



MINOR PROJECT REPORT

ON

AUTOENCODER BASED DISEASE DIAGNOSIS SYSTEM

AT

G D GOENKA UNIVERSITY

Submitted partial in fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

Submitted By:

Name: Harsh Dahiya
(200020223034), Manas Nand
Mohan (200020223028), Kishori
Shyam (200020203051), Lionel
Mwabile (200020223043)

Programme: B.Tech. (CSE & AI)

Under the guidance Of:

Name: Ms. Manka Sharma

Designation: Assistant Professor

Department: Computer Science and
Engineering

Manka Valsu
20/11/23

**Department of Computer Science and Engineering, School
of Engineering and Sciences, GD Goenka University**

December 2023

DECLARATION

We, the undersigned, solemnly declare that the project report- “**AUTOENCODER BASED DISEASE DIAGNOSIS SYSTEM**” is based on our own work carried out during the course of our study under the supervision of Ms. Manka Sharma. We further certify that:

- I. The work contained in the report is original and has been done by us.
- II. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.
- III. We have followed the guidelines provided by the university in writing the report.

Name: **Harsh Dahiya**
Enrollment No: **200020223034**
Program: **B.Tech CSE (AI&ML)**

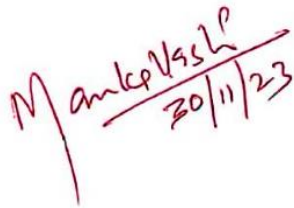
Name: **Manas Nand Mohan**
Enrollment No: **200020223028**
Program: **B.Tech CSE (AI&ML)**

Name: **Kishori Shyam**
Enrollment No: **200020203051**
Program: **B.Tech CSE**

Name: **Lionel Mwabile**
Enrollment No: **200020223043**
Program: **B.Tech CSE (AI&ML)**

CERTIFICATE OF PROJECT COMPLETION

This is to certify that Harsh Dahiya, Manas Nand Mohan, Kishori Shyam, Lionel Mwabile from **(B.Tech CSE & AI)** of **GD Goenka University**, Gurgaon have successfully completed the Minor Project from **July 2023** to **December 2023**. During this project, they have worked on **AUTOENCODER BASED DISEASE DIAGNOSIS SYSTEM** in Computer Science under the guidance of **Ms. Manka Sharma**. Their overall performance during the project duration was enthusiastic and admirable.



Name: **Ms. Manka Sharma**

Designation: **Assistant Professor (SoES)**

Department: **Computer Science and Engineering**

ACKNOWLEDGMENT

This semester I had the opportunity to work on my major project which was indeed a great chance for learning. This has helped me to build my competence, confidence, and credibility through practical skills that help us innovate and build our future. Therefore, I consider myself a very lucky individual as I was provided with this wonderful opportunity.

Bearing this in mind, I am using this opportunity to express my deepest gratitude and special thanks to **Ms. Manka Sharma** who in spite of being extraordinarily busy with her duties, took time out to hear, guide, and keep me on the correct path and allow me to carry out my project at their esteemed organization.

I truly perceive this opportunity as a big milestone in my career development. I will strive to use the procured skills and knowledge in the best possible way and will continue to work on their improvement, in order to attain desired career objectives.

ABSTRACT

Breast cancer remains a critical health concern globally, emphasizing the need for advanced diagnostic tools. This project introduces an innovative Breast Cancer Detection System that integrates machine learning, web development, and Robotic Process Automation (RPA). The main objective is to create a comprehensive system for the early and accurate identification of Invasive Ductal Carcinoma (IDC) in histology images, ultimately leading to timely diagnoses and improved patient outcomes.

The project employs three machine learning approaches: Convolutional Neural Networks (CNNs), transfer learning with EfficientNetV2S, and an autoencoder-based method. Data preprocessing involves a diverse dataset of 162 breast cancer slide images, emphasizing the importance of preparing data for effective model training. Notably, the system features a user-friendly web interface developed using HTML, CSS, and JavaScript, providing seamless interactions for end-users.

Robotic Process Automation (RPA) is seamlessly integrated using Automation Anywhere, automating tasks such as opening Visual Studio Code, starting the Flask server, interacting with the web browser, and capturing and interpreting output. This RPA integration enhances the system's efficiency and usability.

The methodologies showcase the strengths of each machine learning model, with transfer learning using EfficientNetV2S demonstrating superior accuracy in breast cancer detection. The project's conclusion emphasizes the success of the integrated approach, highlighting the pivotal role of RPA in automating essential tasks and the user-friendly web interface for a holistic solution.

Ethical considerations related to responsible machine learning usage in medical contexts are discussed, acknowledging the importance of patient privacy and data security. Additionally, potential deployment constraints, including hardware requirements and compatibility issues, are addressed. The abstract encapsulates the multidimensional nature of the project, combining machine learning, web development, and RPA for an advanced Breast Cancer Detection System.

Table of Contents

I. Introduction	1-6
A. Problem statement.....	2-3
B. Background.....	3-4
C. Objectives.....	4-5
D. Scope.....	5-6
II. Execution summary.....	7-8
III. Technical requirements.....	9-12
IV. About Dataset.....	13-15
V. Tools and Methodologies	16-22
VI. Project Flow.....	23-27
VII. Observation and Results.....	28-35
VIII. Project Conclusion and future directions.....	36-41
IX. Ethical considerations.....	42-43
X. References.....	44-46

Table of Figures

Figure 1: Invasive Ductal Carcinoma vs Invasive Lobular Carcinoma	1
Figure 2: Different types of breast cancers.....	3
Figure 3: IDC+ Areas	4
Figure 4: H/W Requirements.....	9
Figure 5: Software Requirements.....	10
Figure 6: Insights from the Dataset.....	14
Figure 7: Convolutional Neural Network Architecture	17
Figure 8: EfficientNet-V2S Architecture	18
Figure 9: Autoencoder Architecture	18
Figure 10: Default Learning Rate for Adam Optimizer	19
Figure 11: Web Dev Processes Flow.....	21
Figure 12: Significance of RPA.....	22
Figure 13: Flowchart of the complete project	23
Figure 14: Observations of accuracy and loss over different epochs for model 1	30
Figure 15: Confusion matrices for model 1	31
Figure 16: Observations of accuracy and loss over different epochs for model 2	32
Figure 17: Confusion matrices for model 2	33
Figure 18: Confusion matrix of EfficientNet-V2S ased model	37
Figure 19: Observation of EfficientNet-V2S based model over different epochs	38

I. Introduction

Breast cancer, a pervasive and potentially life-threatening disease, is the most commonly diagnosed cancer among women globally. Its prevalence underscores the urgency to advance diagnostic methodologies, aiming not only for early detection but also for refined precision in characterizing subtypes. Early and accurate diagnosis is not merely a medical imperative; it is a beacon of hope, guiding treatment planning and significantly influencing patient outcomes.

In the landscape of breast cancer subtypes, Invasive Ductal Carcinoma (IDC) emerges as a particularly noteworthy entity. Accounting for a substantial proportion of breast cancer cases, IDC is characterized by the infiltration of cancer cells into surrounding healthy breast tissue. The aggressive nature of IDC necessitates not only swift detection but also precise identification and classification for optimal therapeutic interventions. Understanding the intricacies of IDC is imperative for tailoring treatment strategies to the unique characteristics of this subtype.

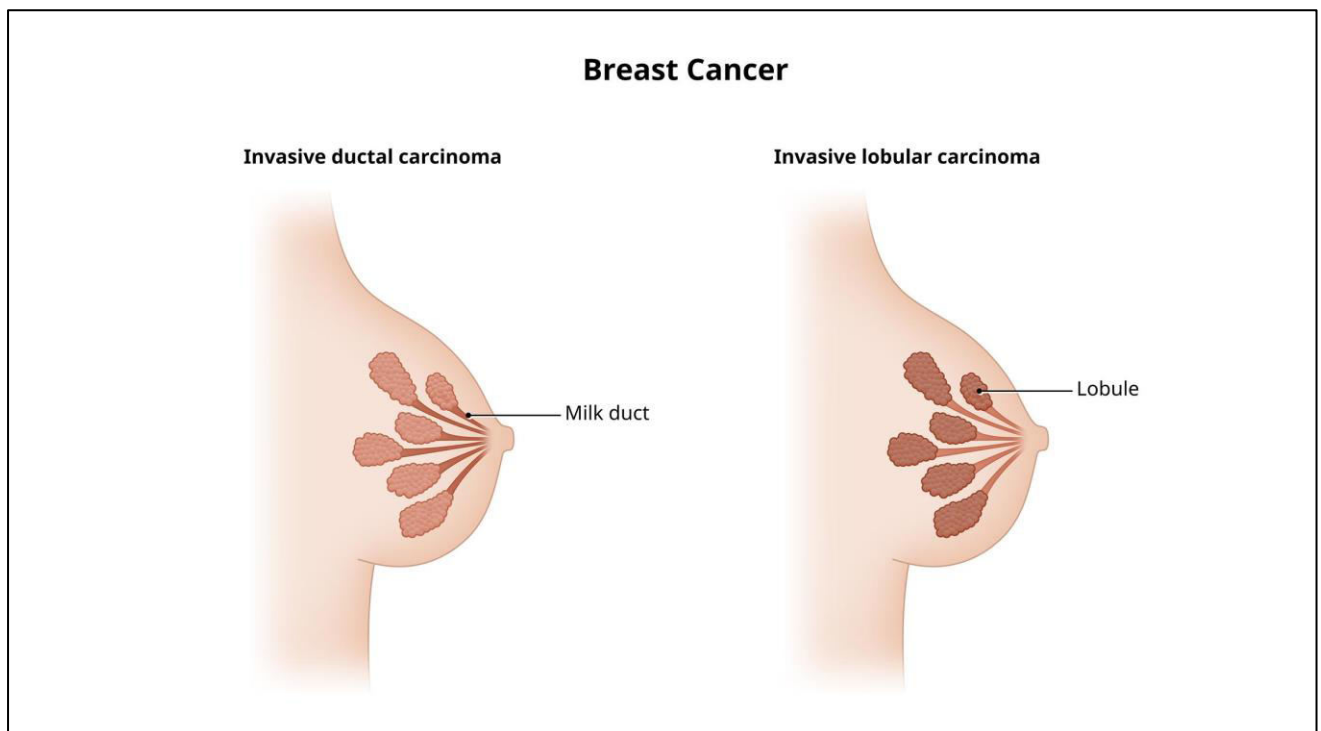


Figure 1: Invasive Ductal Carcinoma vs Invasive Lobular Carcinoma

As we delve into the complexities of breast cancer diagnosis, the traditional methods, while valuable, face inherent challenges. The reliance on manual examination of histopathological slides introduces subjectivity and can be time-consuming. Pathologists, dedicated to their craft, navigate through vast amounts of visual data, seeking patterns and abnormalities. However, the growing volume of diagnostic cases places immense pressure on healthcare systems, urging the integration of innovative technologies to augment human expertise.

This project positions itself at the intersection of medical imaging and machine learning, aiming to bring about transformative changes in breast cancer diagnostics. By leveraging advanced image processing techniques and state-of-the-art machine learning algorithms, we embark on a journey to automate the identification and classification of IDC. The focus on whole mount slide images, with their rich spatial information, promises a holistic understanding of breast tissue pathology.

From a dataset comprising 162 meticulously scanned whole mount slide images at a resolution of 40x, we extract 277,524 patches. These patches, classified into IDC negative and positive instances, form the basis of our exploration into machine learning methodologies. Our goal is not only to develop models that can accurately discern between IDC subtypes but also to contribute to a paradigm shift in breast cancer diagnostics, where technology becomes a synergistic partner with human expertise.

In the subsequent sections, we navigate the tools and technologies that empower our exploration, unravel the methodologies employed in the project flow, contemplate our findings, and cast our vision towards future directions in this critical domain of medical research.

