

Senior Design Project II

Breast Cancer Detection Model

Analysis & Design Report

Yasemin Demirkaya

Supervisor: Zeynep Filiz Eren

June 23, 2023

Contents

1 Introduction

2 Motivation

3 Literature Review

4 Method

5 Technologies and Tools

6 Project Implementation

7 Results Of Models

8 Future Work

9 References

Breast Cancer Detection Model

1 Introduction

Day by day, cancer has become a disease that we hear and witness more and more in our lives. According to International Agency for Research on Cancer data's global estimated number for new cancer cases in 2020 are almost 20 million people. Breast cancer constitutes 11 percent of this data. Breast cancer is one of the most common causes of death among women worldwide. Early detection helps in reducing the number of early deaths. So, millions of women get mammograms each year. Mammography is the most effective method of detecting breast cancer early. Additionally, this instrument enables the detection of additional illnesses and may provide information about the nature of cancer, such as benign, malignant, or normal. Clearly A useful machine learning tool could help many people. Machine learning algorithms can be used for medical oriented research, it advances the system, reduces human errors and lowers manual mistakes.

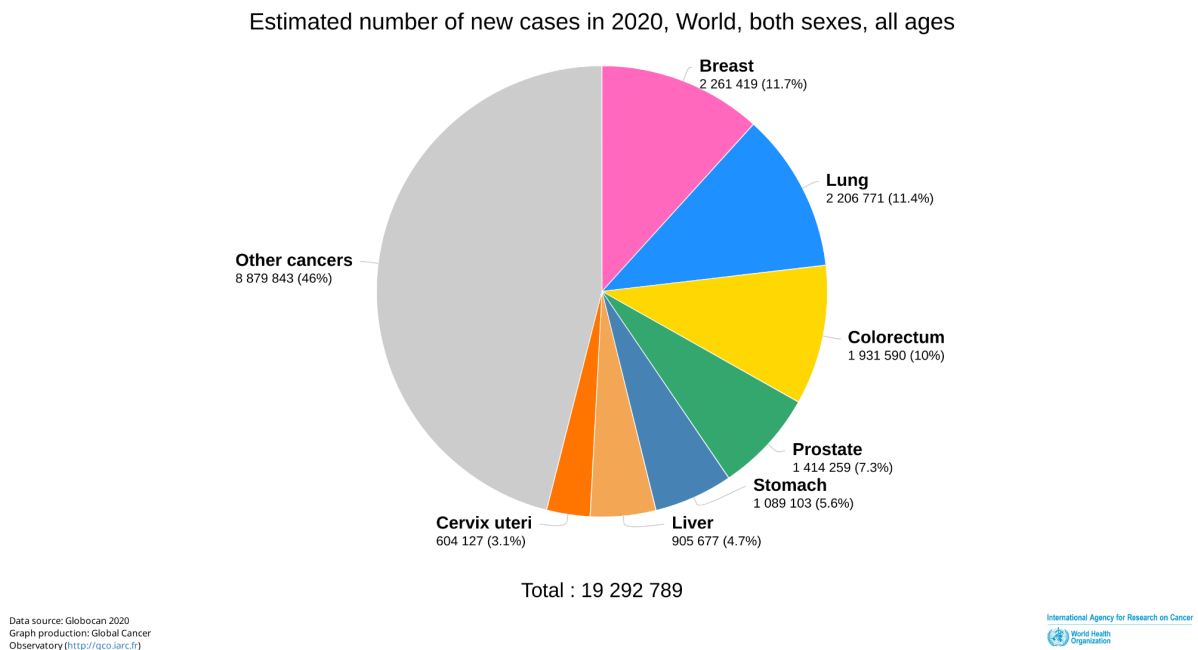


Figure 1: Estimated number of new cases in 2020

2 Motivation

The situation that led me to choose this project is, primarily, the occurrence of cancer in my family. Second is, I want to learn Image processing, Machine learning and Deep learning/Neural Networks for my future. I wanted to work in this project because I was more excited to develop myself in these subjects by doing a project in the field of health.

3 Literature Review

Machine learning is an application of AI (AI) that gives systems the power to automatically learn and improve from experience without being manually programmed[1]. Breast cancer detection is common topic to make machine learning models.

