Group 4 - Deep Learning for Ovarian Cancer Subtype Classification

Abiel Kim[301375278], Jisoo Im[301376613], Pabil Adhikari[301400529], Nardos Solomon[301586148], and Jordan Clough[301436504]

{abielk,jia24,padhikar,nms15,jrc30}@sfu.ca
CMPT340 Spring 2025, Prof. Hamarneh

Abstract. Detecting ovarian cancer is a complex task that often relies on the expert interpretation of histopathological images. As early detection significantly improves survival rates, achieving fast and accurate classification is critical. In this project, we developed a deep learning model to classify five subtypes of ovarian cancer, high-grade serous carcinoma (HGSC), clear cell carcinoma (CC), endometrioid carcinoma (EC), low-grade serous carcinoma (LGSC), and mucinous carcinoma (MC). We used a filtered dataset from the UBC-OCEAN ovarian cancer subtype classification challenge, which focuses on regions essential for subtype discrimination. Our model architecture was based on ResNet-18, trained with the Adam optimizer and a Cosine Annealing Learning Rate Scheduler. We evaluated model performance using accuracy, confusion matrix analysis, and epoch-based training trends. The model achieved a test accuracy of more than 99%. In conclusion, our approaches demonstrate the potential of AI-assisted tools to support the early and reliable diagnosis of ovarian cancer subtypes.

Keywords: Ovarian cancer · Histopathological images · Data augmentation · ResNet-18 · Adam optimizer · Cosine Annealing Learning Rate Scheduler

1 Introduction

Ovarian cancer (OC) is the fifth most common cause of cancer-related death in women and is often diagnosed at a later stage, particularly in postmenopausal patients. The disease affects approximately 40 out of every 100,000 women annually over the age of 50 years. Early detection is critical; The five-year survival rate increases drastically from 3% in Stage IV to 90% in Stage I. Therefore, accurate and early classification of ovarian cancer subtypes is essential to improve diagnosis, treatment planning, and overall patient outcomes [1].

This project presents a deep learning application designed to classify five different subtypes of ovarian cancer, high-grade serous carcinoma (HGSC), clear cell carcinoma (CC), endometrioid carcinoma (EC), low-grade serous carcinoma (LGSC), and mucinous carcinoma (MC), based on histopathological images. Our

application aims to help medical professionals and researchers in oncology, the field of medicine that focuses on the study and treatment of cancer, by automating the identification of these subtypes, which can be challenging due to the overlap of morphological features, meaning the visual characteristics and structure of cancer cells. It is intended for use by oncologists, medical researchers, hospitals, and clinical practitioners, providing valuable support in diagnosis and second opinion scenarios.

To build and train our model, we use a publicly available dataset extracted from the UBC-OCEAN ovarian cancer subtype classification challenge. The data set includes over 25,000 JPEG images across the five subtypes, with each image formatted as a 3-channel RGB tensor. Our application aims to assist cancer physicians and researchers in automating the identification of these subtypes, which can be challenging due to overlapping morphological features and the subjective nature of manual analysis. An accurate subtype classification is crucial not only for diagnosis, but also for determining appropriate treatment plans and predicting patient outcomes. Using deep learning techniques on histopathological image data, our tool can reduce diagnostic variability, improve consistency, and potentially speed up workflow in clinical settings. It is intended for use by oncologists, medical researchers, hospitals, and clinical practitioners, providing valuable support in diagnosis, research, training, and second opinion scenarios where expert consensus is needed.

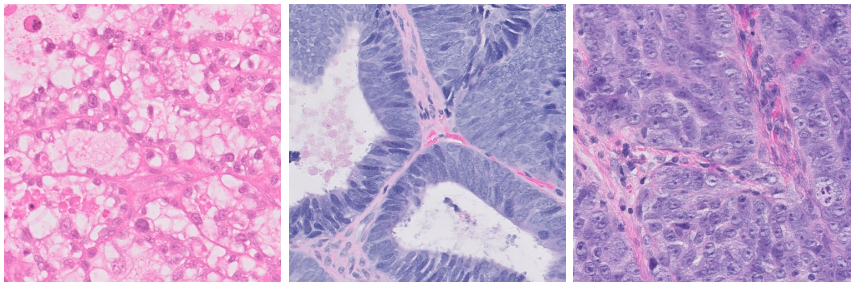
Several previous studies have explored this dataset using a variety of machine learning and deep learning techniques. One recent study used EfficientNet-B0 for feature extraction combined with a fine-tuned k-nearest neighbor (KNN) classifier, achieving exceptional accuracy on a subset of 725 images [2]. Another study evaluated attention-based multiple instance learning classifiers and histopathology foundation models trained on more than 1,800 whole slide images, reaching balanced accuracies of up to 89% to 97% depending on the model and evaluation setup [3]. These studies highlight both the potential and ongoing challenges in this domain, particularly in the construction of generalizable models with efficient computation and robust classification accuracy.

