identifying changes in pixel intensity (gradients) to determine edges within an image.

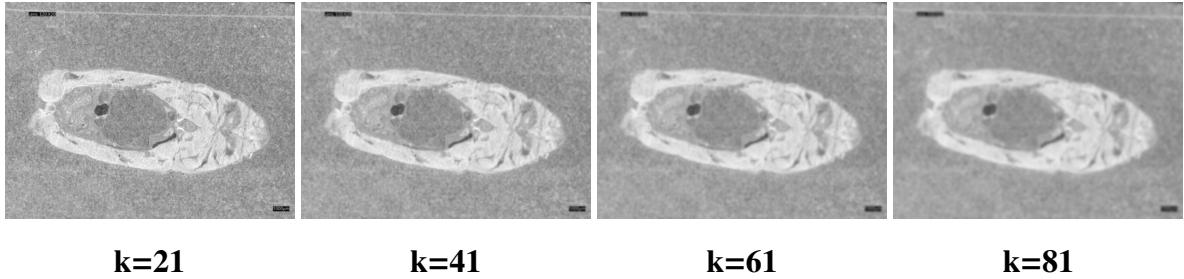


Figure 4.13 Images post-Gaussian blur in different kernel sizes

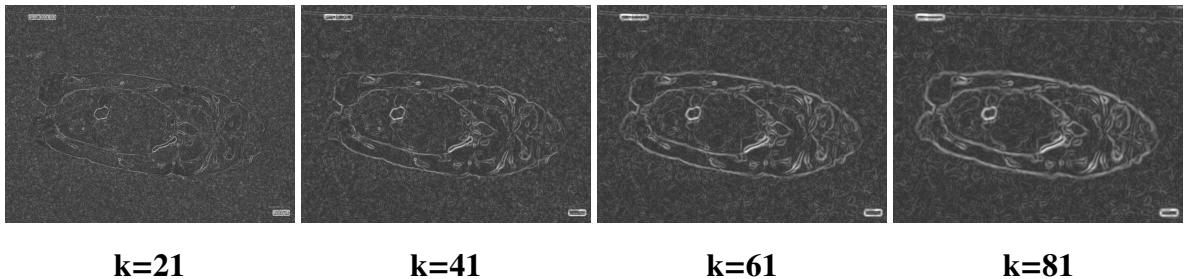


Figure 4.14 Images post-Sobel operator in different kernel sizes

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[31]. 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.

In Figure 4.13, 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. In Figure 4.14 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.

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

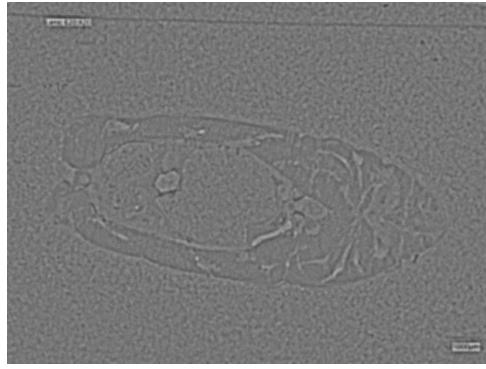
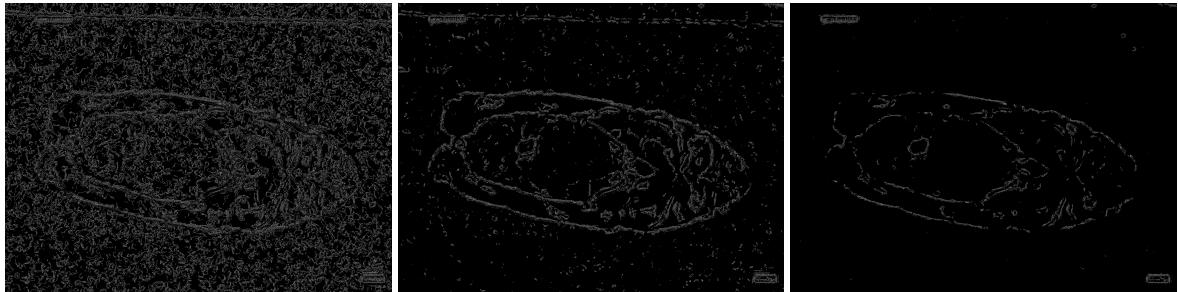


Figure 4.15 Result of applying the Laplacian operator

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:



Low: 2, High: 5

Low: 4, High: 10

Low: 6, High: 15

Figure 4.16 Images post-Canny operator with different thresholds

Among the three results from the Canny edge detection, the performance was best when the low threshold was set at 4 and the high threshold at 10. This setting managed to preserve most of the edge details while effectively eliminating most of the noise. Therefore, 4 and 10 were selected as the thresholds 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 4.17. 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 4.18. 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 4.17 Enhanced Contrast

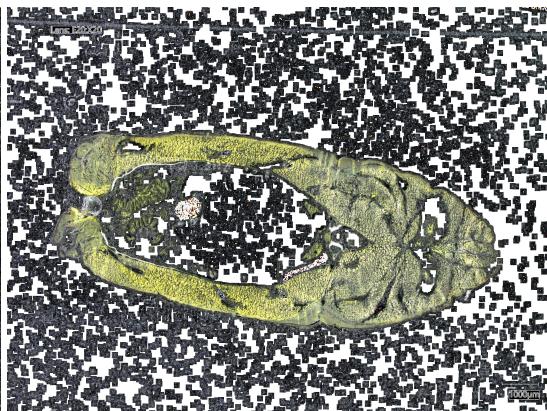


Figure 4.18 Preserved Yellow Pixels

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 4.19.

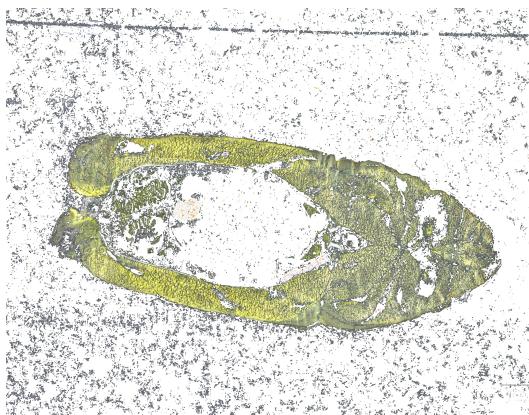


Figure 4.19 Final image

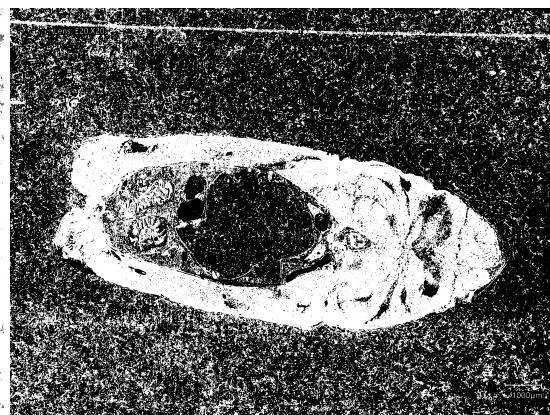


Figure 4.20 The fingerprint algorithm

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 *Performing Threshold Segmentation Using 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 4.20.

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.

4.4.4 *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:

- **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 4.21 and Figure 4.22), showcasing the training and validation accuracy and loss for Models 2a, 2b, and 2c.

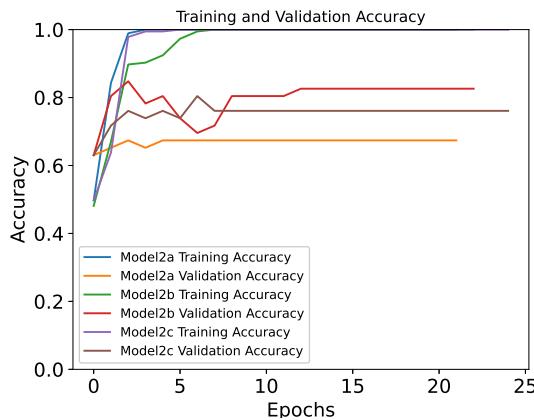


Figure 4.21 Accuracy of Model 2

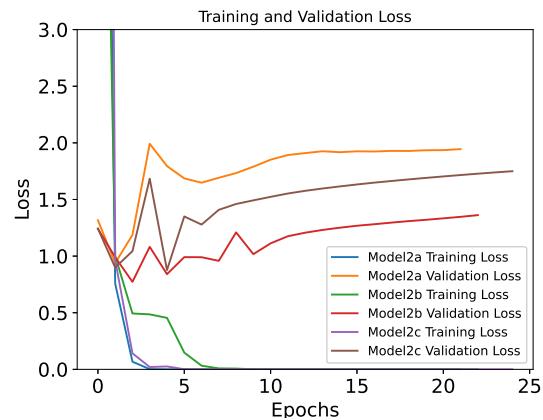


Figure 4.22 Loss of Model 2

4.5.1 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:

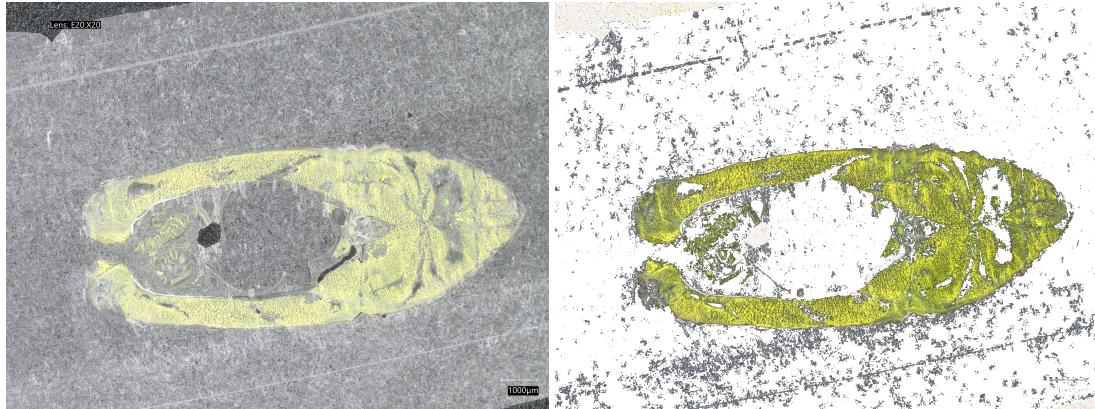


Figure 4.23 Original Image

Figure 4.24 Threshold Segmentation

In Figure 4.24, 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, transfer 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*

In this section, we will utilize pretrained models such as VGG16, VGG19, and InceptionV3, mentioned in previous sections, for transfer learning. To prevent overfitting and optimize the transfer learning process, specific fine-tuning strategies are implemented as follows:

- Early stopping is utilized.
- The learning rate is set relatively low, with VGG16 and VGG19 models set at 1e-5 due to their larger parameter counts. For InceptionV3, which has fewer parameters, the learning rate is slightly higher at 1e-4.
- 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.

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

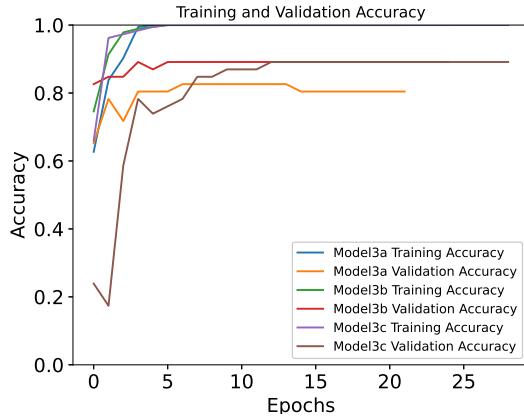


Figure 4.25 Accuracy of Model 3

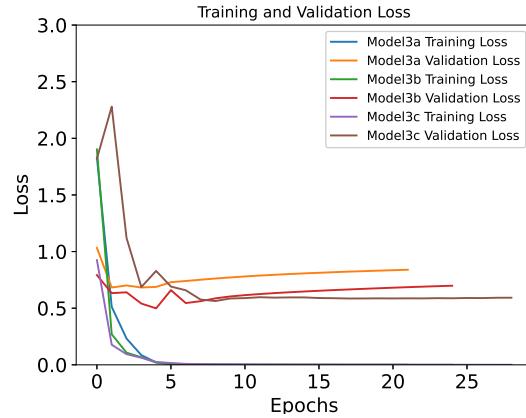


Figure 4.26 Loss of Model 3

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.

4.6.1 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 among 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 may be that Model 3 is based on a deep convolutional network pretrained on a large-scale image dataset, which facilitates more effective feature extraction and the development of a robust feature space.

Consequently, Model 3c (InceptionV3) has been chosen as our final model for further application and testing. This model excels in generalizing to new, unseen data, making it highly suitable for complex tasks such as medical and tissue image analysis, where accuracy and reliability are crucial.

