Group I Project Final Report

Abstract

This project encompasses two main tasks with the goal of exploring the performance of CNN encoders in feature extraction and the potential of transfer learning across diverse datasets. Task 1 focuses on training and evaluating a CNN encoder on colorectal cancer images. The results are visualized using t-SNE. Task 2 extends the study to a prostate cancer dataset and an animal faces dataset by utilizing both the Task 1 CNN encoder and a pre-trained ImageNet encoder. In terms of feature extraction, the ImageNet encoder showed good performance on both datasets, while the Task 1 model performed well only on the cancer dataset. The extracted features by the ImageNet model are employed in training supervised machine learning models such as KNN and RF, and both models showed high test accuracy. The findings highlight the effectiveness of transfer learning in feature extraction and showcase the versatility of successfully extracted features in classical machine learning models.

1. Introduction

Convolutional Neural Networks (CNNs) are a type of Machine Learning (ML) model commonly used in various forms of classification. In the realm of computer vision (CV), CNNs are pivotal in image classification, by learning and extracting features from a sample set of images to generalize data and properly classify similar images outside the sample dataset.

Training a CNN encoder comes with a set of challenges. Training a CNN with millions of parameters is computationally demanding and time consuming. This results in a protracted iteration process making it time-intensive to implement changes to the network. To tackle this issue, more powerful hardware can be used to train the model. Another challenge is the lack of the data. If there are not enough images (since most research data are not widely available to the public [1]), or if the images are too similar, the CNN could have poor generalization on previously unseen data. Data augmentation can help resolve this by tuning image attributes, such as rotating or adjusting the brightness to vary the training set. Transfer learning is another solution to mitigate this problem. It involves using a pre-trained model that was trained with a large amount of data for a similar task with a smaller dataset.

The goal of this project is divided into two main tasks. The first task (Task 1) involves training and evaluating a CNN encoder on images of human tissue (Dataset 1) to classify colorectal cancer and then visualizing the results using t-SNE (t-distributed Stochastic Neighbor Embedding). The second task (Task 2) involves using the trained CNN encoder from the first task and a pre-trained CNN encoder from ImageNet to extract features from two datasets, a prostate cancer dataset (Dataset 2) and animal faces dataset (Dataset 3), then train two supervised machine learning models. The results are utilized to evaluate ML approaches and compare CNN encoder performance on multidisciplinary datasets.

By conducting a series of experiments, we aim to shed light on how two different models' performances in feature extraction on unseen data may differ in the context of transfer learning and how the successfully extracted features can be utilized in other machine learning techniques. Through partial completion of Task 1, our objective was to train the CNN encoder to achieve an 80%-90% test accuracy, as high generalization ability would facilitate successful feature extraction on unseen data. The Task 1 model demonstrated high performance on the test dataset, and the feature extraction was successfully conducted, as evidenced by clearly clustered images into three distinct classes in t-SNE visualization.

Nevertheless, when comparing feature extraction between the Task 1 and ImageNet CNN encoders, it was observed that the Task 1 model successfully extracted features on Dataset 2, but not on Dataset 3. In contrast, the ImageNet encoder performed well on both Dataset 2 and Dataset 3, benefiting from its training on a large annotated dataset.

The extracted features of those two datasets by ImageNet were utilized to perform a K-Nearest Neighbors (KNN) model and a Random Forest (RF) model. For both cases, training and testing accuracies turned out to be high. Therefore, the result we found corroborates the effectiveness of transfer learning on feature extraction, and how the extracted features could be reused in different ML techniques, even the classical ones that cannot handle the datasets with high complexity.