The following sections detail the components of this report:
Section 2, **Materials**, provides a detailed description of the data set used in this project, including its structure, class distribution, image formats, and origin. Section 3, **Methods**, outlines the proposed deep learning approaches, presenting model architecture choices, pre-processing steps, and training strategies, along with relevant diagrams and algorithmic details. Section 4, **Results**, showcases both qualitative and quantitative results through figures, tables, and performance metrics, highlighting the effectiveness of the implemented methods. Section 5, **Accomplishments**, summarizes what we have achieved and learned, including the challenges we encountered and how we addressed or adapted to them. Section 6, **Contributions**, presents a clear breakdown of individual and

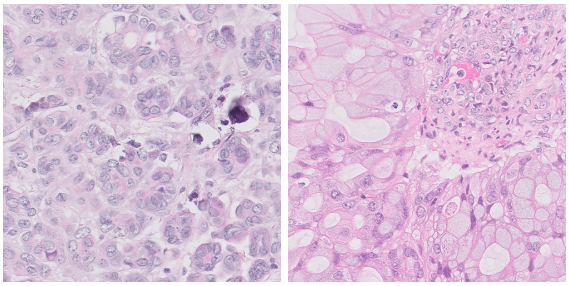
external contributions to the project, detailing who did what and what tools or code were provided or developed. Section 7, **Conclusions and Discussions**, reflects on the overall project, offering a concise summary and a critical analysis of our approach, results, and limitations. Finally, Section 8, **Future Work**, suggests potential directions for further development, whether by us or by future students who may build on this project.

2 Materials

The data used in this project consists of pre-extracted 500x500 pixel image patches derived from the UBC-OCEAN Ovarian Cancer Subtype Classification and Outlier Detection Dataset, hosted on Kaggle, and its associated supplemental masks dataset. The original data set was developed for a competition aimed at improving the machine learning-based classification of ovarian cancer subtypes to support more accessible and reliable clinical diagnostics.



Clear Cell Carcinoma (CC) | Endometrioid Carcinoma (EC) | High-Grade Serous Carcinoma (HGSC)



Low-Grade Serous Carcinoma (LGSC) | Mucinous Carcinoma (MC)

Fig. 1: Representative histopathological image samples from each ovarian cancer subtype.

Each image patch used in this study is an RGB image in the format of (Height, Width), specifically 500x500. These patches were extracted from larger

Table 1: Number of Images per Ovarian Cancer Subtype in the Dataset

dataset-distribution	Number of Images
Clear Cell Carcinoma(CC)	5,295 (20.76%)
Endometrioid Carcinoma(EC)	6,250 (24.51%)
High-Grade Serous Carcinoma(HGSC)	8,747 (34.30%)
Low-Grade Serous Carcinoma(LGSC)	2,720 (10.66%)
Mucinous Carcinoma(MC)	2,491 (9.77%)

histopathological Whole Slide Images (WSIs) and Tissue Microarrays (TMAs), but only the patches were used for analysis. Importantly, regions marked as cancerous (red) and normal (green) in the supplemental tissue masks were excluded during the extraction process. This filtering was performed to focus on other tissue regions, such as necrotic areas or unannotated zones, which can help reduce class imbalance and bias in the training data. Before training, all images were preprocessed using the following steps:

- **Step 1**: Converted to grayscale to reduce input dimensionality.
- **Step 2**: Resized to 496×496 pixels to ensure uniform input size.
- **Step 3**: Data augmentation applied (random horizontal flips and slight rotations) to improve generalization.
- **Step 4**: Normalized using mean = 0.7276 and std = 0.1001, calculated from the datas et.

We split the data set into training and testing sets using an 80:20 ratio, resulting in 20,402 training samples and 5,101 testing samples. PyTorch's `ImageFolder` and `DataLoader` utilities were used for efficient data loading and batching.

3 Methods

For our project, we used the Convolution Neural Network (CNN) architecture and transfer learning to perform multiclass classification of histopathological images of ovarian cancer. The methodology consists of the following stages: (1) data aquisition and preprocessing, (2) data augmentation and normalization, (3) model architecture and modification, (4) training strategy and optimization, and (5) evaluation metrics and visualization. The process of data aquisition, preprocessing and augmentation have been mentioned in the previous section of this report.