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 Testing 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 Predict Accuracy on Test Set

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

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 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:

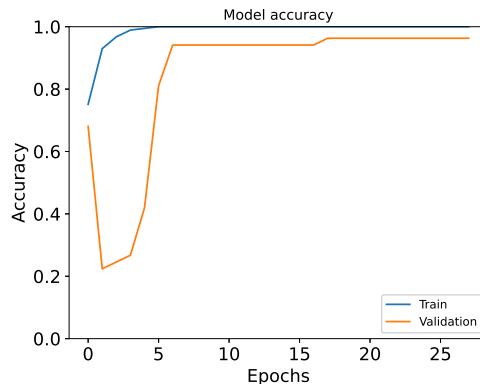


Figure 5.1 Accuracy of Model 4

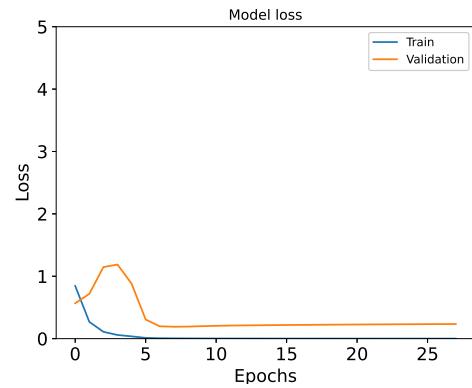


Figure 5.2 Loss of Model 4

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 0.2, 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:

Table 5.2 Model 4 Predict Accuracy on Test Set

| Category | Normal | Horizontal Line | Vertical Line | Slope | Other |
|-------------|--------|-----------------|---------------|-------|-------|
| accuracy(%) | 98.4 | 96.7 | 85.6 | 96.5 | 96.5 |

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 Determining 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:

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 |

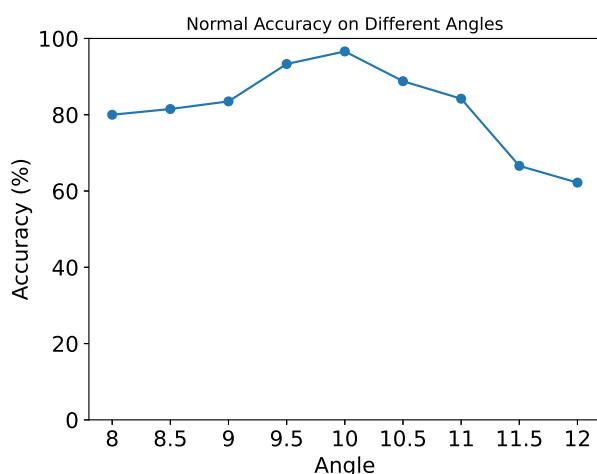


Figure 5.3 Model Accuracy on Different Angle

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 5.3, 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.

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: (as shown in Figure 5.4)

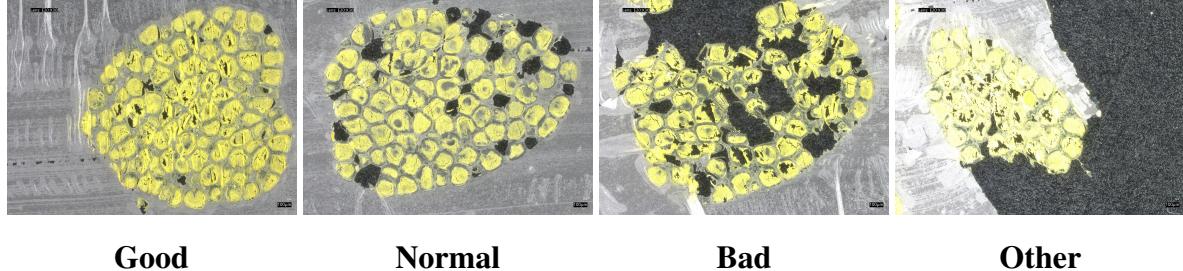


Figure 5.4 Four categories of fish lung tissue

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 5.5 and Figure 5.6.