Many researchers have used CNNs for classification Tasks in medical imaging and computer vision. CNNs were shown to accurately diagnose lung and colorectal cancer subtypes in one machine learning study [2]. They also outperformed human experts in image classification tasks, and another study examined deep learning model reengineering, including data preprocessing and hyperparameter optimization [3]. Furthermore, CNNs were used to categorize similar animal species, showing their potential for rare animal

conservation [4]. A novel study used deep learning and explainable AI to classify prostate cancer from ultrasound and MRI images, improving their interpretability [5]. Jiang and their colleagues' [6] experiments demonstrate CNNs' versatility and challenges across datasets and objectives. They emphasize dataset-specific model training, targeted feature extraction, and model interpretability. This literature is highly relevant to the current project, which trains a CNN for colorectal cancer classification and compares CNN encoder performance on human tissue and animal faces, focusing on precision, recall, F1-score, support, and accuracy metrics.

2. Methodology

2.1. Dataset

Three datasets were used in the implementation of the project. Dataset 1 includes microscopic images of stained human colon tissue taken from histology slides, collected from the NCH Biobank and UMM pathology archive [7]. Dataset 2 contains microscopic images of human prostate tissue taken from histology slides, collected from The Cancer Genome Atlas [8]. Dataset 3 consists of faces of animals with their eyes centered, collected from image websites Flickr and Pixbay [9].

Dataset	1	2	3
Image Size	224x224	300x300	512x512
Original Images	100,000	120,000	16,130
Original Classes	9	3	3
Reduced Images	6000	6000	6000
Reduced Classes	3	3	3

Figure 1. Dataset statistics

For this project, only three classes out of the original nine remained for Dataset 1: Smooth muscle (MUS), normal colon mucosa (NORM), and cancer-associated stroma (STR). The target classes for Dataset 2 and Dataset 3 remained unchanged. Dataset 2 included classes such as Prostate Cancer Tumor Tissue, Benign Glandular Prostate Tissue, and Benign Non-Glandular Prostate Tissue, while Dataset 3 consisted of classes such as cats, dogs, and wildlife animals. Dataset statistics are summarized in Figure 1.

The image datasets were preprocessed beforehand in order to facilitate the learning and improve the generalization. Each image was resized to 224x224x3 to meet the input requirements for ResNet18 and normalized using the ImageNet means and standard deviations, [0.485, 0.456, 0.406] and [0.229, 0.224, 0.225] respectively. The difference between the images only resized and the images resized and normalized can be observed by comparing Figure 2 and Figure 3.

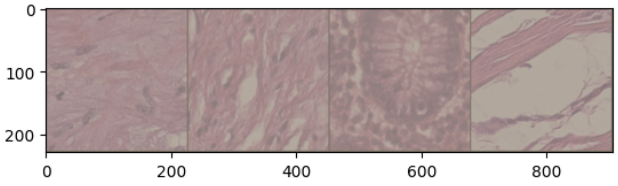

Figure 2. Original images

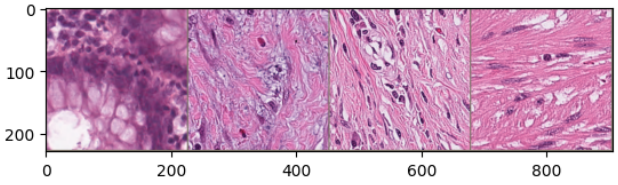

Figure 3. Normalized images

Throughout this project, for both Task 1 and Task 2, the datasets were split with an 80:20 ratio. This ratio was selected to ensure having enough data for training and a sufficient amount for testing, enabling robust evaluation.

2.2. CNN-based classification

For the task of Colorectal Cancer classification, we selected ResNet-18, which consists of five residual blocks. The first block includes a 7x7, 64-convolution layer, and a 3x3 max-pooling layer. The subsequent four blocks contain two 3x3 convolution layers each, with filter numbers of [64, 128, 256, 512], using ReLU activation functions. Each block incorporates a skip connection from the input to the output. ResNet has demonstrated exceptional performance in image classification and various computer vision Tasks, attributed to the introduction of residual learning to mitigate the gradient vanishing problem. Among the ResNet models, ResNet-18 was specifically chosen for its lightweight nature.