3.1 Model Architecture and Transfer Learning

We employed the use of ResNet-18, a type of CNN architecture, as the base model. ResNet-18 was chosen for its ability to find the balance between depth, performance, and computational efficiency [4]. This was especially important in

our case, as we had a smaller data set size and limited GPU resources that could be utilized.

The model was initialized using ImageNet-pretrained weights. This refers to the parameters that have been trained on the ImageNet dataset which is used to capture general visual features [5]. We used these pretrained weights to take advantage of learned low- and mid-level features. Some low-level features include edges of the image, texture and color contrast which is detected in the first few layers of the model using small convolution filters [6]. In the intermediate layers some of the features correspond to the shape of the cell nuclei, cytoplasm texture and cell organization. At this stage, the images are not yet associated with the subtype of cancer. The model was then modified to fit our goal of classifying between 5 different ovarian cancer cells by implementing a final fully connected layer. The original FC layer was replaced with a linear layer with 5 output neurons. This was randomly initialized and trained from scratch. Furthermore, the earlier layers were fine-tuned to improve the accuracy of recognizing specific details of the images.

3.2 Model Training and Optimization

The model was trained for 25 epochs using a batch size of 32. Since this task involves classifying images into multiple ovarian cancer subtypes, we used Cross Entropy Loss, which is well-suited for multiclass classification problems. This loss function measures how different the predicted probability distribution is from the actual labels and encourages the model to assign high confidence to the correct class.

For optimization, we used the Adam optimizer, which is widely used in deep learning because it combines the strengths of two popular methods: momentum and adaptive learning rates. Adam automatically adjusts the learning rate for each parameter during training, helping the model converge faster and more reliably, especially when working with complex image data.

To further improve learning, we used a Cosine Annealing LR Scheduler that gradually decreases the learning rate over time. Starting with a higher learning rate allows the model to explore a wide range of possible solutions, while reducing it later in training helps fine-tune the model for better accuracy. This approach helps the model to converge more effectively and avoid getting stuck in suboptimal solutions.

3.3 Performance Evaluation and Visualization

To assess model performance, several evaluation metrics and visualization techniques were applied:

– **Overall Accuracy:** Measures the proportion of correctly classified image patches in the total number of test samples.

- **Confusion Matrix:** Provides detailed insight into the model's performance across all ovarian cancer subtypes, highlighting both strengths and areas of confusion between specific classes.
- **Performance Over Epochs:** Training and validation accuracy and loss were tracked throughout the 25 epochs. These trends were visualized to monitor the convergence behavior and to detect signs of overfitting or underfitting.

These metrics together offer a comprehensive evaluation of model performance, both quantitatively and qualitatively.

4 Results

We evaluated our model's performance using training loss, accuracy plots, and a confusion matrix.

Training Performance

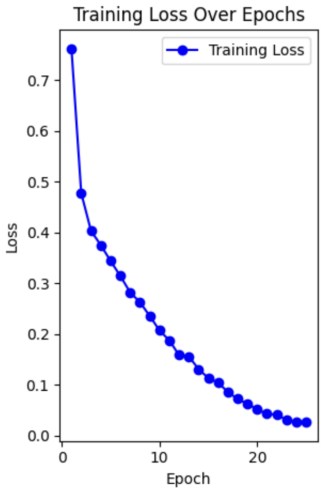


Fig. 2: Training loss over epochs.

The training loss (Figure 1) shows a smooth and steady decline over 25 epochs, dropping from above 0.7 to nearly 0. This consistent decrease indicates stable convergence, which confirms that the model successfully minimized the classification errors in the training set.

Figure 2 shows the accuracy on both validation sets and test in all training epochs. Accuracy improves quickly within the first 10 epochs and gradually increases to almost 99%. The close tracking between validation and test curves

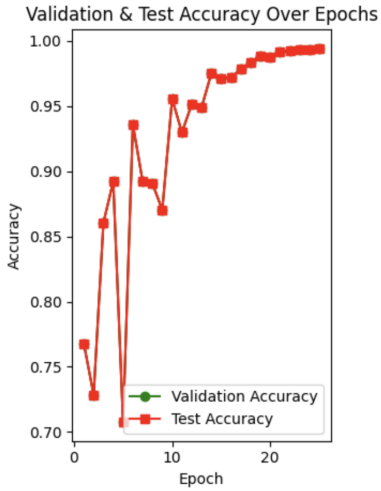


Fig. 3: Validation and test accuracy over epochs.

suggests strong generalization with minimal overfitting.