A. Problem Statement:

The conventional method of breast cancer diagnosis involves manual examination of tissue samples by skilled pathologists. While this approach has been the cornerstone of cancer diagnosis for decades, it is not without limitations. The subjective nature of visual inspection, coupled with the

increasing volume of diagnostic cases, presents challenges in terms of accuracy, speed, and scalability. Specifically, the identification and classification of IDC regions within breast tissue samples demand a more objective and automated approach. The problem at hand is twofold: the need for precise identification of IDC regions in whole mount slide images, and the development of machine learning models capable of automating the classification process. Addressing these challenges can significantly enhance the efficiency of breast cancer diagnosis, enabling earlier interventions and personalized treatment plans.

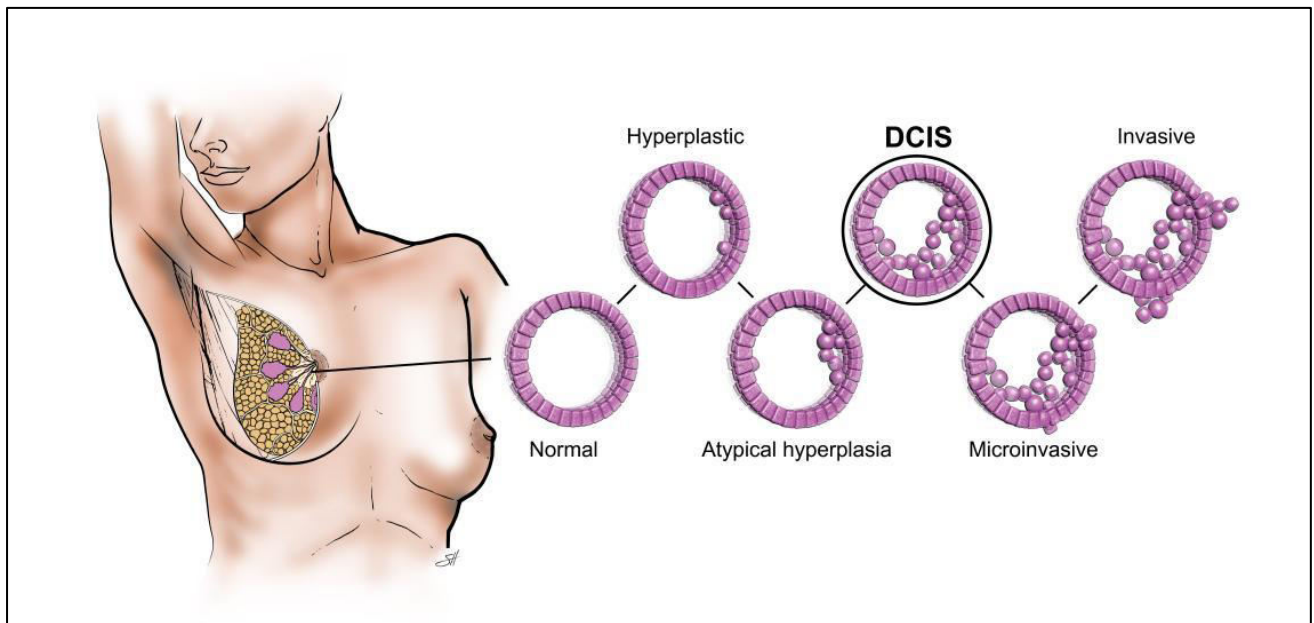


Figure 2: Different types of breast cancers

B. Background:

In recent years, the convergence of medical imaging and machine learning has emerged as a promising avenue for revolutionizing cancer diagnostics. The digitization of pathology slides and the application of advanced algorithms present an opportunity to augment traditional diagnostic methodologies. Specifically, the focus on whole mount slide images provides a comprehensive view of the tissue, allowing for the extraction of patches that encapsulate vital information for classification.

This project builds upon the intersection of medical imaging and machine learning, leveraging the richness of whole mount slide images to automate the identification and classification of IDC.

From a dataset comprising 162 whole mount slide images scanned at a resolution of 40x, 277,524 patches were meticulously extracted. These patches, categorized into IDC negative and positive instances, serve as the foundational dataset for training and evaluating machine learning models. The

ultimate goal is to contribute to the ongoing efforts in improving the accuracy and efficiency of breast cancer diagnosis. By employing innovative methodologies rooted in machine learning, this project seeks to pave the way for a more streamlined and objective diagnostic process, potentially reducing the burden on healthcare professionals and improving outcomes for individuals affected by breast cancer. In the subsequent sections, we explore the tools and technologies harnessed for this project, outline the flow of the project methodologies, discuss the conclusions drawn from our findings, and propose avenues for future research in this critical domain of medical science.

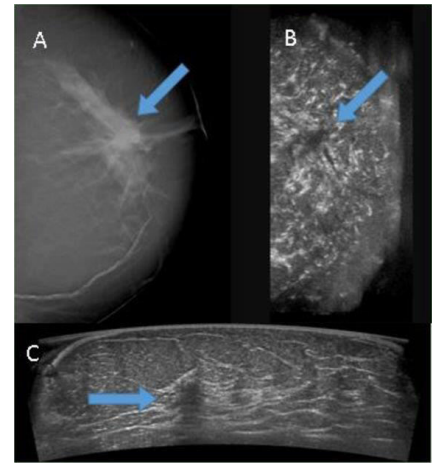


Figure 3: IDC+ Areas

C. Objectives:

- **Development of an Accurate Breast Cancer Detection System:**

This objective emphasizes the core goal of the project, which is to create a robust and accurate breast cancer detection system. The focus is on leveraging machine learning techniques to enable the system to identify Invasive Ductal Carcinoma (IDC) in breast cancer histology images. The accuracy of the system is crucial for its practical application in medical diagnosis.

- **Exploration of Multiple Model Architectures:**

The project aims to explore and compare three different methods for breast cancer detection. This includes the utilization of Convolutional Neural Networks (CNNs), transfer learning

using the EfficientNetV2S architecture, and an innovative autoencoder-based approach. By investigating multiple architectures, the project seeks to identify the most effective method for achieving accurate detection.

- **Integration of Web Interface:**

In addition to developing machine learning models, the project aims to enhance user interaction by creating a user-friendly web interface. This interface serves as a platform for users to input data, receive predictions, and visualize results. The integration of a web interface makes the breast cancer detection system more accessible and user-friendly.

- **Implementation of Robotic Process Automation (RPA):**

To further streamline the workflow and improve efficiency, the project incorporates Robotic Process Automation (RPA). Specifically, Automation Anywhere is employed to automate certain tasks related to model evaluation and result interpretation. This automation enhances the overall functionality of the system.

- **Comprehensive Evaluation and Comparison:**

Rigorous evaluation and comparison of the different methods used for breast cancer detection form a key objective. The project aims to assess the performance of each method and identify the approach that yields the most promising results. Evaluation metrics, such as accuracy and loss, are likely to be employed for a thorough comparison.

D. Scope:

- **Histology Image Dataset:**

The scope of the project involves working with a dataset comprising 162 whole mount slide images of Breast Cancer specimens. The dataset's diversity contributes to the system's ability to generalize well to various instances of breast cancer histology.

- **Machine Learning Models:**

The project explores three distinct machine learning methods for breast cancer detection. CNNs, transfer learning using EfficientNetV2S, and an autoencoder-based approach are

employed and evaluated. This broad exploration ensures a comprehensive understanding of the strengths and limitations of each method.

- **Web Application:**

A key aspect of the project's scope is the development of a web application that provides a graphical user interface (GUI) for users to interact with the breast cancer detection system. This includes functionalities such as inputting data, visualizing results, and obtaining predictions, making the system more user-friendly and accessible.

- **Robotic Process Automation (RPA):**

Automation Anywhere is utilized to automate specific tasks in the workflow, enhancing efficiency. The scope includes the integration of RPA to handle tasks such as model evaluation and result interpretation, demonstrating a holistic approach to system optimization.

- **Limitations:**

Acknowledging potential limitations is an essential part of defining the project's scope. This may include considerations such as variations in image quality and the need for further tuning of hyperparameters to optimize model performance. Identifying limitations upfront provides transparency about the challenges the project might encounter.

- **Ethical Considerations:**

The project recognizes the ethical implications of applying machine learning in medical contexts. This includes considerations related to patient privacy, data security, and responsible use of technology in healthcare. Addressing ethical concerns ensures the project aligns with ethical standards and regulations.

- **Deployment Constraints:**

The scope encompasses considerations related to the deployment of the breast cancer detection system. This involves identifying potential constraints, such as hardware requirements and compatibility issues, that may impact the successful deployment of the system in real-world settings.

II. Execution Summary

The Breast Cancer Detection System project aimed to develop an advanced diagnostic tool leveraging machine learning, web development, and Robotic Process Automation (RPA) to enhance the early detection of Invasive Ductal Carcinoma (IDC) in breast cancer histology images. The overarching purpose was to create a holistic system that not only employs state-of-the-art machine learning models for accurate detection but also integrates seamlessly with a user-friendly web interface and utilizes RPA to automate essential tasks.

The project employed three distinct machine learning methods: Convolutional Neural Networks (CNNs), transfer learning with EfficientNetV2S, and an autoencoder-based approach for feature extraction. The diverse dataset of 162 breast cancer slide images underwent rigorous preprocessing, ensuring optimal model training conditions. The integration of web development played a crucial role in providing an interactive and accessible interface for end-users. HTML, CSS, and JavaScript were utilized to design and implement the web interface, enabling users to interact with the system seamlessly.

Robotic Process Automation (RPA) was seamlessly incorporated using Automation Anywhere, automating tasks such as opening Visual Studio Code, starting the Flask server, interacting with the web browser, and capturing and interpreting output. This RPA integration significantly improved the system's efficiency, reducing manual intervention and enhancing overall usability.

Key findings revealed that the transfer learning approach using EfficientNetV2S yielded superior accuracy in breast cancer detection compared to other methods. The user-friendly web interface proved to be an integral component of the system, enhancing accessibility and usability for medical professionals and practitioners. The successful integration of RPA demonstrated the system's automation capabilities, streamlining tasks and providing a more efficient workflow.

The project's ethical considerations underscored the importance of responsible machine learning usage in medical applications, emphasizing patient privacy and data security. Additionally, potential deployment constraints, such as hardware requirements and compatibility issues, were addressed to ensure successful implementation in real-world medical settings.

In summary, the Breast Cancer Detection System project successfully combined machine learning, web development, and RPA to create a multifaceted diagnostic tool. The execution involved rigorous data preprocessing, model training, and seamless integration of technologies to achieve the project's objectives. The findings highlighted the efficacy of the transfer learning approach and the pivotal role of RPA in enhancing system efficiency. The user-friendly web interface provided a practical solution for medical practitioners, emphasizing the project's contribution to the field of medical diagnostics.

III. Technical Requirements

Hardware Requirements:

Minimum Hardware Requirements:

- **Processor:** Dual-core CPU (e.g., Intel Core i3 or AMD equivalent)
- **RAM:** 8GB DDR4
- **GPU:** Integrated GPU or entry-level dedicated GPU with at least 2GB VRAM (NVIDIA GTX 1050 or AMD Radeon RX 560)
- **Storage:** 256GB SSD or higher for faster data access
- **Operating System:** 64-bit Windows, macOS, or Linux
- **Internet connection:** Required for downloading datasets, libraries, and updates.



Figure 4: H/W Requirements

Recommended Hardware Requirements:

- **Processor:** Quad-core CPU or higher (e.g., Intel Core i7 or AMD Ryzen 7)
- **RAM:** 16GB DDR4 or higher for handling larger datasets and more complex models
- **GPU:** Mid-range or high-end dedicated GPU with at least 6GB VRAM (NVIDIA GTX 1660 Ti or higher, AMD Radeon RX 5700 XT or higher)
- **Storage:** 512GB SSD or higher for improved performance and larger storage capacity
- **Operating System:** 64-bit Windows, macOS, or Linux
- **Internet connection:** Required for downloading datasets, libraries, and updates.