In the realm of ImageNet classification, the training and test losses of the layer-18 plain network were notably higher compared to the outcomes achieved with the ResNet-18 model [10]. When comparing ResNet-18 and ResNet-34 models, the 34-layer ResNet exhibited relatively lower training errors and superior generalization. Nevertheless, the discrepancy was approximately 2-3%, and the 18-layer ResNet also demonstrated commendable performance [10]. Ultimately, considering the computational resource constraints when using Google Colab, we opted for the lighter ResNet-18 model.

The training times exhibit a decreasing trend with each epoch, starting from 28.02 seconds in the initial epoch and

stabilizing at around 20.49 seconds by epoch 21. Validation times remain consistent at 1.27 to 1.56 seconds per epoch. The total training time for 21 epochs is approximately 469.68 seconds. The measured FLOPS for our ResNet-18 model are 1.17×10^{11}, highlighting the computational complexity. The input passes through Convolutional Layers and Residual Blocks, yielding a representation of size 6000 x 512. Instead of employing Fully Connected Layers for subsequent processing, t-SNE is applied to the extracted 512 output features, enabling the visualization of class labels.

2.3. Feature extraction

Feature extraction in Task 2 is carried out using pre-trained ResNet18 models. One of the models is the one we trained in Task 1 using Dataset 1 and fine-tuned to this specific dataset (we refer to this as the Task 1 model). The other model is a pre-trained model using ImageNet weights that has never been trained on anything else and is known for its immense diversity and volume, allowing it to extract generalized features (we refer to this as the ImageNet model).

These pre-trained models operate as fixed feature extractors when applied to new datasets. The classification layer is removed from the models, and they are just used to extract features. Every image from the new datasets is then sent through the network until it reaches the flattening layer. By doing so, when we feed a single image to the classifier, it generates 512 features that represent the image. As a consequence, for a given dataset, each data point is transformed to a feature vector, yielding a new dataset of 6000 data points with 512 features/columns per data point. These high-dimensional features are then projected into two dimensions for display using t-SNE. Figures 8–11 show the t-SNE visualization for every scenario.

On Dataset 2 features extracted by the ImageNet model, we utilize KNN and RF ML algorithms for classification. The simple, instance-based learning approach KNN classifies a new sample using a similarity measure (typically distance functions). It classifies the new data point like the majority of its neighbors based on the 'K' nearest labeled training data points. RF, on the other hand, uses ensemble learning to train several decision trees and output the mode of their classes. RF is noted for its resilience and low training data overfitting. Both classification systems have benefits. While straightforward to learn and rapid to categorize, KNN can be slow when examining large datasets. RF, which requires more processing power and memory, often provides better accuracy because of its diversity of judgment perspectives and is best for datasets with many characteristics, like our high-dimensional extracted features.

3. Results

3.1. Experiment setup

For Task 1, where we trained the CNN encoder from scratch, we selected the categorical cross-entropy (CCE) loss function and Adam optimizer as the objective function and optimizer, respectively. To control the learning algorithm and achieve the best result, hyperparameter search was conducted using the Orion framework. A learning rate of 0.004083 was utilized, and training was conducted for 21 epochs with a batch size of 64, as determined by random hyperparameter search. The learning rate was sampled from a log-uniform distribution between 1e-4 and 0.1. Batch size options were limited to 16, 32, and 64, while the number of epochs ranged between 15 and 30. The maximum experiment trial number was set to 10. All these ranges were constrained as described due to computational complexity.

3.2. Main results

Following the hyperparameter search, training of the ResNet18 model was implemented. Validation was performed at the end of each epoch, and both training loss/accuracy and validation loss/accuracy were measured and compared, as shown in Figure 4.