Classification Performance

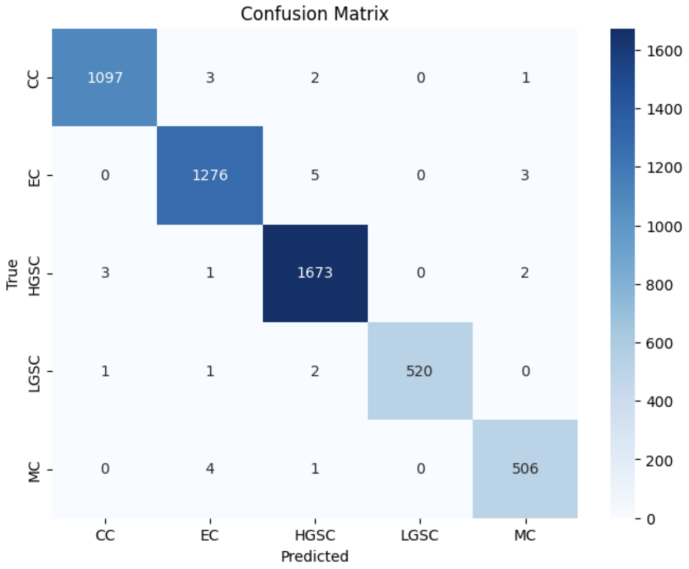


Fig. 4: Confusion matrix on test set.

The confusion matrix (Figure 3) illustrates the performance of the model in the five subtypes of ovarian cancer. The majority of predictions lie along the diagonal, indicating high accuracy for all classes. In particular:

- **CC(Clear Cell Carcinoma)**: the class had 1,103 tests, with 1,097 correctly predicted, 0.988 (99.46% accuracy).
- **EC**: correctly classified 1,276 out of 1,284 samples, resulting in 0.9938 (99.38%) accuracy.
- **HGSC(High-Grade Serous Carcinoma)**: the most represented class with 1,679 test samples, the model correctly predicted 1,673, achieving an accuracy of approximately 0.9938 (99.64%).
- **LGSC**: 520 out of 524 samples were accurately classified, giving an accuracy of 0.9924 (99.24%).
- **MC**: the smallest class in the dataset, the model achieved 0.9902 (99.02%) accuracy, correctly predicting 506 out of 511 samples.

Overall, these results demonstrate the effectiveness of the model in learning different patterns across cancer subtypes and applying that knowledge to unseen data with high confidence. Our model was able to achieve an accuracy higher than 99%.

5 Accomplishments

For our project, we were able to successfully classify five subtypes of ovarian cancer cells by applying different techniques within deep learning. Our model achieved $> 99\%$ for validation and test datasets by integrating and fine-tuning the ResNet-18 model. This high level of accuracy demonstrates our model's strong ability to generalize features across a variety of classes.

We developed a complete classification framework using PyTorch. In turn, this allowed us to incorporate the use of data loading from Kaggle, data augmentation, model training, fine-tuning, evaluation metrics, and generate visuals. The 99% validation and test accuracy exhibits the high performance of the model.

Furthermore, the training curves (Figure 1 & 2) and confusion matrix (Figure 3) in the results section also suggest low overfitting coupled with high performance. In terms of achievements,implementing advanced training strategies such as cosine annealing learning rate scheduler, randomized data augmentation, and automated checkpoints were vital to ensure a robust model. Additionally, we performed visual analysis of the trained model's performance through the use of training curves and a confusion matrix. The use of such metrics aided in identifying class-wise performance and learning behavior. Overall, we were able to achieve minimal misclassifications among certain classes, HGSC and LGSC, that would otherwise be more difficult to differentiate.

Throughout the project, we were introduced to many new concepts. The main lessons learned are as follows. ResNet-18 is still very effective in classifying histopathological images despite being pre-trained on natural images (ImageNet) as long as you implement the appropriate fine-tuning. Furthermore, data augmentation is essential for improving generalization, especially when there are limited datasets. This is because data augmentation is a technique used to increase the diversity of a dataset [7]. Additionally, in terms of evaluation metrics, training loss is not the only important measure. In fact, monitoring validation accuracy along side can greatly prevent overfitting within the model. It also allows for earlier detection of issues with generalization. Incorporating visualization can further improve the accuracy of a model since it provides more information. For example, by generating a confusion matrix you are able to assess any imbalances and recognize which classes experience more misclassifications.