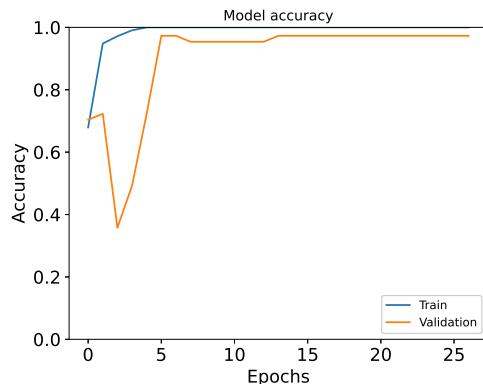


Figure 5.5 Accuracy of Model 5

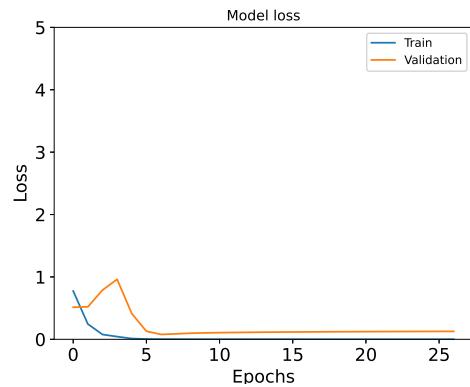


Figure 5.6 Loss of Model 5

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 Table 5.4.

Table 5.4 Model Predict Accuracy on New Test Set

| Category | Bad | Good | Normal | Other |
|-------------|------|------|--------|-------|
| Accuracy(%) | 94.1 | 98.2 | 94.7 | 95.0 |

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, showed below, highlights the structured approach taken to integrate the InceptionV3 architecture effectively within the training framework.

- Input Layer: 1152*864
- Base Model: InceptionV3
- Global Average Pooling Layer
- Fully Connected Layer (Number of nodes based on Labels) - Output Layer
- Learning Rate: 1e-4, Optimizer: Adam
- Loss Function: Cross-Entropy, Performance Metric: Accuracy
- Early Stopping: Enabled

6.2 Future work

6.2.1 Enhancing Classification Methods

Current research indicates that existing classification methods are promising but limited to five categories. By expanding these categories, we could enhance our understanding of how cutting angles impact sample quality and improve the precision of our analyses. A broader classification range would also better predict optimal cutting angles for various tissue types and conditions. Adopting a more detailed classification could facilitate a shift to linear analytical methods, where increased categories serve as discrete points, enabling accurate modeling of relationships between cutting angles and sample quality through linear regression.

Linear Discriminant Analysis (LDA) is particularly useful for refining how we determine optimal cutting angles in relation to tissue quality. This method could simplify the prediction of cutting parameters and improve control over tissue sectioning. However, transitioning to a linear discriminant framework introduces challenges. Unlike binary models that output probabilities, linear regression models require a direct correlation between variables like cutting angles and tissue quality, which might not be inherently linear.

Furthermore, linear models demand significantly more data, which can prolong data collection, increase its complexity, and require more computational power. Current systems using TensorFlow and InceptionV3 are already straining GPU resources, highlighting the need for enhanced hardware. These advancements, crucial for the effective use of linear discriminant analysis, will necessitate substantial time and resources, as well as further theoretical and practical research. For example, Jie Wen's work on "Robust Sparse Linear Discriminant Analysis" integrates sparsity into the LDA model, making it more robust and suitable for complex applications[32].

6.2.2 Exploring the Impact of Other Parameters on Cutting Quality

In previous experiments, we established a model by setting the cutting angle as the independent variable and the cutting quality as the dependent variable. However, in reality, cutting quality may be influenced by other parameters such as cutting speed, feed rate (chip thickness), and tool wear.

In future work, if our focus is on the factors affecting cutting quality, research on these additional variables will be necessary. Indeed, the impact of these parameters on quality can be intuitively represented by a function:

$$Q = f(\theta, v, f, w) \quad (6.1)$$

Here, Q represents cutting quality, θ denotes the cutting angle, v indicates the cutting speed, f stands for the feed rate, and w symbolizes tool wear. As for the specific form of this function, i.e., the weights of each parameter, it will require extensive experimental data for statistical analysis and fitting. This represents yet another challenge.

6.2.3 *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[33]. 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[34]. 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.

6.2.4 *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"[35]. 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 Conclusion

This research significantly enhances the understanding of optimizing biopsy parameters by integrating deep learning with biomedical tissue sectioning devices. Employing the InceptionV3 model adapted through transfer learning, the study showcases a robust framework for high-precision assessment of tissue section quality. This innovative approach not only refines the accuracy of tissue analysis but also revolutionizes the operational methods of tissue sectioning.

The study identified a clear correlation between the cutting angles and the quality of tissue sections, offering a practical method to enhance future section quality. The model's effectiveness across various tissue types, such as fish ovaries and lung tissues, underscores its broad adaptability and potential for widespread application.