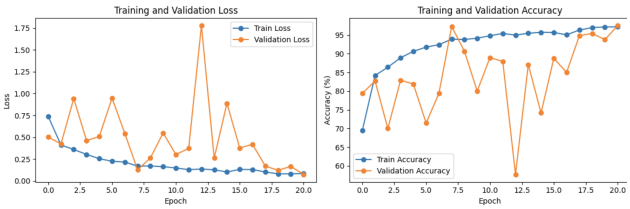

Figure 4. Training, validation loss and accuracy

Despite some fluctuations during training, the validation loss demonstrated convergence, and the accuracy reached 97.5%. Using Scikit-learn's classification report, we calculated precision, recall, and F1-score, as depicted in Figure 5. The testing accuracy, F1-score of the macro-average, and that of the weighted average all turned out to be 98.0%, consistent with the validation result.

	precision	recall	f1-score	support
MUS	0.96	0.99	0.98	398
NORM	0.98	1.00	0.99	395
STR	0.99	0.94	0.96	359
accuracy			0.98	1152
macro avg	0.98	0.98	0.98	1152
weighted avg	0.98	0.98	0.98	1152

Test Loss: 0.0838, Test Accuracy: 97.67%

Figure 5. Classification report

After training the CNN encoder, we performed dimensionality reduction using t-SNE. The image datasets were fed to the model whose classification layer was removed. The extracted features of each data point were then stacked, resulting in a shape of (4800, 512) in the end. Without feature extraction, when the image is directly fed to the model, each data cluster is not distinguishable from each other and hard to interpret, as shown in Figure 6.

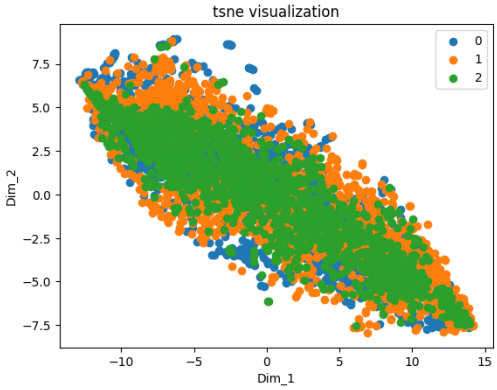

Figure 6. t-SNE visualization without feature extraction on Dataset 1

This result reflects the nature of the datasets, wherein datasets with different labels appear extremely similar on the surface level. However, with proper feature extraction, three different classes are clearly divided, as illustrated in Figure 7. While some data points may slightly deviate, most of class 2 (cancer images) tend to cluster in the area of negative values in both Dimension 1 and Dimension 2, unlike class 0 and class 1. This result indicates that the model successfully extracted features and separated higher-dimensional data, allowing us to observe the features of cancer images that differentiate them from the other two non-cancer image classes. Therefore, Task 1 concluded with obtaining a CNN encoder demonstrating good generalization power and confirming successful feature extraction through t-SNE visualization.

By examining the t-SNE visualizations of Dataset 2 and Dataset 3, which were processed by the ImageNet and Task 1 models, we can draw many inferences regarding their feature extraction capabilities. For the feature of Dataset 2 extracted by the ImageNet model as visualized in Figure 8, the clusters are somewhat intertwined. Classes 0 and 1 are well separated from class 2, but are intermingled among themselves. This demonstrates a reasonable level of class differentiation, albeit not as sharp as one might expect for clear-cut classification tasks. When we shift our attention to the t-SNE visualization of dataset 2 as processed by the task 1 model, the clusters exhibit the same behaviors, with classes 0 and 1 substantially interspersed but well separated from

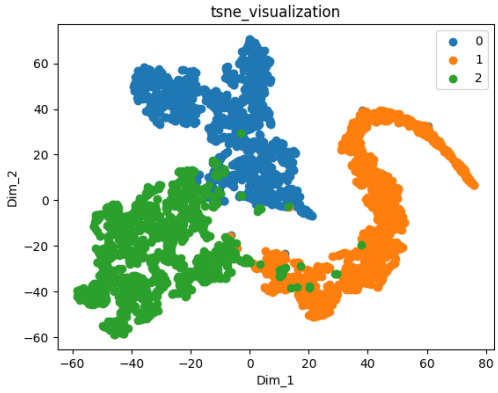

Figure 7. t-SNE visualization with feature extraction on Dataset 1

class 2. This shows that the models are quite effective at differentiating between class 2 and the first two classes. This is reasonable given that class 2 in dataset 2 contains malignant tissues that differ significantly from the other classes.