Initially, we faced challenges with accessing and loading the data from Kaggle (an external source) into the appropriate directories for the five subclasses. However, we overcame this using direct local path handling where it checks if the notebook is run in Google Colab or in a local environment. Another issue we faced throughout the project was our limited GPU. This meant some members had to use Google Colab's free T4 GPU which was still very slow while running the epochs. Overall, training time took longer and members with a greater GPU capacity took the lead in that regard. In the earlier stages of training the subtle differences between cellular structures in the images made classification more difficult. As a result, the augmentation was adjusted and further data balancing was performed. Despite the good performance we were not able to validate our model using other external datasets. This is a limitation that should be addressed to improve our model further. Our model was also trained on 500 x 500 images which are patches of a larger slide. We were unable to get access to this large slide of images to train our model which resulted in greater accuracy however training on the later would improve the usability of our model in the real world.

6 Contributions

- **Student 1 Abiel J. Kim**: The distribution of my workload was primarily split into two aspects with respect to the software development life cycle: Team coordination and technical development of the deep neural network. In terms of team coordination, I took initiative of the opening phase of our project by setting up project resources and spearheading the brainstorming, planning, and analyses phases. Further, I remained active in the coordination of team synchronization meetings and ensuring everyone was aligned with respect to project goals and deadlines given our varying levels of ML experience. In terms of technical development, I actively participated in design discussions and implemented the changes that led to our highest model accuracy (more than 99%). I achieved this by diverging from the team's initial

grayscale conversion and instead preserving color information to leverage the data complexity space and the pretrained weights of our RESNET18. Another significant technical contribution I made was patching 2 bugs that had been overlooked in our implementation. Namely, the first bug was a data leak mistake that introduced influence from the test set to the train set and secondly a critical blunder that invoked the training data preprocessing pipeline for the test dataset as well.

- **Student 2 Jisoo Im**: My contributions focused primarily on the research, documentation, and presentation aspects of the project. I conducted a literature review of relevant academic papers to provide context and support for our approach. I was responsible for writing the Abstract, Introduction, Materials, Results, Conclusion and Discussions, Acknowledgments, and part of the Methodology sections in the final report. The Methodology section was originally drafted by Nardos, and I took over and expanded it into the model training, optimization techniques, and performance evaluation sections. In addition, I edited the PowerPoint presentation that Nardos initially created, adding a code review slide for the final demo video.
- **Student 3 Pabil Adhikari**: For this project I contributed in a variety of different ways. Firstly, I presented the ovarian cancer Kaggle dataset to my group members who all agreed to use it for our project. Following this, I built on Jordan's first implementation of the model and added a Kaggle API call which allowed the dataset to be directly downloaded within Google Colab, as well as modifying the data directory path to ensure the data was placed in the correct location. This was important as 3 of us did not have access to dedicated GPUs, so we had to use Colab to train the model instead. Additionally, I implemented the learning rate scheduler using CosineAnnealingLR and configured the learning rate to update after every epoch instead of every batch, which followed the suggested approach found in the PyTorch documentation. The choice to use CosineAnnealingLR was suggested to me by Jordan. Furthermore, I added code to track, visualize, and save changes in the model which recorded the loss, validation accuracy, and test accuracy after every epoch. These metrics were then plotted in a new notebook tab, which allowed for easier monitoring of the models' training progress and overall performance. Beyond coding contributions I also wrote the future works section of this report and along with Jordan, reworked the slide deck and recorded the final demo video.
- **Student 4 Nardos Solomon**: For this project my contributions were mainly centered around research, the report, and the presentation. I conducted research by looking at other academic reports that worked on a similar data set and project. I shared my findings with Jisoo to help in our approach in writing the report. I also created a rough plan for the methodology section of the report based on the code written by our team members and research of relevant content. I wrote the methodology section, with a part of it delegated to Jisoo. I was also responsible for writing the accomplishments in the final report. Furthermore, I was responsible for small changes throughout different sections with the help of Jisoo. Additionally, I created

the whole presentation for our video demo and Jisoo worked on code review slide. Furthermore, to aid those recording the video demo, I created a script for the video. The script and presentation were then modified while the video was being filmed.