However, the research also points out the limitations of traditional image preprocessing techniques. Initial enhancements through preprocessing did not significantly improve performance and sometimes obscured essential details for accurate classification. This suggests that preserving the original image data might be more advantageous than applying extensive preprocessing.

The study proposes enhancements to the tissue sectioning process by expanding classification methods and performance optimization. This includes integrating more classification categories and linear analytical methods such as Linear Discriminant Analysis (LDA) to refine the understanding of the relationship between cutting parameters and sample quality. Furthermore, exploring additional factors that influence cutting quality is also crucial. Future work will focus on optimizing the computational framework and parallel processing, and examining additional parameters, such as cutting speed and feed rate, to enhance the model's predictiveness and the quality of tissue sections. Implementing a real-time feedback system that dynamically adjusts cutting parameters using machine learning is anticipated to drive histological preparation towards full automation, ensuring consistently high-quality tissue sections.

In conclusion, this project not only underscores the critical role of deep learning in advancing biomedical research and applications but also sets the stage for substantial improvements in tissue sectioning technology. These advancements could significantly enhance the yield and efficiency of tissue sample processing, offering new strategies and methodologies for future developments in tissue sectioning technology and having a lasting impact on biomedicine.

7 Project management, consideration of sustainability and health and safety

7.1 Project management

The primary tool for managing the project was a Gantt chart, as illustrated in Figure 7.1. This chart served as a visual tool to set and track realistic timelines for completing various sections of the project. It was updated throughout the project to reflect changes, including additions and deletions of project segments.

Regular updates to the chart, combined with weekly meetings with the supervisor, ensured the project remained on schedule.



Figure 7.1 Gantt chart

7.2 Health and Safety

The only potential safety hazards in the laboratory stem from the operation of experimental equipment, particularly during the preparation of biological sections and the use of microscopes. The preparation of sections involves the use of extremely sharp tissue slicers, necessitating great caution to avoid cutting fingers. We adhere strictly to laboratory protocols, wearing gloves and avoiding direct contact with blades. Moreover, to minimize injuries and ensure consistency in experiments, we mostly employ the automatic cutting function of the machine rather than manual cutting. This means that during the section preparation process, the experimenter is only responsible for adjusting the blade while the machine handles the rest.

While capturing images under the microscope, the primary risk involves accidentally touching the microscope's lenses, which could damage them. To prevent this, we minimize manual handling and utilize the microscope's automatic adjustment features to control focus and aperture. This approach helps prevent contamination of the lens during manual adjustments, thereby maintaining the quality of imaging.

7.3 *Sustainability*

Concerning sustainability, the only outputs from our experiments are biological tissue sections and glass slides. These used materials are disposed of strictly according to waste segregation protocols. Biological tissue sections are scraped off and placed in biological waste bins, whereas glass slides, given their sharpness, are wrapped and placed in bins designated for glass. These practices ensure a clean laboratory environment and protect personnel from injuries caused by sharp objects.

Additionally, we implement basic energy-saving measures related to the use of electricity and other resources in the lab. Ensuring that equipment is completely turned off when not in use helps to minimize energy wastage. Although these measures may seem minor, they contribute significantly to environmental protection and energy conservation over the long term.