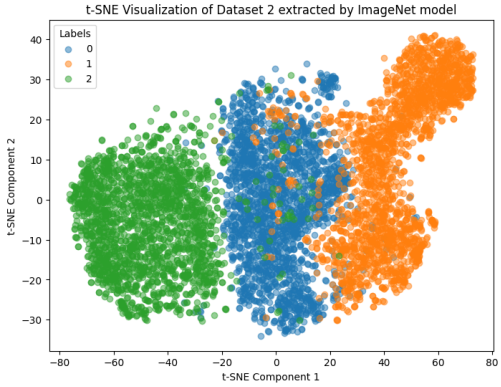

Figure 8. t-SNE visualization of Dataset 2 extracted by ImageNet model

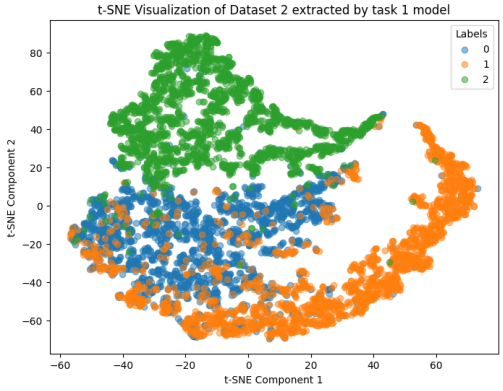

Figure 9. t-SNE visualization of Dataset 2 extracted by Task 1 model

With Dataset 3, the ImageNet model performs exceptionally well in differentiating the 3 classes. Figure 10 shows the visualization of Datset 3 extracted by ImageNet model. Each class forms well-defined clusters, with minimal overlap between them. This indicates that the ImageNet model is quite effective at separating the features of the different classes in this dataset. Compared to class 2, where the cluster dispersed into smaller clusters, each cluster of classes 0 and 1 is more compacted. This is because, in contrast to classes 0 and 1, which represent cats and dogs, respectively, class 2 include several species of wild animals containing different features for each species.