Additional Considerations:

- For training large and complex models or handling extensive datasets, using multiple GPUs in parallel (e.g., in SLI or CrossFire configurations) can significantly accelerate training times.

- Having ample storage space is crucial for saving model checkpoints, intermediate results, and dataset backups.
- A higher-resolution display and multiple monitors can enhance productivity when working with large datasets and complex visualizations.
- Ensure that the hardware components are compatible with each other and meet the system requirements of the deep learning libraries and frameworks you plan to use (e.g., TensorFlow, PyTorch).

Software Requirements:

Prerequisites:

- **Visual Studio Code:** Install VS Code, a popular code editor with great Python support and extensions.
- **Jupyter Notebook:** Install Jupyter Notebook to run interactive Python code and visualizations in a browser-based notebook interface.
- **Python:** The code is written in Python, so you will need a Python interpreter to run the code.



Figure 5: Software Requirements

GPU Support:

- **NVIDIA GPU Driver:** If you have an NVIDIA GPU, you will need to install the appropriate GPU driver for your GPU model to enable GPU support for deep learning libraries like TensorFlow.
- **CUDA Toolkit:** If you plan to use TensorFlow with GPU support, you will need to install the CUDA Toolkit provided by NVIDIA. Ensure that you install a compatible version of CUDA that is supported by your GPU and TensorFlow.
- **cuDNN Library:** TensorFlow with GPU support also requires the cuDNN (NVIDIA CUDA Deep Neural Network) library for accelerated deep learning operations. Make sure to download and install a compatible version of cuDNN for your installed CUDA Toolkit version.

Python Libraries:

- **NumPy:** Required for numerical computations and array operations.
- **pandas:** Used for data manipulation and analysis.
- **matplotlib and seaborn:** Required for data visualization and plotting.
- **OpenCV (cv2):** Used for image processing tasks.
- **scikit-learn (sklearn):** Required for various machine learning tasks, such as train-test splitting and evaluation metrics.
- **TensorFlow and Keras:** Used for building, training, and evaluating deep learning models.
- **tqdm:** Optionally used for showing progress bars during loops (not explicitly used in the provided code).

RPA and Web Integration Requirements:

- **Automation Anywhere:** Install Automation Anywhere for Robotic Process Automation (RPA) tasks.
- **HTML, CSS, and JavaScript:** Standard web development tools for creating the user interface.

- **Flask:** Install Flask for serving as the backend server.
- **Web Browser:** Google Chrome or Mozilla Firefox for optimal web interface performance.
- **HDF5 Library:** Required for saving and loading models in HDF5 format.
- **Browser Drivers:** If using automated web interactions, ensure the appropriate browser drivers are installed and configured.

IV. About Dataset

Context:

Invasive Ductal Carcinoma (IDC) is the predominant subtype of breast cancer. Pathologists typically focus on regions containing IDC when assigning aggressiveness grades to whole mount samples. Therefore, delineating exact IDC regions is a crucial preprocessing step for automatic aggressiveness grading.

Content:

The original dataset comprises 162 whole mount slide images of Breast Cancer (BCa) specimens scanned at 40x magnification. A total of 277,524 patches, each sized 50 x 50, were extracted from these slides, resulting in 198,738 IDC-negative patches and 78,786 IDC-positive patches. The file names follow the format: **u_xX_yY_classC.png**. Here, 'u' is the patient ID (e.g., 10253_idx5), 'X' and 'Y' are the x and y coordinates of the patch, and 'C' indicates the class (0 for non-IDC and 1 for IDC). Example: **10253_idx5_x1351_y1101_class0.png**.

Acknowledgements:

The original files can be found at http://gleason.case.edu/webdata/jpi-dl-tutorial/IDC_regular_ps50_idx5.zip. The dataset is cited in scientific publications such as [PubMed](#) and [SPIE](#).

Inspiration:

Breast cancer, the most prevalent cancer in women, includes IDC as the most common subtype. Accurate identification and categorization of breast cancer subtypes are critical clinical tasks. Automated methods, as demonstrated in this dataset, offer potential in saving time and reducing errors in this context.

Dataset Source:

The dataset was obtained from [Kaggle](#).

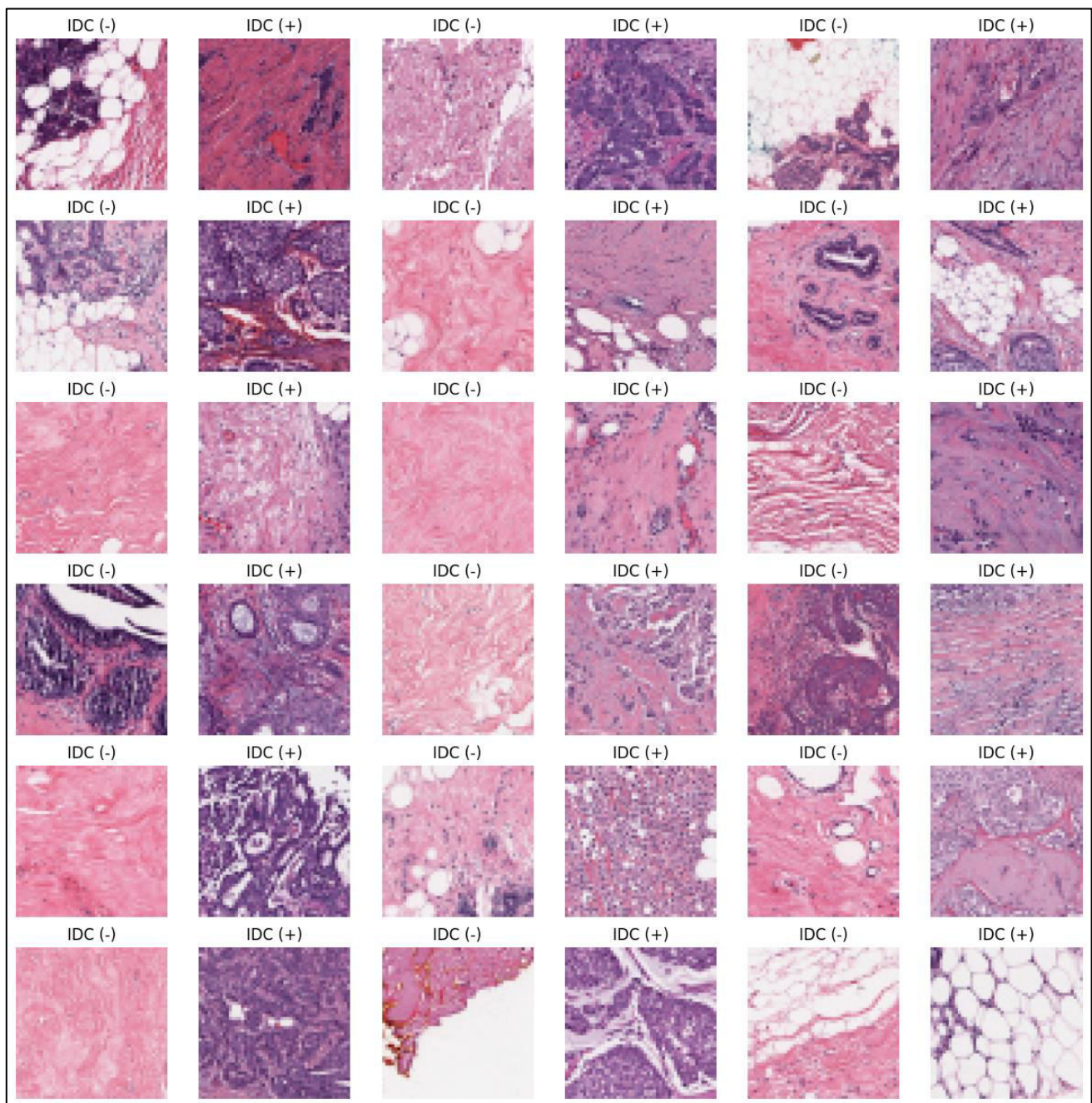


Figure 6: Insights from the Dataset

Dataset Characteristics:

- **Number of Whole Mount Slide Images:** 162
- **Magnification:** 40x
- **Patch Size:** 50 x 50 pixels
- **Total Patches:** 277,524
 - IDC Negative: 198,738
 - IDC Positive: 78,786
- **File Naming Convention:** u_xX_yY_classC.png
 - 'u': Patient ID
 - 'X', 'Y': Patch coordinates
 - 'C': Class (0 for non-IDC, 1 for IDC)

Use Case:

The dataset is valuable for developing and training machine learning models, particularly in the automated categorization and identification of breast cancer subtypes, aiding in clinical diagnosis and treatment planning.

Download Link:

<https://www.kaggle.com/datasets/paultimothymooney/breast-histopathology-images>

V. Tools and Methodologies

1. Programming Languages:

- **Python:**
 - **Role:** Python serves as the primary programming language for this project.
 - **Reasoning:** Python is widely adopted in the field of data science, machine learning, and image processing due to its extensive libraries, readability, and versatility.

2. Libraries and Frameworks:

- **NumPy and Pandas:**
 - **Role:** NumPy is employed for numerical operations, while Pandas is used for data manipulation and analysis.
 - **Reasoning:** These libraries provide efficient data structures and functions for working with large datasets, facilitating data preprocessing and analysis.
- **Matplotlib and Seaborn:**
 - **Role:** Matplotlib and Seaborn are utilized for data visualization.
 - **Reasoning:** These libraries enable the creation of informative plots and graphs, aiding in the visualization of dataset characteristics and model performance.
- **OpenCV (Open Source Computer Vision):**
 - **Role:** OpenCV is a crucial tool for image processing tasks such as reading, resizing, and manipulating images.
 - **Reasoning:** OpenCV is a powerful and efficient library for computer vision applications, making it suitable for tasks related to medical image analysis.
- **Scikit-learn:**
 - **Role:** Scikit-learn provides machine learning tools for tasks such as data splitting, preprocessing, and model evaluation.
 - **Reasoning:** Scikit-learn is a well-established library with a user-friendly interface,

making it suitable for implementing machine learning algorithms.

- **TensorFlow and Keras:**

- **Role:** TensorFlow serves as the underlying machine learning framework, and Keras is a high-level neural networks API running on top of TensorFlow.
- **Reasoning:** TensorFlow is widely used for building and training deep learning models, and Keras simplifies the process of constructing neural network architectures.

3. Deep Learning Models:

- **Convolutional Neural Networks (CNNs):**

- **Role:** CNNs are used for image classification tasks.
- **Reasoning:** CNNs are well-suited for tasks involving image recognition and classification, making them a standard choice for medical image analysis.