– **Student 5 Jordan Clough**: My work in this project was heavily focused on the development of the model. I developed the baseline model on my own. This consisted of loading in the data set, creating the custom data loaders. As well as defining the model and creating the transforms that augment the image to provide a more robust model. I worked with Pabil to continue engineering the model and further increasing the models accuracy. I suggested to Pabil that we could aid in tuning the hyper parameters by implementing the Cosine Annealing learning rate scheduler. Abiel implemented changes by going away from converting to gray scale images and keeping the original 3 channel images. I then further tuned the model with changes to batch size and the transform. This finalized the changes to our model and I trained it to 50 epochs and stabilized the model at over 99% test accuracy. I then implemented a way to analyzed our completed model by developing the confusion matrix plot and output of total accuracy. I also worked with many group members on the demonstration and summary video of our model.

7 Conclusion and Discussions

In this project, we developed a convolutional neural network model using ResNet-18 with transfer learning in PyTorch to classify histopathological images of ovarian cancer into five distinct cell types. The model was trained on a curated data set after pre-processing and augmentation, and its performance was evaluated using metrics such as precision, precision, and confusion matrices. Our approach aimed to take advantage of the efficiency of a lightweight yet effective CNN architecture to capture complex features from high-resolution medical images.

While ResNet-18 provided a solid foundation with relatively fast training and good generalization, the project faced several limitations. One key challenge was class imbalance, which affected the model's ability to predict minority classes such as Mucinous Carcinoma (MC) accurately. Although transfer learning helped mitigate issues related to limited data, some classes still suffered from low recall, indicating that more domain-specific features may be necessary. In addition, medical images often contain subtle variations that are difficult to distinguish even for experts, making high-accuracy classification inherently difficult.

Despite these limitations, the project confirmed that the application of deep learning methods to medical image analysis is feasible and impressive. The performance of the model shows promise for further development, and the training process offered valuable insights into how CNNs respond to specialized data domains such as histopathology.

8 Future Work

Future improvements for this project could be made by replacing our current ResNet-18 model with a custom EfficientNet based CNN architecture. Implementing this new model would allow us to tailor it to our specific dataset which could optimize performance. The adjustments that could be made would be modifying the depth, width, and resolution scaling factor of EfficientNet to better suit our dataset. Further adjustments could be made by changing the way the data is augmented using augmentation policies like AutoAugment or RandAugment which can help prevent the model from overfitting during the data training phase. Additionally, experimenting with different optimizers could improve stability and convergence. While Adam is a widely adopted optimizer, the use of alternatives like AdamW (which improves weight decay handling), Lookahead (which reduces gradient variance leading to better training stabilization) could further improve stability and refine the overall model. Further adjustments could explore using different fine tuning strategies or adjusting the type of learning rate scheduler, which could improve the models overall performance.

Acknowledgements

We would like to thank all those who contributed to the success of this project. Special thanks to Professor. Ghassan Hamarneh and the teaching assistants for their guidance and feedback. We would like to acknowledge the use of Generative AI tools, specifically ChatGPT by OpenAI, which assisted us in this project. ChatGPT was used to support problem-solving, clarify complex concepts, and provide guidance on writing and formatting. All outputs from ChatGPT were critically reviewed and edited to align with the goals and academic integrity standards of this project. We gratefully acknowledge the data sources that made this project possible:

- The UBC-OCEAN Kaggle competition
- The Extracted Ovarian Cancer Cell Images dataset by HuyNguyen

These data sets provided high-quality histopathological images essential for training and evaluating our classification model.

We also express our appreciation for the educational resources that contributed to our understanding of neural networks. In particular, 3Blue1Brown YouTube playlist on Neural Networks offered clear and intuitive visual explanations of machine learning and deep learning, and the article *Introduction to Neural Networks (Part 1)* on Medium by Deep Learning Demystified provided foundational insights that supported our research and learning process.

References

1. R. M. Ghoniem et al. Multi-modal evolutionary deep learning model for ovarian cancer diagnosis. *Symmetry*, 13(4):643, April 2021.

2. S. K. Behera et al. Deep fine-knn classification of ovarian cancer subtypes using efficientnet-b0 extracted features: a comprehensive analysis. *Journal of Cancer Research and Clinical Oncology*, 150(7), July 2024.
3. J. Breen et al. A comprehensive evaluation of histopathology foundation models for ovarian cancer subtype classification. *npj Precision Oncology*, 9(1), January 2025.
4. K. He et al. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. Accessed: 2025-03-26.
5. O. Russakovsky et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec 2015.
6. Y.-L. Boureau et al. Learning mid-level features for recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2559–2566, 2010.
7. TensorFlow Authors. Data augmentation. `https://www.tensorflow.org/tutorials/images/data_augmentation`, 2023. Accessed: 2025-04-1.