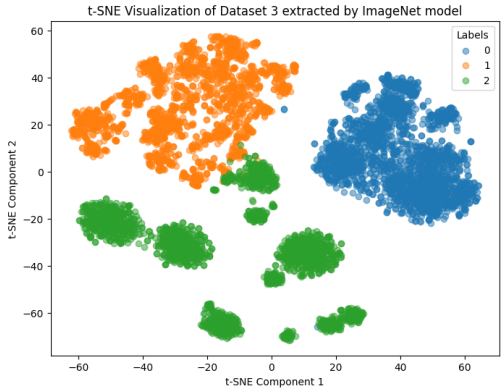

Figure 10. t-SNE visualization of Dataset 3 extracted by ImageNet model

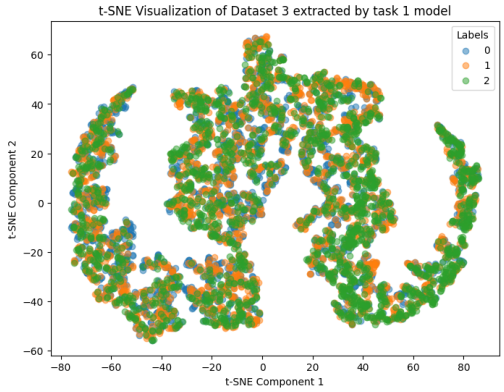

Figure 11. t-SNE visualization of Dataset 3 extracted by Task 1 model

In contrast, Task 1 model's handling of Dataset 3 is markedly different. The clusters here are not as well-defined, with a notable intermingling of data points from the different classes, as illustrated in Figure 11. The overlap suggests that the features extracted for these classes by Task 1 model are not as discriminative. The reason for this decline in performance when compared to Dataset 2 is because the Task 1 model was trained only on Dataset 1, which has more properties in common with Dataset 2. When presented with the task of discriminating between new classes other than medical images, Task 1 model is therefore less reliable. On the other hand, ResNet-18 model, when pretrained with ImageNet weights, was trained on a dataset comprising 1,000 different classes. ImageNet is a large-scale dataset that includes a wide variety of images across these classes, making the model well-suited for diverse image classification Tasks, which explains for good results on both Dataset 2 and Dataset 3.

Based on the t-SNE visualization results from the first half of Task 2, the scenario of Dataset 2 extracted by ImageNet model was selected to complete the rest of the task. Two classical ML methods, KNN and RF, were implemented using the feature of Dataset 2 extracted by ImageNet model, in order to assess the performance and usefulness of the feature extraction in relation to the ML methods.

KNN classification report:

	precision	recall	f1-score	support
0	0.91	0.99	0.95	385
1	0.99	0.94	0.97	393
2	1.00	0.96	0.98	422
accuracy			0.96	1200
macro avg	0.97	0.97	0.96	1200
weighted avg	0.97	0.96	0.97	1200

Figure 12. KNN Classification report

Before KNN classification was performed, grid search was conducted to find the best hyperparameter. With the parameters of, k from 1 to 40, leaf size from 20 to 40 and distance metrics of cosine and Minkowski with p being 1 or 2 for Manhattan and Euclidean distance. With 5-fold cross validation, the optimal parameters were determined to be k of 5, cosine distance, leaf size of 20, and p of 1.

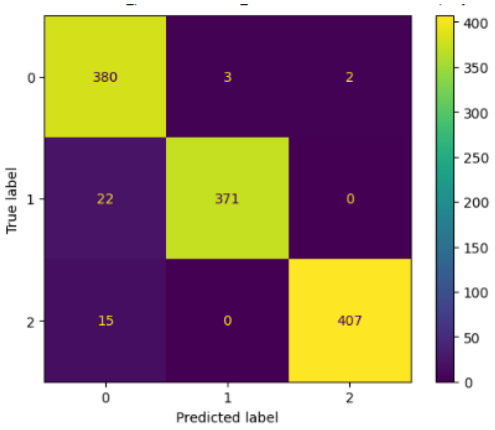

Figure 13. KNN Confusion Matrix

Using the parameters determined by the grid search, the model was assessed on precision, recall, F1-score, support

and accuracy, as shown in Figure 12. Confusion matrix was also used to analyze the model's performance further, as depicted in Figure 13. The results are consistent with the t-SNE extraction on Dataset 2 from Figure 8, with an accuracy of 96% showing that the features classes cluster together well. The confusion matrix is also consistent with the t-SNE where classes 1 and 2 have no overlap, and most errors are the result of misclassifying class 1 or 2 as class 0.

For RF technique, grid search was also executed to search best hyperparameters. The best parameters result was reported to be gini index criterion, no max depth, and minimal sample split of 4. The best estimator returned by the grid search showed the test accuracy of 0.96. The same tendency as KNN was observed in this model's performance that most of the error comes from the misclassification of class 1 or 2 as class 0.