References

- [1] K. P. Murphy, "Machine learning: a probabilistic perspective," Cambridge: The MIT Press, pp. 32, 2012.
- [2] M. Zimmermann, J. Lampe, S. Lange, I. Smirnow, A. Königsrainer, C. Hann-von-Weyhern, F. Fend, M. Gregor, M. Bitzer, and U. M. Lauer, "Improved reproducibility in preparing precision-cut liver tissue slices," *Cytotechnology*, vol. 61, no. 3, pp. 145-152, 2009. <https://doi.org/10.1007/s10616-009-9246-4>
- [3] E. Klimuszko, K. Orywal, T. Sierpinska, J. Sidun, and M. Golebiewska, "Evaluation of calcium and magnesium contents in tooth enamel without any pathological changes: in vitro preliminary study," *Odontology*, vol. 106, no. 4, pp. 369-376, 2018. <https://doi.org/10.1007/s10266-018-0353-6>
- [4] A. Hrzenjak, F. Moinfar, F. A. Tavassoli, B. Strohmeier, M.-L. Kremser, K. Zatloukal, and H. Denk, "JAZF1/JJAZ1 gene fusion in endometrial stromal sarcomas: molecular analysis by reverse transcriptase-polymerase chain reaction optimized for paraffin-embedded tissue," *The Journal of Molecular Diagnostics*, vol. 7, no. 3, pp. 388-395, 2005. [https://doi.org/10.1016/S1525-1578\(10\)60568-5](https://doi.org/10.1016/S1525-1578(10)60568-5)
- [5] L. Song, M. Mino, J. Yamak, V. Nguyen, D. Lopez, V. Pham, A. Fazelpour, V. Le, D. Fu, M. Tippin, E. Uchio, and X. Zi, "Flavokawain A reduces tumor-initiating properties and stemness of prostate cancer," *Frontiers in Oncology*, vol. 12, 2022. <https://doi.org/10.3389/fonc.2022.943846>
- [6] T. Azuma, Y. Murata, Y. Hokii, S. Akiyama, and Y. Shinozaki, "Evaluation of microcracks formed by grinding in machinable lithium silicate," *Dental Materials*, vol. 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] F. A. Veer, "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, 2022. eBook ISBN: 9781003023555.
- [8] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen, "Medical image classification with convolutional neural network," in 13th International Conference on Control Automation Robotics & Vision (ICARCV), Singapore, pp. 844-848, 2014.
- [9] H. Zhou and Q. Sun, "Research on Principle and Application of Convolutional Neural Networks," IOP Conference Series: Earth and Environmental Science, vol. 440, no. 4, p.042055, 2020. doi: 10.1088/1755-1315/440/4/042055. <https://dx.doi.org/10.1088/1755-1315/440/4/042055>
- [10] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat, "CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope," *Electronics*, vol. 10, no. 20, p.2470, 2021.
- [11] D. M. Montserrat, Q. Lin, J. Allebach, and E. J. Delp, "Training Object Detection And Recognition CNN Models Using Data Augmentation," *Electronic Imaging*, vol. 29, no. 10, pp. 27-27, 2017.
- [12] S. M. Anwar, M. Majid, A. Qayyum et al., "Medical Image Analysis using Convolutional Neural Networks: A Review," *Journal of Medical Systems*, vol. 42, p.226, 2018. <https://doi.org/10.1007/s10916-018-1088-1>
- [13] S. K. Sudhakar, "MCFT-CNN: Malware classification with fine-tune convolution neural networks using traditional and transfer learning in Internet of Things," *Future Generation Computer Systems*, vol. 125, pp. 334-351, 2021. <https://doi.org/10.1016/j.future.2021.06.029>
- [14] L. Torrey and J. Shavlik, "Transfer Learning," in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, 1st ed., E. Soria Olivas, J. Guerrero, M. Martinez-Sober, J. Magdalena-Benedito and A. Serrano López, Ed., Hershey, PA: IGI Global, pp. 242-264, 2010. <https://doi.org/10.4018/978-1-60566-766-9.ch011>
- [15] Z. N. K. Swati, Q. Zhao, M. Kabir, F. Ali, Z. Ali, S. Ahmed, and J. Lu, "Brain tumor classification for MR images using transfer learning and fine-tuning," *Computerized Medical Imaging and Graphics*, vol. 75, pp. 34-46, 2019. <https://doi.org/10.1016/j.compmedimag.2019.05.001>
- [16] E. C. Orenstein and O. Beijbom, "Transfer Learning and Deep Feature Extraction for Planktonic Image Data Sets," in 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, pp. 1082-1088, 2017. doi: 10.1109/WACV.2017.125.
- [17] P. Kora, C. P. Ooi, O. Faust, U. Raghavendra, A. Gudigar, W. Y. Chan, K. Meenakshi, K. Swaraja, P. Plawiak, and U. R. Acharya, "Transfer learning techniques for medical image analysis: A