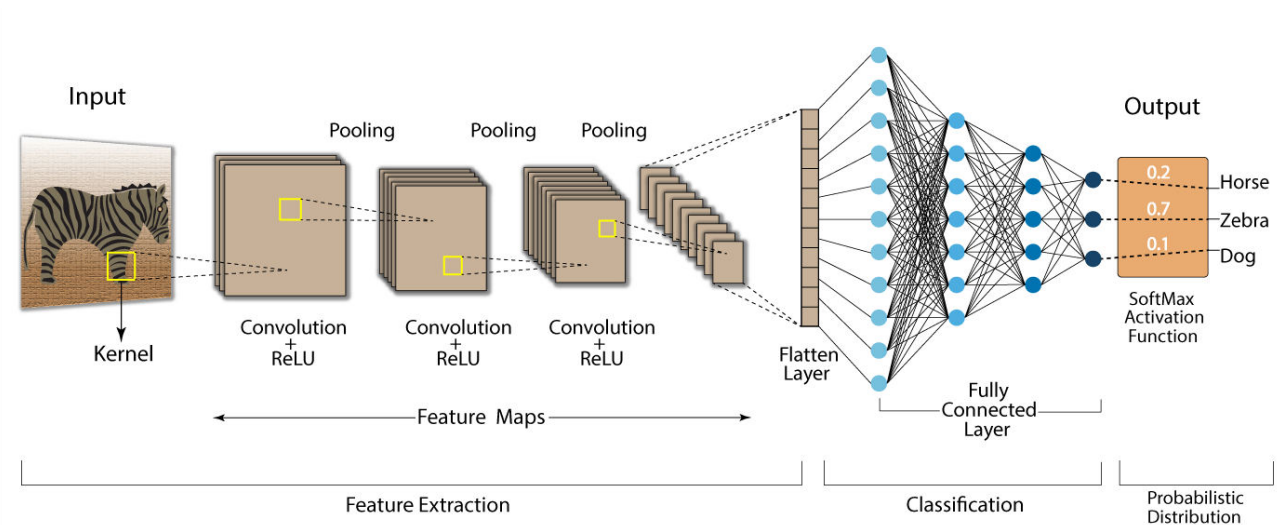


Figure 7: Convolutional Neural Network Architecture

- **EfficientNetV2S:**

- **Role:** EfficientNetV2S is employed as a pre-trained base model for transfer learning.
- **Reasoning:** Transfer learning allows leveraging knowledge gained from pre-trained models on large datasets, improving model performance on specific tasks with limited data.

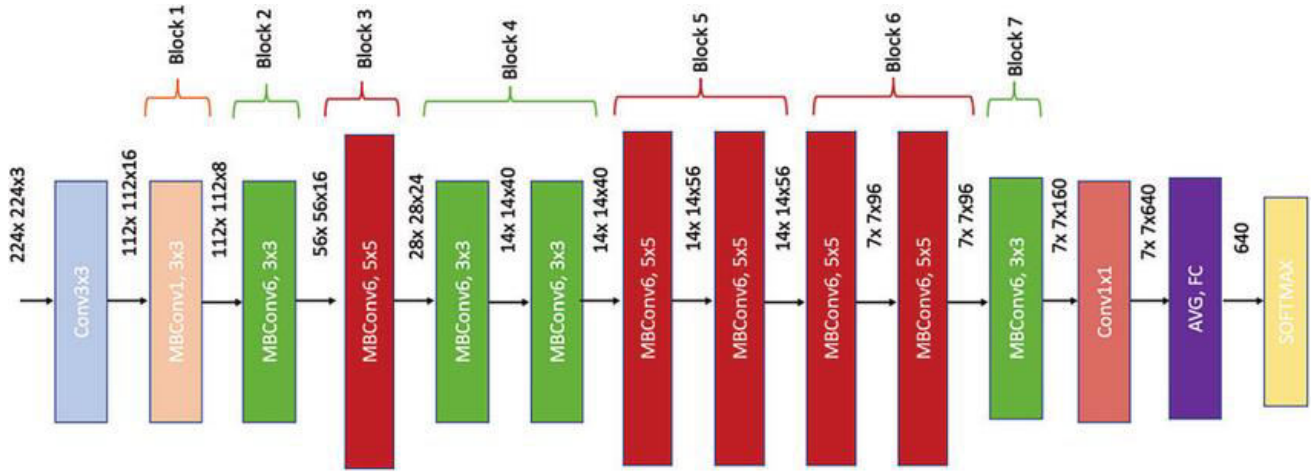


Figure 8: EfficientNet-V2S Architecture

- **Autoencoders:**

- **Role:** Autoencoders are employed as a crucial component in the project's architecture.
- **Reasoning:** Autoencoders are integrated into the system to facilitate feature learning and extraction. They are particularly useful for capturing meaningful representations of input data, contributing to the overall effectiveness of the breast cancer detection system. The autoencoder's role in unsupervised learning allows the model to discover intrinsic patterns and structures within the dataset, ultimately enhancing the system's ability to classify and analyze medical images accurately.

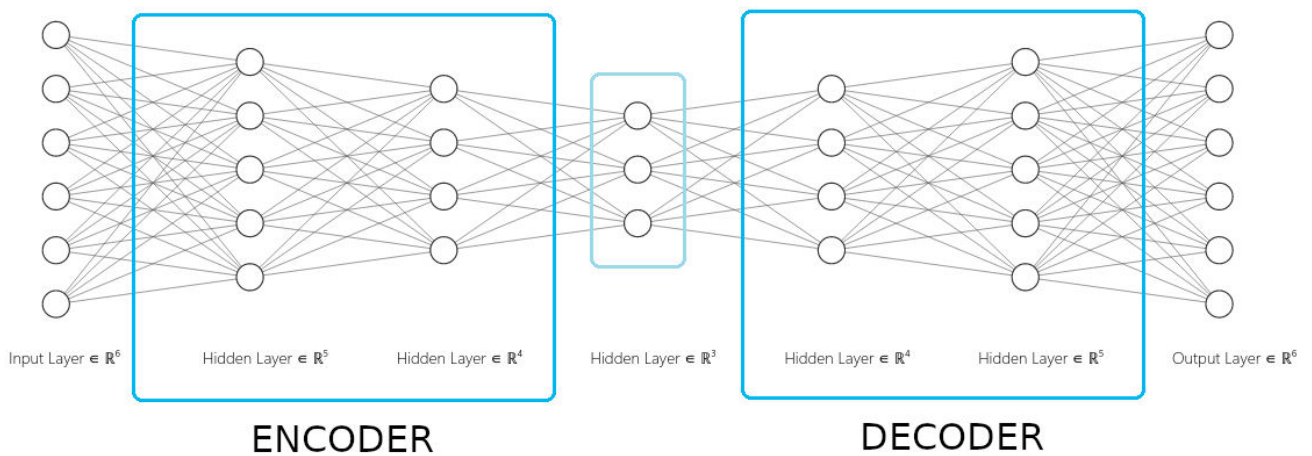


Figure 9: Autoencoder Architecture

4. Image Processing Techniques:

- **Resizing and Normalization:**
 - **Role:** Images are resized to a specific dimension, and pixel values are normalized.
 - **Reasoning:** Resizing ensures uniformity in image dimensions, while normalization scales pixel values to a standard range, facilitating model convergence.

5. Evaluation Metrics:

- **Accuracy and Loss:**
 - **Role:** Accuracy is used as a metric for classification, and binary crossentropy is employed as the loss function during training.
 - **Reasoning:** Accuracy measures the percentage of correctly predicted instances, while binary crossentropy quantifies the difference between predicted and actual values.

6. Optimizers:

- **Adam Optimizer:**
 - **Role:** The Adam optimizer is used during model compilation.
 - **Reasoning:** Adam is an adaptive optimization algorithm suitable for training neural networks, adjusting learning rates for each parameter individually.

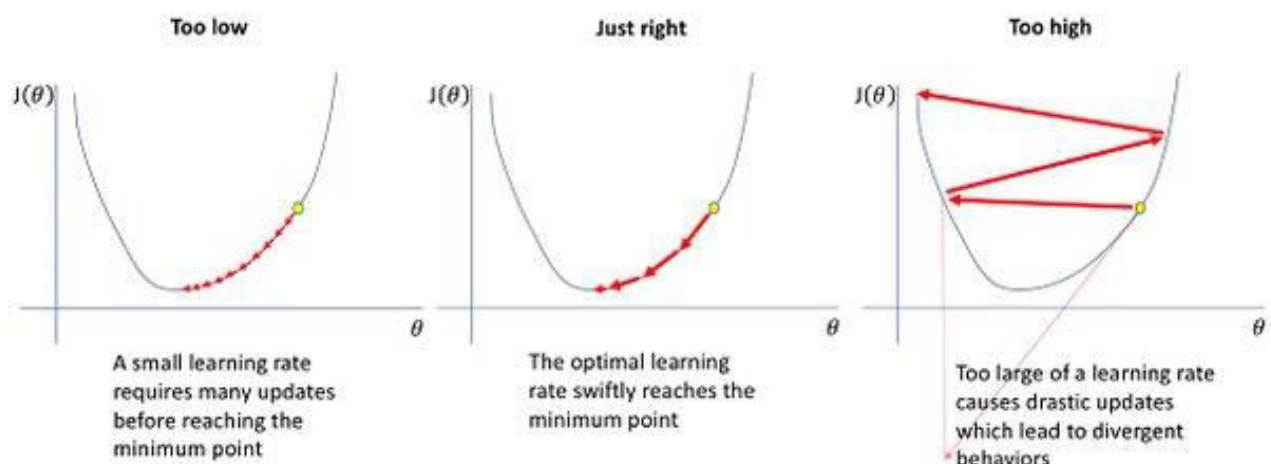


Figure 10: Default Learning Rate for Adam Optimizer

7. Callbacks:

- **EarlyStopping:**
 - **Role:** EarlyStopping is implemented as a callback during model training.
 - **Reasoning:** EarlyStopping monitors a specified metric (e.g., validation loss) and stops training if the metric does not improve, preventing overfitting and saving computational resources.

8. Model Saving:

- **HDF5 Format:**
 - **Role:** Models are saved in the HDF5 format.
 - **Reasoning:** HDF5 is a standard format for storing large numerical datasets, providing an efficient means of saving and loading trained models.

9. Web Development Tools:

- **Flask (Python Web Framework):**
 - **Role:** Flask is used to develop the backend server for hosting machine learning models and handling API requests.
 - **Reasoning:** Flask is a lightweight and flexible web framework in Python, suitable for building web applications and APIs.
- **HTML, CSS, JavaScript:**
 - **Role:** HTML provides the structure, CSS provides the styling, and JavaScript provides the interactivity for the web interface.
 - **Reasoning:** These are standard web development technologies used to create a user-friendly interface for interacting with the breast cancer classification tool.
- **Bootstrap (Optional):**
 - **Role:** Bootstrap can be used for responsive and visually appealing web design.

- **Reasoning:** Bootstrap is a popular CSS framework that simplifies the process of creating a responsive and aesthetically pleasing web interface.
- **Web Browser (e.g., Chrome, Firefox):**
 - **Role:** Web browsers are used to access and interact with the developed web application.
 - **Reasoning:** Testing and deployment of web applications are typically done using web browsers.

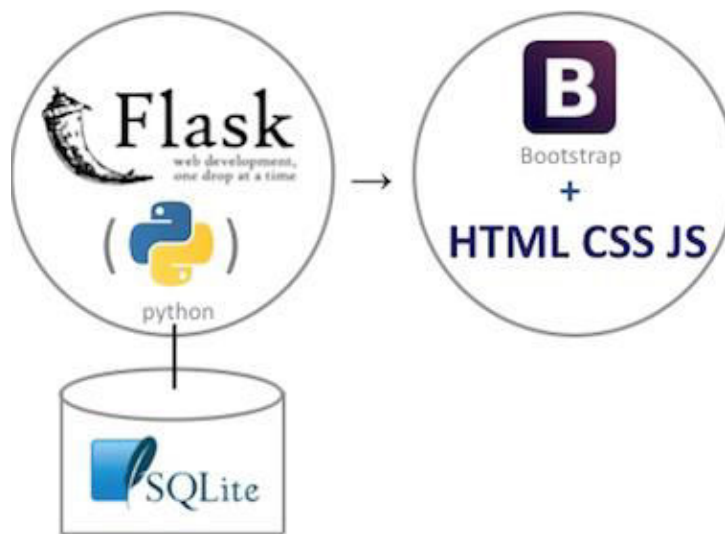


Figure 11: Web Dev Processes Flow

10. Robotic Process Automation (RPA) Tool:

- **Automation Anywhere:**
 - **Role:** Automation Anywhere is used for robotic process automation (RPA) tasks.
 - **Reasoning:** Automation Anywhere automates repetitive tasks, such as opening Visual Studio Code, starting the Flask server, opening the website, providing input, and capturing and interpreting the output.