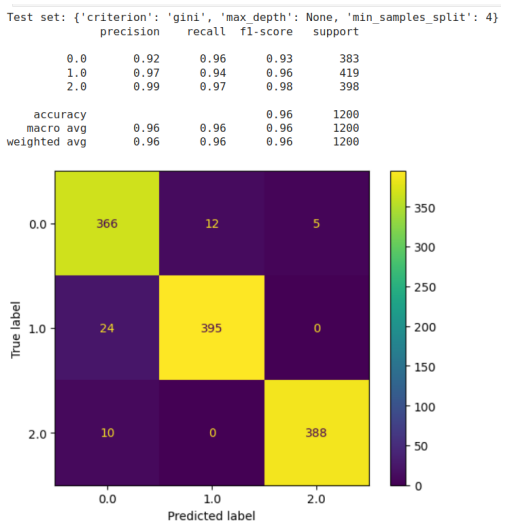

Figure 14. RF classification report and confusion matrix with Gridsearch

Considering that RF utilizes multiple decision trees to generate the likelihood of being in each class at any point, we can see that the dataset is separated in a very beneficial manner, with all classes having mostly distinct boundaries between each other and overlap between the three is minimal and leads to notable results.

These results from Task 2 align with our expectations at the beginning phase of this project, but the implications vary. We anticipated that the model from Task 1 would perform better feature extraction on Dataset 2 than Dataset 3, as efficiently as the ImageNet model, given that Dataset 2 comes from the same domain of medical imaging. The implication we had in mind was more about how a model trained with a smaller dataset could effectively handle other datasets within the same domain, making the use of the ImageNet model less necessary and alleviating the pressure of training a model with a massive image dataset.

However, the opposite conclusion was derived from our actual experiment: a substantial pre-trained model like ImageNet could efficiently extract features from smaller datasets, and the resulting values could be reused to successfully train less powerful ML models. This finding also alleviates the pressure associated with building a highly complex model using a large dataset. Therefore, throughout this project and various experiments, we have gained insights not only about the basic concepts and implementation of ML but also about one of the ways to mitigate the challenges that CNN faces.

References

[1] M. S. Hosseini, B. E. Bejnordi, V. Q.-H. Trinh, D. Hasan, X. Li, T. Kim, H. Zhang, T. Wu, K. Chinniah, S. Maghsoudlou, *et al.*, "Computational pathology: A survey review and the way forward," *arXiv preprint arXiv:2304.05482*, 2023. 1

[2] A. Hage Chehade *et al.*, "Lung and colon cancer classification using medical imaging: a feature engineering approach," *Physical and engineering sciences in medicine*, vol. 45, no. 3, pp. 729–746, 2022. 1

[3] D. Sarvamangala and R. Kulkarni, "Convolutional neural networks in medical image understanding: a survey," *Evol. Intel.*, vol. 15, pp. 1–22, 2022. 1

[4] E. Xi, "Image classification and recognition based on deep learning and random forest algorithm," 06 2022. 2

[5] M. R. Hassan, M. F. Islam, M. Z. Uddin, G. Ghoshal, M. M. Hassan, S. Huda, and G. Fortino, "Prostate cancer classification from ultrasound and mri images using deep learning based explainable artificial intelligence," *Future Generation Computer Systems*, vol. 127, pp. 462–472, 2022. 2

[6] W. Jiang, V. Banna, N. Vivek, A. Goel, N. Synovic, G. K. Thiruvathukal, and J. C. Davis, "Challenges and practices of deep learning model reengineering: A case study on computer vision," 2023. 2

[7] J. N. Kather, N. Halama, and A. Marx, "100,000 histological images of human colorectal cancer and healthy tissue," Apr. 2018. 2

[8] Y. Tolkach, "Datasets digital pathology and artifacts, part 1," May 2021. 2

[9] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[10] S. R. Kaiming He, Xiangyu Zhang and J. Sun, "Deep residual learning for image recognition," 2015. 2

Gantt Chart The tentative schedule for this project is outlined in Figure 15. Until the end of October, the proposal will be revised based on feedback and then finalized. The group aims to complete Task 1 by the second week of November and finish writing the progress report. The rest of the schedule will be dedicated to Task 2 and preparing the final presentation and report.