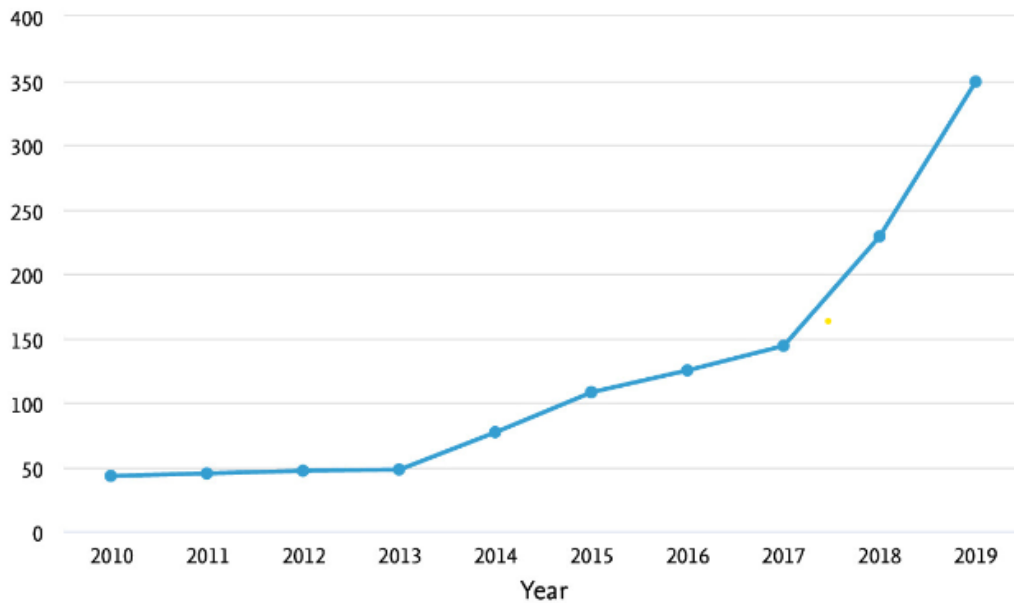


Figure 2: Graph of Machine Learning for Classification and Detection of Breast Cancer Publications

Most popular dataset for this challenge is Wisconsin Breast Cancer dataset. Many researches complete with this dataset. Agarap and Abien Fred M. published an article about comparing some Machine Learning algorithms with Wisconsin Breast Cancer dataset[2]. Researchers compared algorithms which are Linear Regression, Multilayer Perceptron, Nearest Neighbor search, Softmax Regression, and Support Vector Machine by measuring their classification test accuracy. Results show that all the presented Machine Learning algorithms performed well (all exceeded 90% test accuracy) on the classification task. The Multilayer Perceptron algorithm stands out among the implemented algorithms with a test accuracy of $\approx 99.04\%$. On the other hand, Naji and Mohammed Amine presented a comparison of five machine learning algorithms also: Support Vector Machine, Random Forest, Logistic Regression, Decision tree (C4.5) and K-Nearest Neighbours on the Wisconsin Diagnostic Breast Cancer dataset[3]. Researches observed that Support vector Machine outperformed all other classifiers and achieved the highest accuracy (97.2%). All their work is done in the Anaconda environment based on python programming language and Scikit-learn library.

Image preprocessing is important for the correct classification of disease images also. Researches shows that, classifying models to detecting breast cancer from dataset of images is based on machine learning and image processing techniques are required. This models combines image preprocessing, feature extraction, feature selection, and machine learning techniques to aid in the classification and identification breast cancer. For disease categorization and detection, the model makes use of the machine learning techniques such as least square support vector machine, KNN, random forest, and Naïve Bayes[4]. Mammogram images contain various types of noise. Researches used image filtering techniques to remove these noises. A geometric mean filter is used to remove noise from the input images on MIAS dataset.

Radiologists cannot easily provide accurate manual evaluation due to the huge number of mammograms generated in widespread screening. Therefore, a Computer Aided-Design system has been developed to detect the indicators of breast cancer and improve the accuracy of diagnosis [5]

Another article mention about One of the main challenges of breast cancer image processing; the lack of training data. To address this challenge and optimize the performance, they have utilized a transfer learning technique which is where the deep learning models train on a task, and then fine-tune the models for another task. They have employed transfer learning in two ways: Training their proposed model first on the same domain dataset, then on the target dataset, and training the model on a different domain dataset, then on the target dataset. They have empirically proven that the same domain transfer learning optimized the performance.[6]

4 Method

- **Data Collection**

For this project I have a couple of choices to collect dataset. So I decide to concatenate 3 different datasets for making predictions.

DDSM Mammography

This data collected from: <https://www.kaggle.com/datasets/skooch/ddsm-mammography>.

The dataset consists of negative images from the DDSM dataset and positive images from the CBIS-DDSM dataset. The data was pre-processed to convert it into 299x299 images. The data is stored as tfrecords files for TensorFlow in the link.