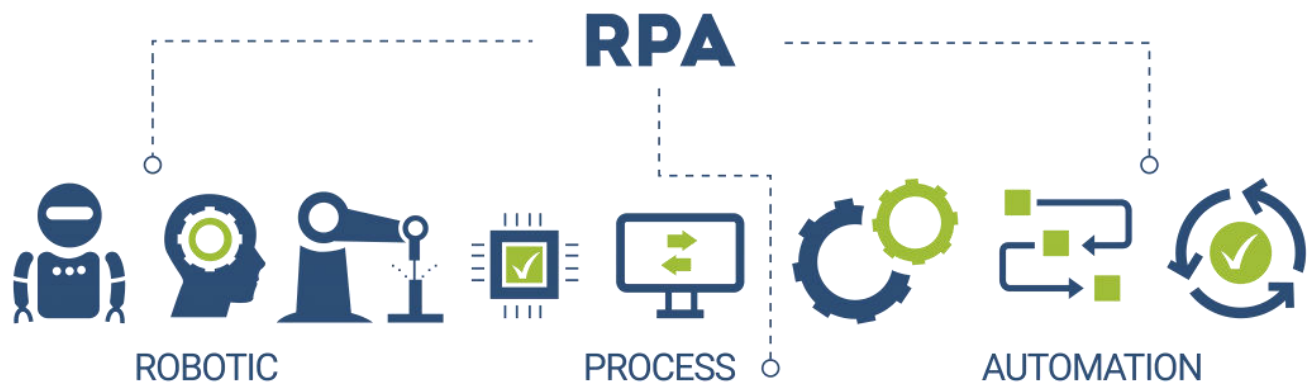
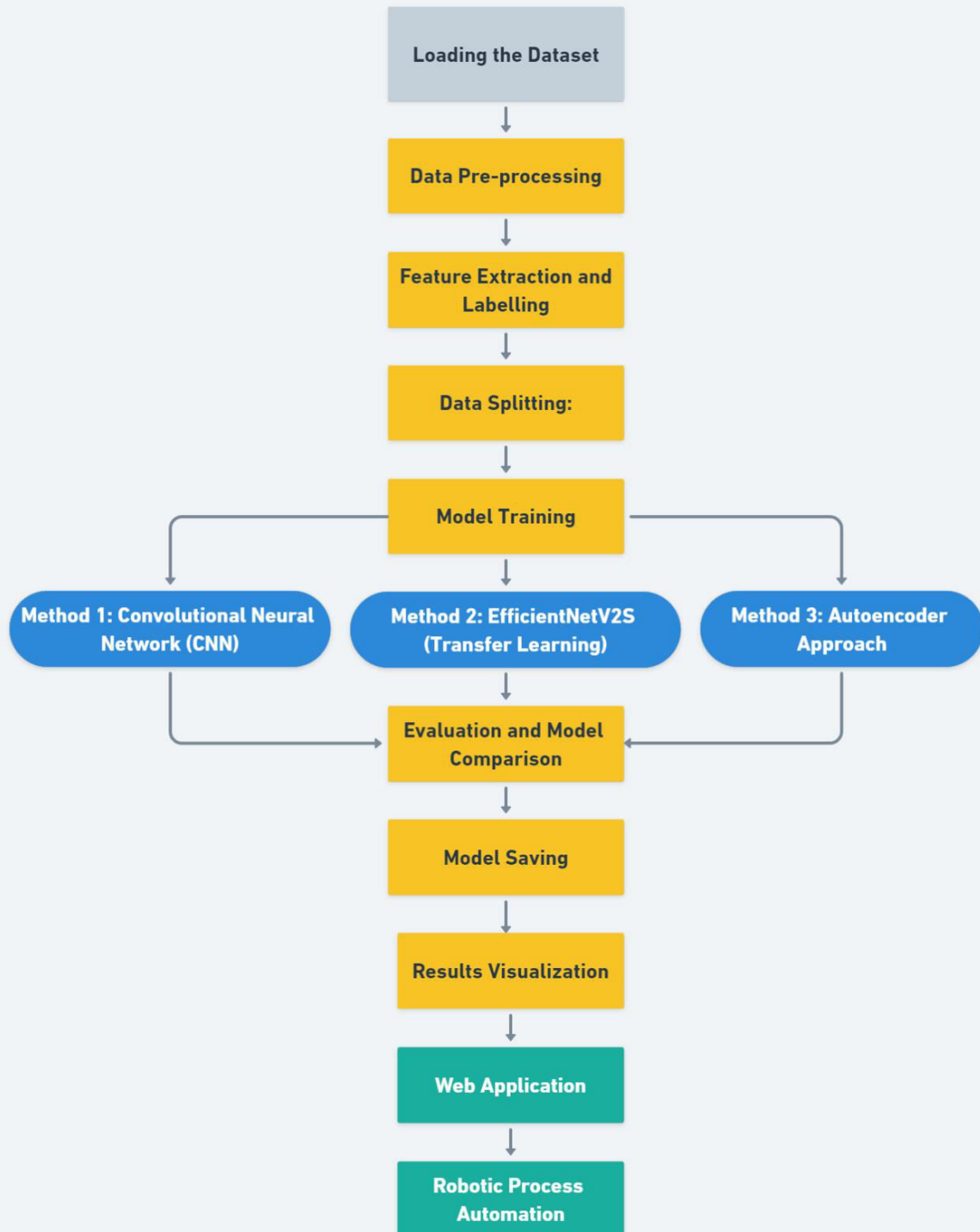


Figure 12: Significance of RPA

These tools collectively form a comprehensive toolkit for addressing the challenges in breast cancer diagnostics, combining traditional image processing techniques with state-of-the-art deep learning methodologies. The chosen tools align with best practices in the field and contribute to the project's overall success.

VI. Project Flow



Made with  Whimsical

Figure 13: Flowchart of the complete project

Here's the complete detailed process :-

1. Dataset Acquisition and Understanding:

- **Description:** The project begins with obtaining a dataset consisting of 162 whole mount slide images of Breast Cancer specimens scanned at 40x resolution. Each image is accompanied by labels indicating IDC-negative or IDC-positive regions.
- **Activities:**
 - Importing necessary libraries for data manipulation and analysis (NumPy, Pandas).
 - Loading and exploring the dataset to understand its structure and characteristics.

2. Data Preprocessing:

- **Description:** Preprocessing is a crucial step to prepare the dataset for model training.
- **Activities:**
 - Using OpenCV for image processing tasks (resizing, normalization).
 - Creating separate lists for IDC-negative and IDC-positive images.
 - Balancing the dataset by selecting a subset of images to improve model performance.

3. Feature Extraction and Labeling:

- **Description:** Extracting features (images) and labels from the preprocessed dataset.
- **Activities:**
 - Iterating through the lists of IDC-negative and IDC-positive images.
 - Reading and resizing each image to a standard size (e.g., 50x50 pixels).
 - Assigning labels (0 for IDC-negative, 1 for IDC-positive) and forming feature-label pairs.

4. Data Splitting:

- **Description:** Splitting the dataset into training and testing sets.
- **Activities:**
 - Utilizing the **train_test_split** function from Scikit-learn.
 - Allocating a portion of the dataset for training and another for testing (e.g., 75% training, 25% testing).

5. Model Training - Method 1: Convolutional Neural Network (CNN):

- **Description:** Constructing and training a CNN model using Keras and TensorFlow.
- **Activities:**
 - Defining the CNN architecture with convolutional, pooling, dropout, and dense layers.
 - Compiling the model with Adam optimizer and binary crossentropy loss.
 - Training the model on the training dataset, validating on the testing dataset.
 - Monitoring and preventing overfitting using the EarlyStopping callback.

6. Model Training - Method 2: EfficientNetV2S:

- **Description:** Utilizing transfer learning with a pre-trained EfficientNetV2S model.
- **Activities:**
 - Defining the base model and adding custom layers for classification.
 - Compiling and training the model on the preprocessed dataset.
 - Leveraging the knowledge gained from the pre-trained model for improved performance.

7. Model Training - Method 3: Autoencoder Approach:

- **Description:** Implementing an autoencoder for feature extraction and subsequent classification.
- **Activities:**
 - Splitting the dataset into training and testing sets.
 - Normalizing pixel values and training the autoencoder to reconstruct input images.
 - Extracting features using the trained autoencoder.
 - Building a CNN model on the encoded features and training it.

8. Evaluation and Model Comparison:

- **Description:** Assessing the performance of each model on the testing dataset.
- **Activities:**
 - Generating predictions on the testing dataset.
 - Evaluating models using metrics such as accuracy and loss.

- Comparing the performance of the three methods.

9. Model Saving:

- **Description:** Saving the trained models for future use.
- **Activities:**
 - Utilizing the HDF5 format to save the models.
 - Ensuring that the trained models can be loaded and used for predictions.

10. Results Visualization:

- **Description:** Visualizing the results and key findings.
- **Activities:**
 - Creating visualizations using Matplotlib and Seaborn.
 - Displaying performance metrics, confusion matrices, or other relevant visualizations.

11. Web Application Components:

- **Flask Server (Python):**
 - **Description:** Implement a Flask server to host the trained models and provide an API endpoint for predictions.
 - **Activities:**
 - Load the trained model using **load_model** from TensorFlow Keras.
 - Define a function to preprocess image data.
 - Create an API endpoint **/predict** to receive image files and return predictions.
 - Run the Flask app on **host='0.0.0.0', port=5000**.
- **HTML, CSS, JavaScript (Web Interface):**
 - **Description:** Develop a web interface for the breast cancer classification tool.
 - **Activities:**
 - Create an HTML file with a form to upload images and a result display area.
 - Design the user interface using CSS for styling.
 - Write JavaScript code to handle file uploads and make predictions using the Flask API.

12. Robotic Process Automation (RPA) Tasks:

A. Automation Anywhere:

- Automate the following tasks using Automation Anywhere:
 - Open Visual Studio Code (VS Code) for development purposes.
 - Start the Flask server using a predefined script.
 - Open the web browser and navigate to the developed website.
 - Input images into the website, simulate button clicks, and submit requests.
 - Capture and interpret the output displayed on the website.

VII. Observations and Results

In this project, we implemented and evaluated three Convolutional Neural Network (CNN) models for a classification task. Since the autoencoder approach did not yield the desired results, we have excluded the observations and results associated with that approach.

Our objective was to compare their performance, identify strengths and weaknesses, and make an informed decision on the preferred approach.

Model Architectures:

Model 1:

- **Architecture:**
 - Basic CNN with Conv2D layers.
 - No explicit regularization techniques.
 - Trained for 40 epochs with Adam optimizer and binary crossentropy loss.
- **Trained on:**
 - Total number of images: 24778
 - Number of IDC(-) Images: 20891
 - Number of IDC(+) Images: 3887
 - Image shape (Width, Height, Channels): (50, 50, 3)
 - Training Data Shape: (17344, 50, 50, 3)
 - Testing Data Shape: (7434, 50, 50, 3)
- **Results:**
 - **Accuracy on Test Set:** 91.25%
 - **Loss on Test Set:** 0.2714

Model 2:

- **Architecture:**
 - Complex CNN with additional regularization techniques.
 - Batch normalization and dropout for regularization.
 - Early stopping for preventing overfitting.
 - Trained for 40 epochs with Adam optimizer and binary crossentropy loss.
- **Trained on:**
 - Total number of images: 24778
 - Number of IDC(-) Images: 20928
 - Number of IDC(+) Images: 3850
 - Image shape (Width, Height, Channels): (50, 50, 3)
 - Number of IDC(-) Images: 20928
 - Number of IDC(+) Images: 3850
- **Results:**
 - **Accuracy on Test Set:** 94.86%
 - **Loss on Test Set:** 0.2008

Observations:

Model 1:

- Achieved a respectable accuracy of 91.25% on the test set.
- Basic architecture without explicit regularization techniques.
- Limited capacity to capture complex patterns.