- review," *Biocybernetics and Biomedical Engineering*, vol. 42, no. 1, pp. 79-107, 2022. <https://doi.org/10.1016/j.bbe.2021.11.004>
- [18] L. Guachi-Guachi, J. Ruspi, P. Scarlino, A. Poliziani, S. Ciancia, D. Lunni, G. Baldi, A. Cavazzana, A. Zucca, M. Bellini, G. A. Pedrazzini, G. Ciuti, M. Controzzi, L. Vannozzi, and L. Ricotti, "Convolutional neural networks applied to microtomy: Identifying the trimming-end cutting routine on paraffin-embedded tissue blocks," *Engineering Applications of Artificial Intelligence*, vol. 126, p. 106963, 2023. <https://doi.org/10.1016/j.engappai.2023.106963>
- [19] V. Andrarczyk and P. F. Whelan, "Deep Learning in Texture Analysis and Its Application to Tissue Image Classification," in A. Depeursinge, O. S. Al-Kadi, and J. R. Mitchell, Eds., *Biomedical Texture Analysis*, Academic Press, pp. 95-129, 2017. ISBN 9780128121337. <https://doi.org/10.1016/B978-0-12-812133-7.00004-1>
- [20] Y. Xu, Z. Jia, L. B. Wang, Y. Ai, F. Zhang, M. Lai, E. I. C. Chang, "Large scale tissue histopathology image classification, segmentation, and visualization via deep convolutional activation features," *BMC Bioinformatics*, vol. 18, no. 1, p. 281, 2017. <https://doi.org/10.1186/s12859-017-1685-x>
- [21] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to OpenCV," in 2012 Proceedings of the 35th International Convention MIPRO, Opatija, Croatia, pp. 1725-1730, 2012. Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6240859&isnumber=6240598>
- [22] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*, O'Reilly Media, Inc., 2008.
- [23] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, 1986. doi: 10.1109/TPAMI.1986.4767851. Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4767851&isnumber=4767846>
- [24] Y. Yue and J. Zhu, "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, 2017. doi: 10.1109/ICIVC.2017.7984539. Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7984539&isnumber=7984442>
- [25] M. Shaha and M. Pawar, "Transfer Learning for Image Classification," in 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, pp. 656-660, 2018. doi: 10.1109/ICECA.2018.8474802.
- [26] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv preprint arXiv:1409.1556, 2014. Available at: <https://arxiv.org/abs/1409.1556>

- [27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," arXiv preprint arXiv:1512.00567, 2015. Available at: <https://arxiv.org/abs/1512.00567>
- [28] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," arXiv preprint arXiv:1610.02357, 2016. Available at: <https://arxiv.org/abs/1610.02357>
- [29] S. R. Shah, S. Qadri, H. Bibi, S. M. W. Shah, M. I. Sharif, and F. Marinello, "Comparing Inception V3, VGG 16, VGG 19, CNN, and ResNet 50: A Case Study on Early Detection of a Rice Disease," *Agronomy*, vol. 13, no. 6, p. 1633, 2023. <https://doi.org/10.3390/agronomy13061633> [Accessed 30 April 2024].
- [30] E. Klimuszko, K. Orywal, T. Sierpinska, et al., "Evaluation of calcium and magnesium contents in tooth enamel without any pathological changes: in vitro preliminary study," *Odontology*, vol. 106, pp. 369-376, 2018. doi: 10.1007/s10266-018-0353-6. <https://doi.org/10.1007/s10266-018-0353-6>.
- [31] E. S. Gedraite and M. Hadad, "Investigation on the effect of a Gaussian Blur in image filtering and segmentation," in Proceedings ELMAR-2011, Zadar, Croatia, pp. 393-396, 2011. Available at: <https://ieeexplore.ieee.org/abstract/document/6044249>.
- [32] J. Wen et al., "Robust Sparse Linear Discriminant Analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 390-403, 2019. doi: 10.1109/TCSVT.2018.2799214. Available at: <https://doi.org/10.1109/TCSVT.2018.2799214>.
- [33] O-C. Novac, M. C. Chirodea, C. M. Novac, N. Bizon, M. Oproescu, O. P. Stan, C. E. Gordon, "Analysis of the Application Efficiency of TensorFlow and PyTorch in Convolutional Neural Network," *Sensors*, vol. 22, no. 22, 8872, 2022. Available at: <https://doi.org/10.3390/s22228872>.
- [34] P. Fua and K. Lis, "Comparing Python, Go, and C++ on the N-Queens Problem," 2020. Available at: <https://arxiv.org/abs/2001.02491>.
- [35] L. Guachi-Guachi, J. Ruspi, P. Scarlino, A. Poliziani, S. Ciancia, D. Lunni, G. Baldi, A. Cavazzana, A. Zucca, M. Bellini, G. A. Pedrazzini, G. Ciuti, M. Controzzi, L. Vannozzi, and L. Ricotti, "Convolutional neural networks applied to microtomy: Identifying the trimming-end cutting routine on paraffin-embedded tissue blocks," *Engineering Applications of Artificial Intelligence*, vol. 126, p. 106963, 2023. Available at: <https://www.sciencedirect.com/science/article/pii/S0952197623011478>.

All code will be open sourced on **GitHub**.