	Task Name	Duration	Start	ETA
1	Complete project execution (Due date: 2023-12-05)	49 days	2023-10-16	2023-12-04
2	Proposal revision (Due date: 2023-10-22)	6 days	2023-10-16	2023-10-21
3	Waiting for the feedback and finalizing proposal	7 days	2023-10-23	2023-10-29
4	Task 1: Model Training	12 days	2023-10-30	2023-11-10
5	Data Processing	3 days	2023-10-30	2023-11-01
6	Model architecture selection	3 days	2023-11-02	2023-11-04
7	Model Training	3 days	2023-11-05	2023-11-07
8	Performance Evaluation	3 days	2023-11-08	2023-11-10
9	Progress Report Writing (Due date: 2023-11-12)	4 days	2023-11-08	2023-11-11
10	Task 2: Feature Extraction & ML Classification	14 days	2023-11-13	2023-11-26
11	Feature Extraction with Encoder from Task 1	3 days	2023-11-13	2023-11-15
12	Feature Extraction with Pre-trained ImageNet model	3 days	2023-11-16	2023-11-18
13	Dimensionality Reduction	3 days	2023-11-19	2023-11-21
14	ML Classification	3 days	2023-11-22	2023-11-24
15	Performance Evaluation	2 days	2023-11-25	2023-11-26
16	Final Presentation Preparation	8 days	2023-11-27	2023-12-04
17	Final Report	4 days	2023-11-27	2023-11-30
18	Final Presentation	4 days	2023-12-01	2023-12-04
*	Create private github repository and commit	49 days	2023-10-16	2023-12-04

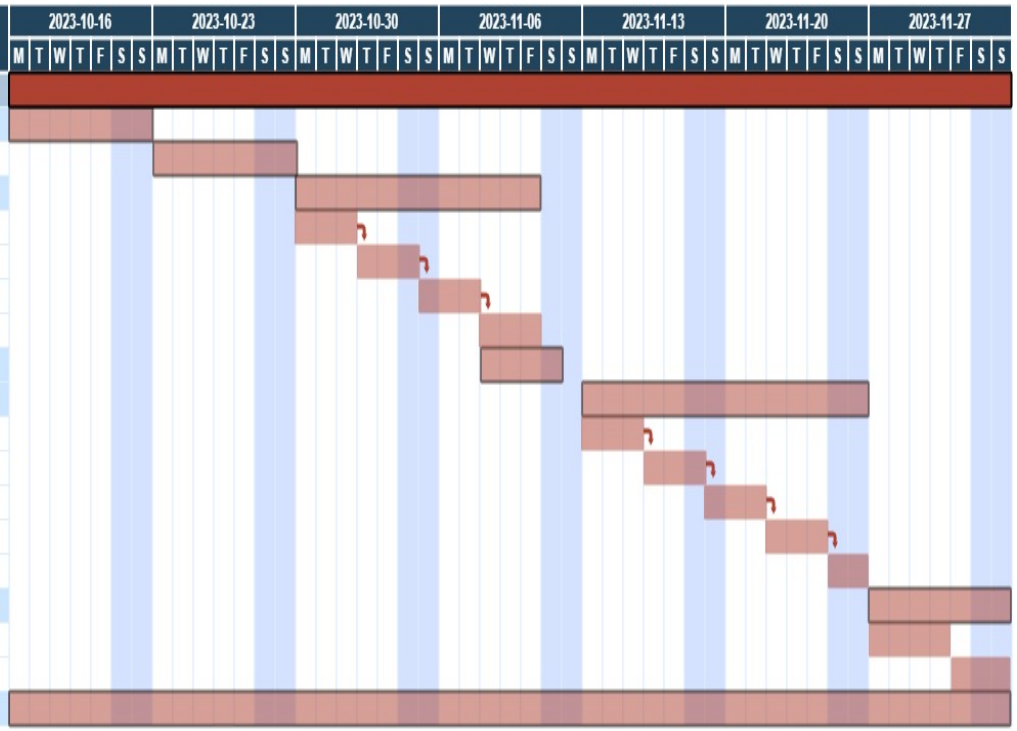

Figure 15. Project's Gantt Chart