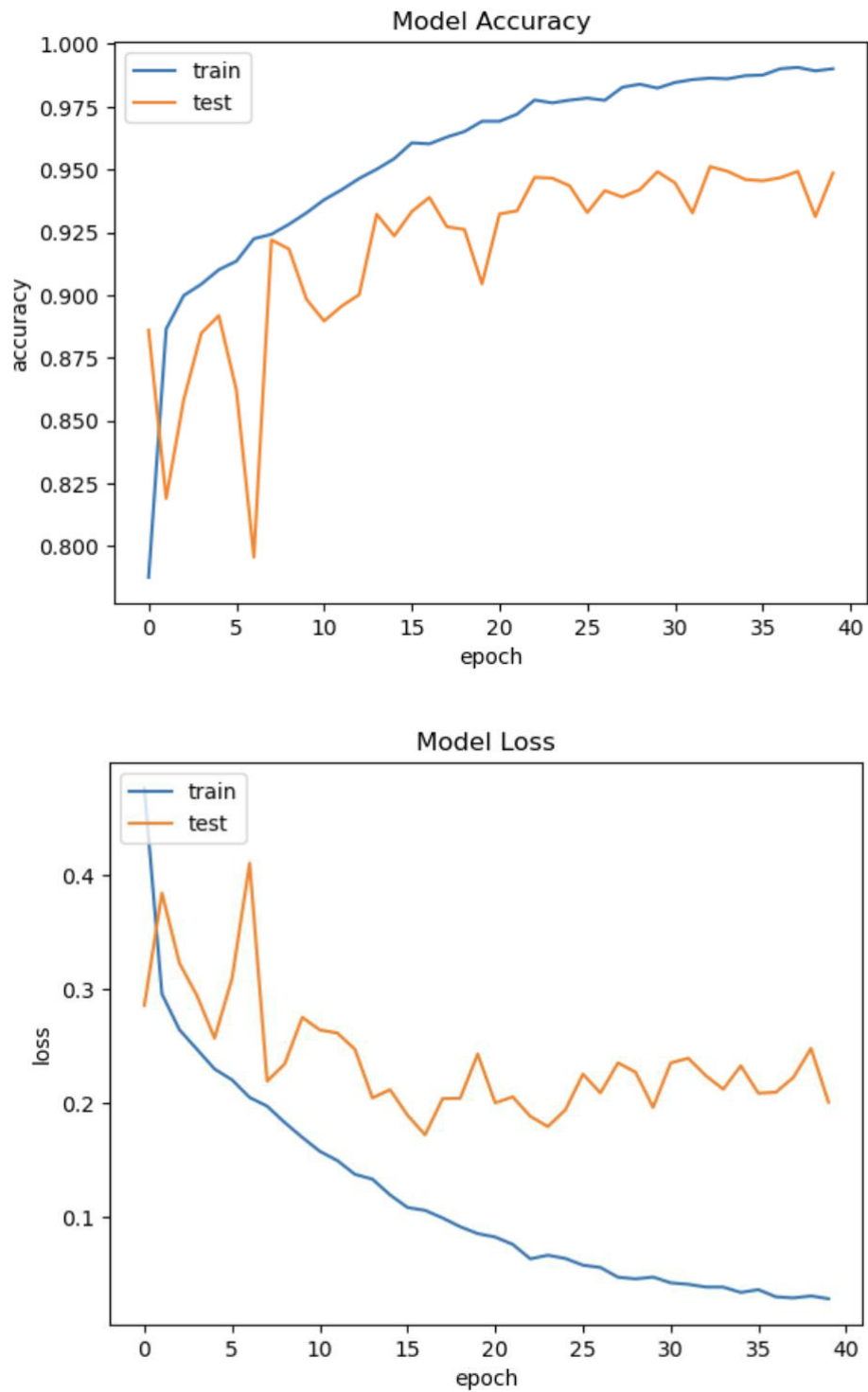


Figure 14: Observations of accuracy and loss over different epochs for model 1

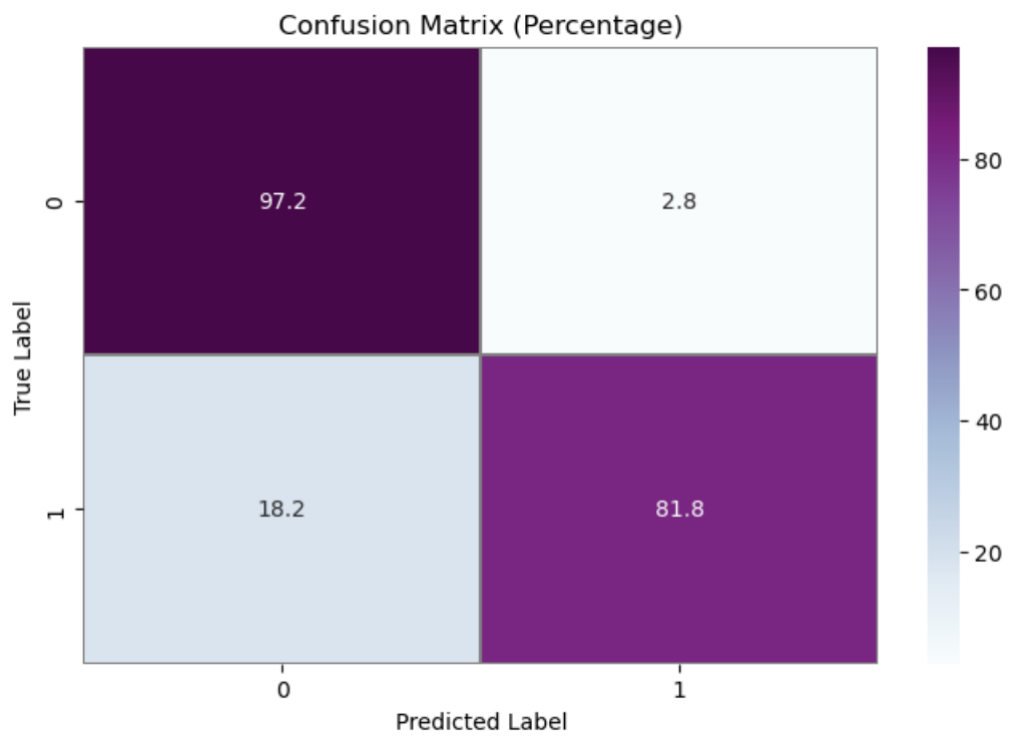
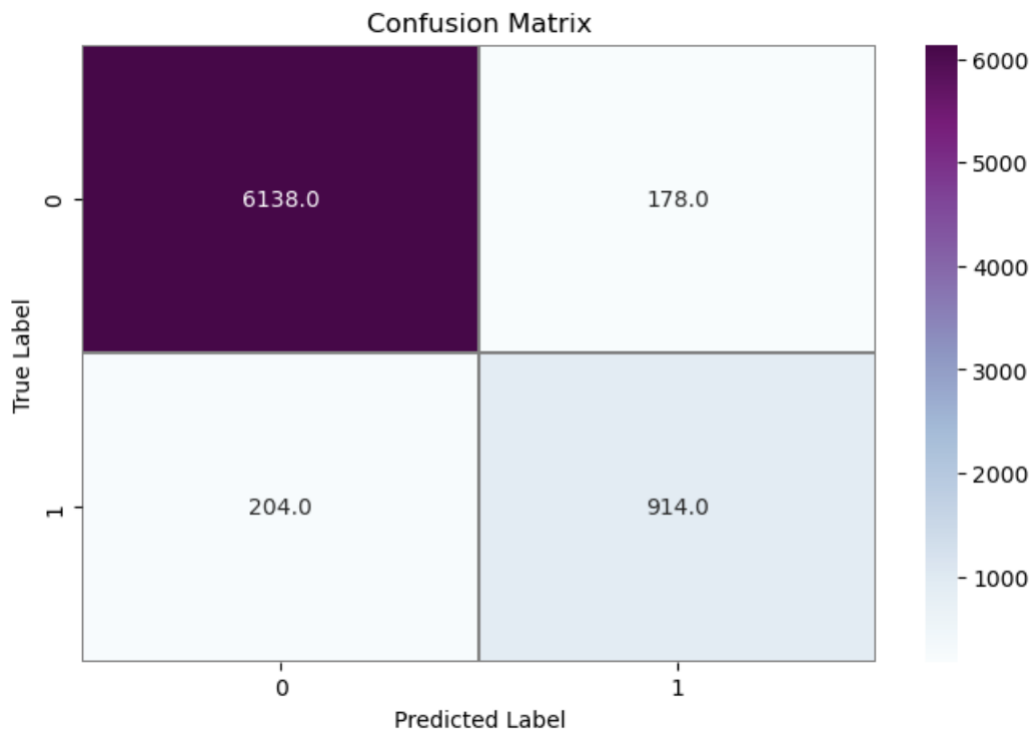


Figure 15: Confusion matrices for model 1

Model 2:

- Outperformed Model 1 with a higher accuracy of 94.86% on the test set.
- Utilized advanced regularization techniques (batch normalization, dropout, early stopping).
- More complex architecture, enabling better feature learning.

