

# Project 2

## FYS-STK4155

Krithika Gunasegaran, Oskar Idland, Erik Røset & Arangan Subramaniam  
*University of Oslo, Department of Physics*  
(Dated: November 21, 2024)

Add abstract here

<https://github.com/Oskar-Idland/FYS-STK4155-Projects>

### I. INTRODUCTION

### II. THEORY

### III. METHODS & IMPLEMENTATION

#### A. Data Preprocessing and Model Architecture

The dataset consisted of breast cancer histopathological images classified into three categories: normal, malignant and benign. Each malignant and benign image included an associated mask file denoting regions of interest, which were excluded from this analysis. All images were preprocessed to a standardized size of  $224 \times 224$  pixels to maintain consistency with standard deep learning architectures while preserving sufficient detail for medical diagnosis.

The dataset was partitioned into training (563 images), validation (100 images), and test (117 images) sets, representing approximately 72%, 12% and 15% of the data respectively. Stratification was maintained across splits to preserve class distribution, particularly important given the significant class imbalance (benign: 437, malignant: 210, normal: 133). Data augmentation techniques were employed to address this imbalance, specifically targeting the minority classes (malignant and normal). The augmentation pipeline included random horizontal flips with a high probability ( $p=0.9$ ) to maximize the diversity of cell orientations, constrained rotation ( $\pm 15$  degrees) to maintain anatomical plausibility, and color jittering to enhance robustness against staining variations common in histopathological samples.

Three distinct deep learning architectures were implemented and compared:

- A custom simple CNN was designed with specific consideration for the small dataset size. The architecture comprised two convolutional layers ( $3 \rightarrow 16 \rightarrow 32$  channels) with  $3 \times 3$  kernels, chosen to capture local cellular features while maintaining computational efficiency. ReLU activation was selected for its non-linearity and reduced likelihood of vanishing gradients. Max pooling operations follow each convolutional layer to achieve spatial dimensionality reduction while preserving important features. The resulting  $56 \times 56 \times 32$  feature maps

are flattened into a 100,352-dimensional vector, followed by fully connected layers reducing to 512 and 128 neurons respectively. This progressive reduction in dimensionality ( $100,352 \rightarrow 512 \rightarrow 128 \rightarrow 3$ ) creates an information bottleneck that forces the network to learn increasingly abstract representations of the input data, with the final layer outputting probabilities for our three classes.

- A MobileNet architecture[1], selected for its efficient depthwise separable convolutions which significantly reduce computational complexity while maintaining performance. This architecture has demonstrated success in medical image classification tasks where computational resources may be limited in clinical settings.[2]
- A pre-trained ResNet101 model[3], chosen for its deep residual learning framework that effectively addresses the degradation problem in deep networks and general good performance in medical classification tasks[4]. The model was pre-trained on ImageNet, allowing it to leverage general feature detection capabilities, while its final layer was modified to accommodate our three-class classification task.

#### B. Training and Evaluation

The models were trained using the Adam optimizer, selected for its adaptive learning rate capabilities and robust performance across different neural architectures. Learning rates were empirically determined: 0.001 for both Simple CNN and MobileNet, and a lower rate of 0.00005 for ResNet101 to prevent catastrophic forgetting of pre-trained features. A step learning rate scheduler (step size=7,  $\gamma=0.1$ ) was implemented to facilitate convergence to better optima. Training utilized cross-entropy loss, appropriate for our multi-class classification task, and incorporated early stopping (patience=2, minimum epochs=5) to prevent overfitting while ensuring sufficient learning time.

Models were evaluated using a comprehensive set of metrics: accuracy, precision, recall, and F1-score, with particular attention to per-class performance given the imbalanced nature of our dataset. Confusion matrices

were generated to provide detailed insight into class-wise performance and misclassification patterns.

### C. Implementation Details

The implementation used the PyTorch framework and was executed on GPU hardware. Training was conducted with a batch size of 8, chosen to balance between computational efficiency and stable gradient updates. Data normalization used ImageNet statistics (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]) to ensure com-

patibility with pre-trained models and standardize feature scales. All experiments were conducted with a fixed random seed for reproducibility, with final evaluation performed on the held-out test set to ensure unbiased performance assessment.

## IV. RESULTS & DISCUSSION

## V. CONCLUSION

## REFERENCES

- [1] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, *CoRR* **abs/1704.04861** (2017), 1704.04861.
- [2] P. Sharma, D. R. Nayak, B. K. Balabantaray, M. Tanveer, and R. Nayak, *Neural Networks* **169**, 637 (2024).
- [3] The PyTorch Foundation, “ResNet101,” <https://pytorch.org/vision/main/models/generated/torchvision.models.resnet101.html> (Accessed: November 2024).
- [4] Y. Wang, Z. Feng, L. Song, X. Liu, and S. Liu, *Computational and Mathematical Methods in Medicine* **2021**, 2485934 (2021), <https://onlinelibrary.wiley.com/doi/pdf/10.1155/2021/2485934>.

## Appendix A: Code

Link to our GitHub repository: <https://github.com/Oskar-Idland/FYS-STK4155-Projects>