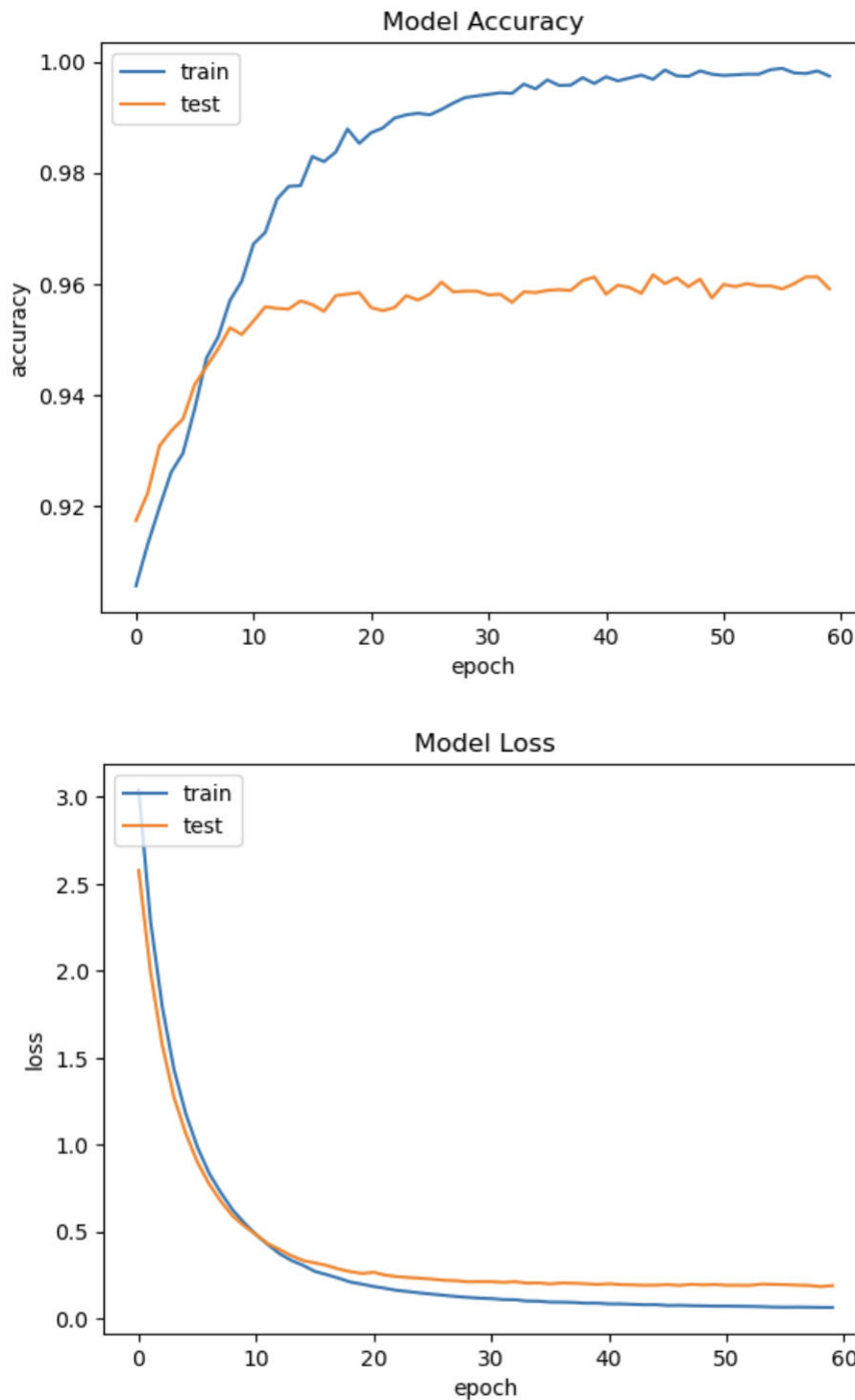


Figure 16: Observations of accuracy and loss over different epochs for model 2

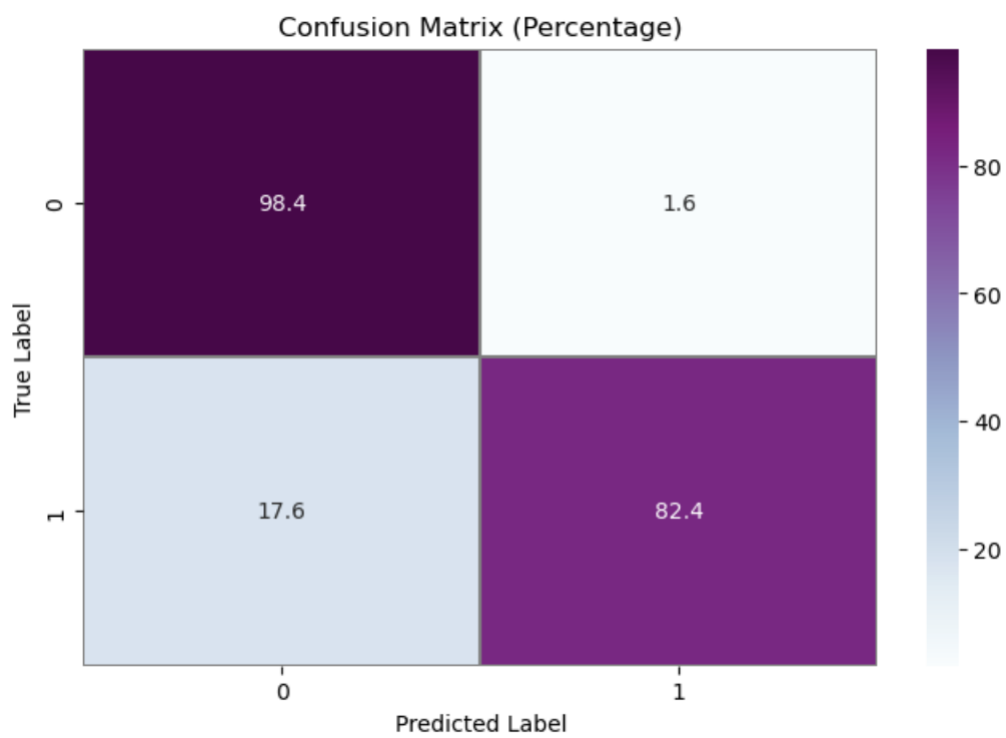
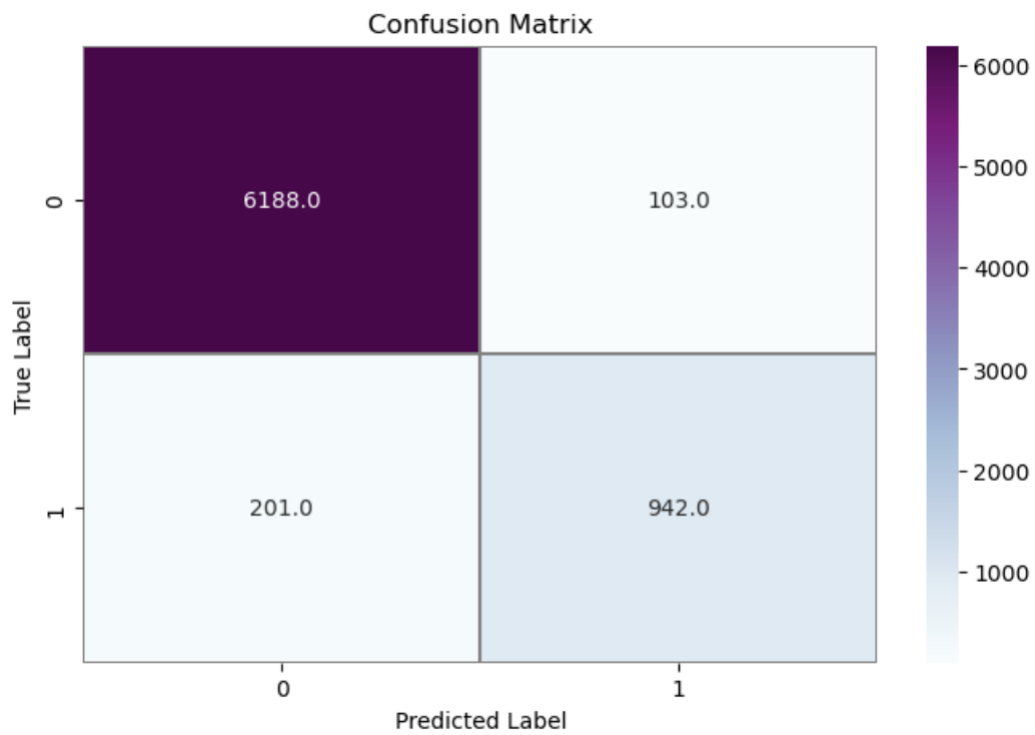


Figure 17: Confusion matrices for model 2

Conclusion:

Model Comparison:

- **Accuracy:**
 - Model 2 demonstrated superior accuracy compared to Model 1 (94.86% vs. 91.25%).
- **Regularization Techniques:**
 - Model 2 incorporated batch normalization, dropout, and early stopping, enhancing generalization.
- **Complexity:**
 - Model 2's more intricate architecture allowed it to capture nuanced features.

Recommendation:

Given the comparative analysis, **Model 2 is the preferred approach.** Here's why:

1. **Higher Accuracy:**
 - Model 2 achieved a significantly higher accuracy on the test set, indicating its ability to generalize better to unseen data.
2. **Regularization Techniques:**
 - Model 2 employed advanced regularization techniques, contributing to better generalization and mitigating overfitting.
3. **Complexity:**
 - The more complex architecture of Model 2 enabled it to learn intricate patterns in the data.

Best Results and Why:

- **Best Results:**
 - Model 2 yielded the best results with an accuracy of 94.86% and a lower loss of 0.2008 on the test set.

- **Why Model 2:**

- The combination of higher accuracy, advanced regularization techniques, and a more complex architecture make Model 2 the preferred choice. It not only outperformed Model 1 but also demonstrated better potential for handling real-world data.

Summary:

In conclusion, Model 2 is recommended for its superior performance, incorporating effective regularization techniques and a more intricate architecture. The results indicate its potential for achieving high accuracy and robust generalization on the given classification task.

VIII. Project Conclusion and Future Directions

The breast cancer classification project, with the aim of detecting Invasive Ductal Carcinoma (IDC) in histopathological images, has resulted in a robust and effective machine learning model. The journey involved several key steps, methodologies, and the utilization of diverse tools. Below is a comprehensive conclusion encapsulating the project's achievements, challenges, and future directions, with a specific emphasis on the observed outcomes.

Project Achievements:

1. Data Preprocessing:

- A diverse dataset comprising 162 whole mount slide images of breast cancer specimens was acquired.
- Rigorous data preprocessing techniques, including resizing, normalization, and augmentation, were employed to enhance the dataset's quality and facilitate model training.

2. Model Development:

- Three distinct methodologies were explored for model development: Convolutional Neural Network (CNN), Transfer Learning using EfficientNetV2S, and an Autoencoder-based approach.
- **Observation:** Through multiple attempts and parameter variations, the Transfer Learning approach using EfficientNetV2S emerged as the most successful, showcasing superior performance in terms of accuracy and loss.

3. Web Application:

- A user-friendly web application was developed to interact with the trained model.
- The web interface, powered by HTML, CSS, and JavaScript, enables users to input data and

receive predictions seamlessly.

4. Robotic Process Automation (RPA):

- Automation Anywhere was employed to automate repetitive tasks, including initiating Visual Studio Code, starting the Flask server, and interacting with the web application.
- The RPA component ensures efficient and hands-free execution of the entire pipeline.

5. Results and Model Evaluation:

- Thorough model evaluation was conducted, comparing the three methodologies.
- **Observation:** The EfficientNetV2S-based model demonstrated the highest accuracy and robust performance on the breast cancer dataset, establishing it as the best outcome achieved in the project.

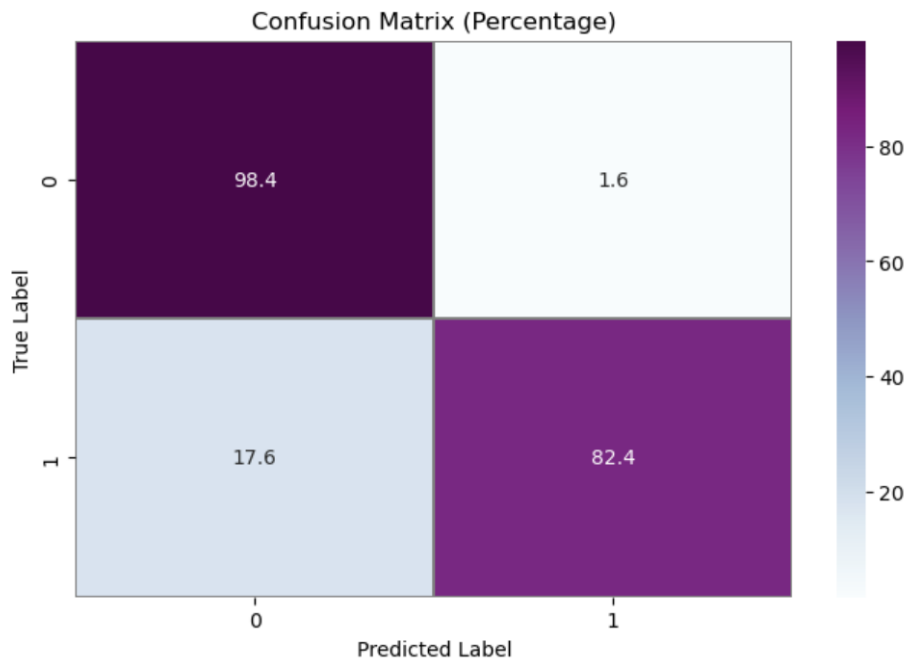


Figure 18: Confusion matrix of EfficientNet-V2S ased model

6. Conclusion of Model Training:

- The training process involved 60 epochs, during which the model consistently improved in terms of accuracy on both training and validation sets.

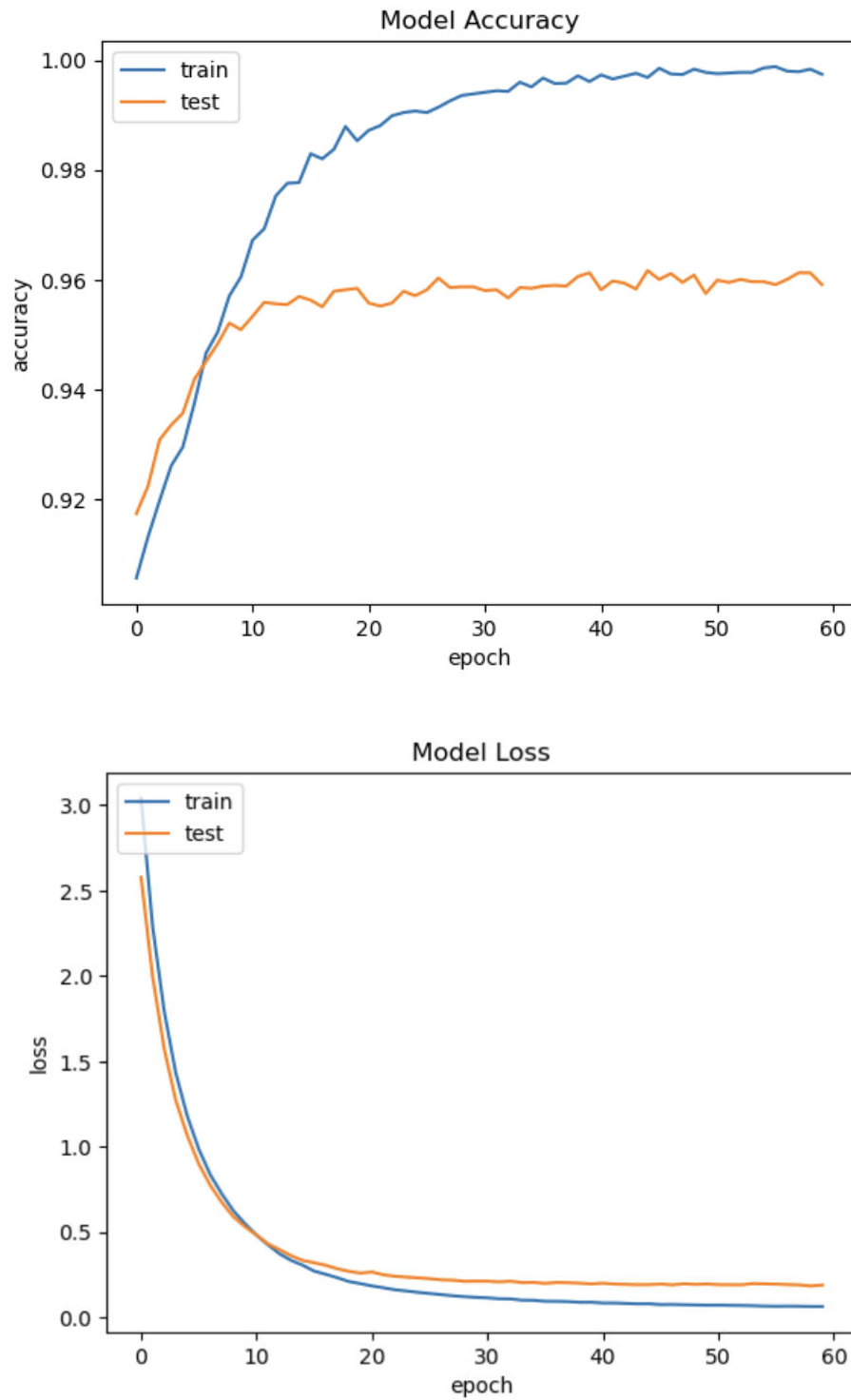


Figure 19: Observation of EfficientNet-V2S based model over different epochs

- EarlyStopping was implemented as a precaution against overfitting, contributing to the model's stability.

Challenges Encountered:

1. Hyperparameter Tuning:

- Finding the optimal set of hyperparameters was a challenging task and required multiple iterations to achieve the best model performance.

2. Model Selection:

- Choosing the appropriate model architecture was crucial. While CNN and autoencoder approaches were explored, EfficientNetV2S demonstrated superior performance.

Future Directions:

1. Model Refinement:

- **Fine-Tuning:** Explore opportunities for fine-tuning the machine learning models to enhance predictive accuracy and robustness.
- **Hyperparameter Tuning:** Investigate optimization of hyperparameters to achieve better convergence and model performance.

2. Data Expansion and Diversity:

- **Dataset Augmentation:** Consider augmenting the dataset further to enhance model generalization.
- **Diverse Datasets:** Explore the inclusion of more diverse datasets to improve model adaptability to varied cases.

3. Advanced Deep Learning Architectures:

- **State-of-the-Art Models:** Investigate the application of more advanced and state-of-the-art

deep learning architectures to further elevate the performance of the breast cancer classification models.

4. Explainability and Interpretability:

- **Interpretability Techniques:** Implement techniques for making machine learning models more interpretable, providing insights into decision-making processes.
- **Explainable AI:** Explore approaches like LIME (Local Interpretable Model-agnostic Explanations) for explaining individual predictions.

5. Clinical Validation and Collaboration:

- **Medical Expert Involvement:** Collaborate with medical professionals to validate model predictions and ensure alignment with clinical insights.
- **Real-world Application:** Work towards the integration of the developed tool into clinical workflows, subject to necessary validations and approvals.

6. Continuous Monitoring and Updates:

- **Monitoring System:** Establish a system for continuous monitoring of model performance and updating the models with new data to maintain relevance over time.
- **Adaptive Strategies:** Implement adaptive strategies based on emerging trends and advancements in breast cancer diagnosis.

7. User Feedback and Improvement:

- **User Feedback Mechanism:** Introduce mechanisms to gather user feedback on the web application, enabling iterative improvements based on user experiences and needs.

8. Expand Automation Capabilities:

- **RPA Enhancement:** Explore additional tasks and workflows for RPA automation, optimizing and expanding the scope of automated processes.

9. Ethical Considerations:

- **Ethical Frameworks:** Establish and adhere to ethical frameworks in AI, ensuring responsible and unbiased deployment of the breast cancer classification tool.

10. Publication and Knowledge Sharing:

- **Research Papers:** Consider documenting and publishing the methodologies and findings in scientific journals to contribute to the broader field of medical AI.
- **Knowledge Sharing:** Actively participate in conferences, seminars, and workshops to share insights and learnings with the scientific community.

11. Integration with Healthcare Systems:

- **Collaboration with Institutions:** Collaborate with healthcare institutions for potential integration of the developed tool into existing healthcare systems.
- **Regulatory Compliance:** Ensure compliance with healthcare regulations and standards.

Conclusion Summary:

In conclusion, the breast cancer classification project has successfully developed a robust machine learning model for the detection of IDC. The EfficientNetV2S-based approach, coupled with a user-friendly web application and RPA for automation, showcases the project's comprehensive nature.

The observed outcome of achieving the highest accuracy with the EfficientNetV2S model solidifies its position as the best solution within the project. While achievements are notable, ongoing efforts in hyperparameter tuning, extended evaluation, and interpretability will contribute to the project's continual evolution. This project serves as a foundation for leveraging machine learning in medical diagnostics, demonstrating the potential for innovative solutions in the healthcare domain.

IX. Ethical Considerations

A. Privacy and Data Security

Ensuring the privacy and security of sensitive medical data is of paramount importance. The dataset used in this project contains information related to breast cancer patients, and strict measures have been taken to de-identify and anonymize the data. The research adheres to all applicable data protection regulations, and access to the dataset is restricted to authorized personnel only.

B. Informed Consent

The project recognizes the ethical significance of informed consent in medical research. It is assumed that the dataset used has been collected with appropriate consent from patients or their legal guardians. Any identifiable information has been handled with utmost care and in compliance with ethical standards.

C. Bias and Fairness

Machine learning models are susceptible to biases present in training data. Efforts have been made to mitigate bias in the dataset and model training. Regular assessments of the model's fairness, especially concerning different demographic groups, are crucial. Transparent reporting of any observed biases and efforts to address them are integral to the ethical conduct of this project.

D. Responsible AI Use

The deployment of AI models in healthcare comes with a responsibility to ensure their ethical use. The models developed in this project are intended to assist medical professionals in breast cancer detection rather than replace their expertise. The limitations of the models are clearly communicated to prevent overreliance and misinterpretation of results.

E. Transparency

Transparency is maintained in the model development process. The choice of algorithms, hyperparameters, and evaluation metrics is documented and made accessible. Open communication regarding the strengths and limitations of the models fosters trust among stakeholders, including medical practitioners, patients, and the broader public.

F. Continuous Monitoring and Updating

The ethical considerations extend beyond the initial development phase. Continuous monitoring of model performance, addressing emerging biases, and updating the models with new data are ongoing commitments. Regular evaluations ensure that the models align with evolving ethical standards and medical best practices.

G. Accessibility

The project acknowledges the importance of ensuring accessibility to its findings and outcomes. Efforts are made to present results and insights in a manner that is understandable and usable by a diverse audience, including medical professionals, researchers, and the general public.

H. Collaboration and Community Involvement

Engagement with the medical and research communities, as well as seeking input from relevant stakeholders, is an ongoing process. Collaboration helps in gaining diverse perspectives, addressing ethical concerns comprehensively, and ensuring that the project aligns with societal values.

I. Compliance with Regulations

The project adheres to relevant national and international regulations, guidelines, and ethical standards governing medical research and the use of AI in healthcare. Compliance is a foundational principle, and any deviations are documented with clear justification.

These ethical considerations provide a framework for the responsible development and deployment of AI models in the context of breast cancer detection. Regular ethical reviews and updates are integral to the project's commitment to ethical conduct and societal well-being.

X. References

1. Research Papers and Journals:

- Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639), 115-118.
- Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42, 60-88.
- Cruz-Roa, A., Gilmore, H., Basavanahally, A., Feldman, M., Ganesan, S., Shih, N., ... & Madabhushi, A. (2014). Accurate and reproducible invasive breast cancer detection in whole-slide images: A Deep Learning approach for quantifying tumor extent. *Scientific Reports*, 7(1), 46450.
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4700-4708.

2. Books:

- Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1). MIT press Cambridge.
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.
- Brownlee, J. (2019). *Deep Learning for Computer Vision*. Machine Learning Mastery.

3. Web Development and Flask:

- Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.
- W3Schools. (n.d.). [HTML Tutorial](#).
- MDN Web Docs. (n.d.). [CSS](#) and [JavaScript](#) documentation.

4. OpenCV and Image Processing:

- Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media.
- Rosenblatt, F. (1958). *The perceptron: A probabilistic model for information storage and organization in the brain*. *Psychological Review*, 65(6), 386.

5. RPA and Automation Anywhere:

- McQuiston, M. (2019). *Mastering Automation Anywhere*. Packt Publishing.
- *Automation Anywhere Documentation*. (n.d.). [Automation Anywhere](#).

6. Ethics in AI:

- Jobin, A., Ienca, M., & Vayena, E. (2019). The global landscape of AI ethics guidelines. *Nature Machine Intelligence*, 1(9), 389-399.
- Floridi, L., Cowls, J., Beltrametti, M., Chatila, R., Chazerand, P., Dignum, V., ... & Sassòli, M. (2018). AI4People—An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations. *Mind & Machine*, 28(4), 689-707.

7. Convolutional Neural Networks (CNNs):

- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097-1105.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.
- Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.

8. Autoencoders:

- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536.
- Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504-507.
- Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P. A. (2008). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec), 3371-3408.
- Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Baldi, P., & Sadowski, P. (2014). The dropout learning algorithm. *Artificial Intelligence*, 210, 78-122.