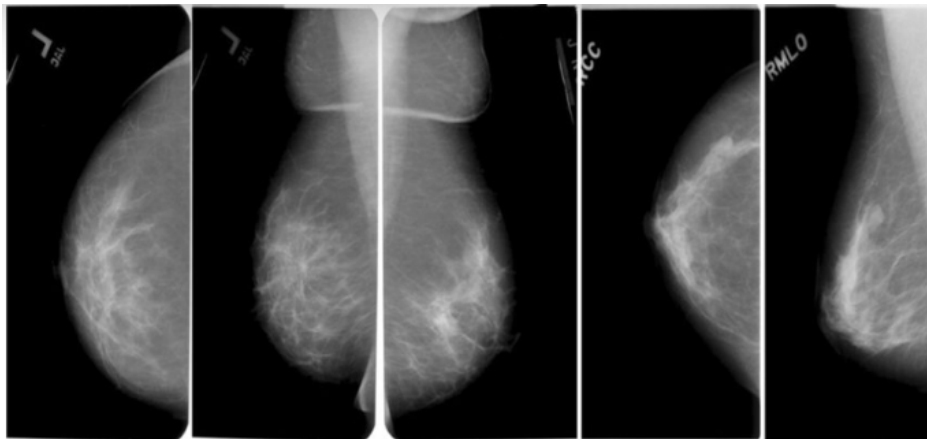


Figure 3: Mammography Images from DDSM dataset

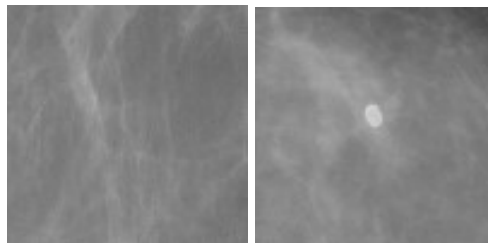


Figure 4: ROI Images from DDSM dataset (Normal, Cancer)

Like I was mentioned dataset stored as tfrecords so I convert the records images in png format to test model in my web app. I create 5 different models for this dataset.

Breast Ultrasound Images Dataset

This data collected from <https://www.kaggle.com/datasets/aryashah2k/breast-ultrasound-images-dataset> (kaggle). The data reviews the medical images of breast cancer using ultrasound scan. On the link you can see images like below:

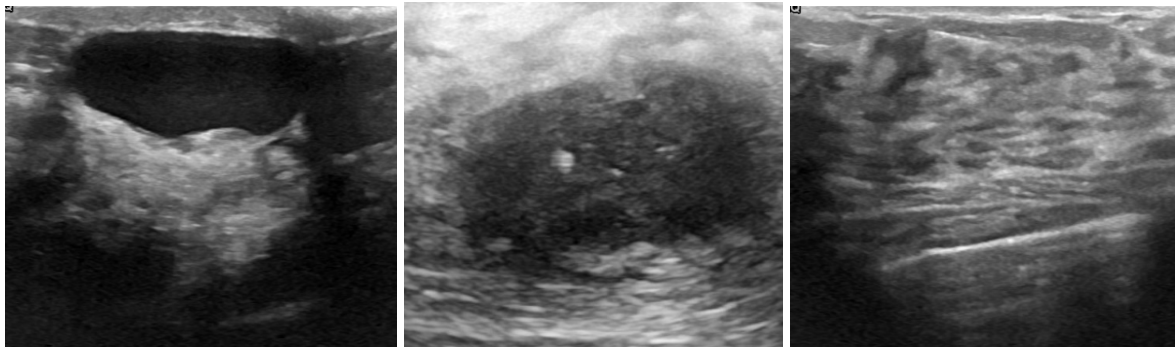


Figure 5,6 and 7: Benign tumor, Malignant tumor and Normal breast

This link contain ultrasound images of breast that already clustered. Another method of breast cancer screening is ultrasound imaging, which often uses a low dose of frequencies to create images of the breast, but keeps the contrast image very small. Ultrasound can detect and identify breast mass nodes and is mostly used for ease, volume, non-invasiveness, and low-cost [7]. We can't see patients' information but we know which one has tumor or not. And files are in png format. I created 4 different models for this dataset.

Breast Cancer Wisconsin (Diagnostic) DataSet

This data collected from <https://www.kaggle.com/datasets/paultimothymooney/breast-histopathology-images>.

On the link you can see images like below:

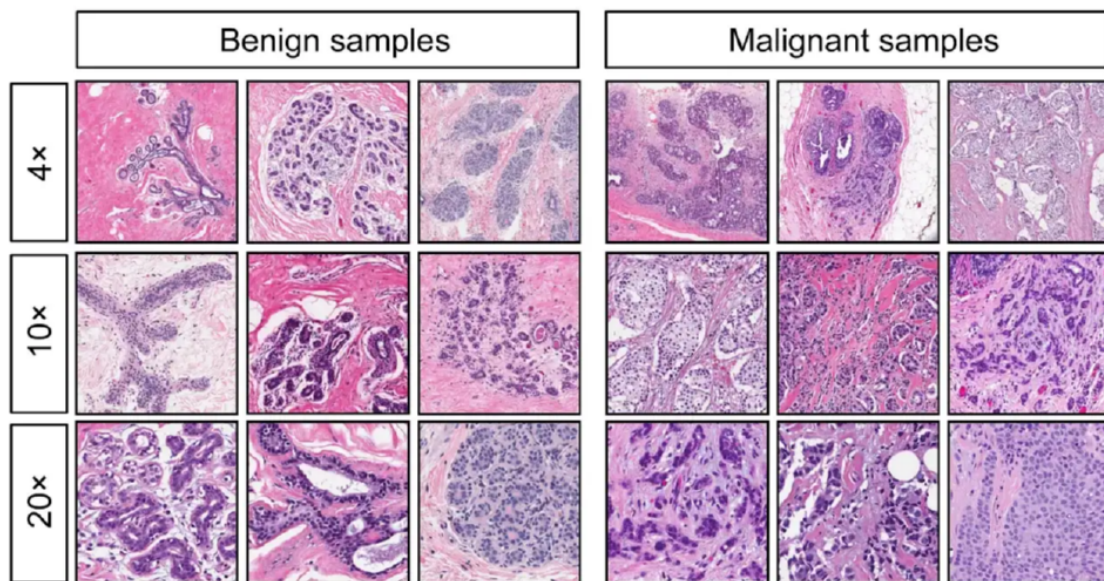


Photo 8: Wisconsin Breast Cancer DataSet Images

This dataset contains 198,738 IDC(-) image patches; 78,786 IDC(+) image patches. I create five different models for this dataset.

- **Techniques**

Deep Learning Classification Algorithm

Image processing requires deep learning methods that use data to train neural network algorithms to do various machine learning tasks.

Convolutional neural networks (CNNs) are particularly powerful neural networks which you'll use to classify different types of objects for the analysis of images.

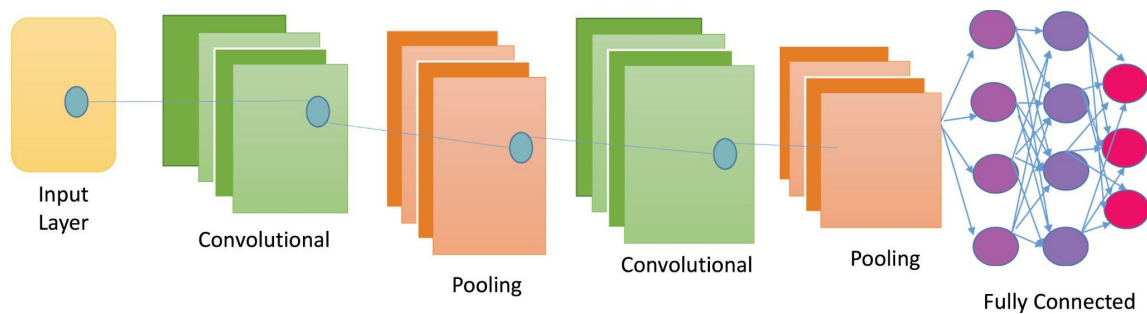
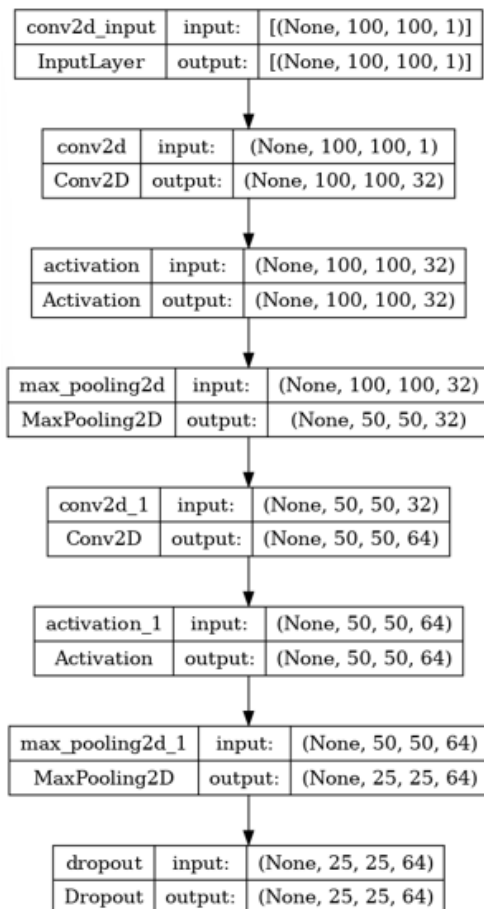


Figure 9: Basic CNN Architecture

A regular neural network has an input layer, hidden layers, and an output layer. The input layer essentially accepts a vast variety of different inputs, while the hidden layers perform calculations based on the inputs, and finally, the output layer will deliver the outcome of the calculations. A regular neural network contains neurons that are connected to neurons in the previous layer, each neuron having its own specific weight. This means there are no assumptions about the data being inputted into the network. I used CNN deep learning models mostly in my project.

Explaining of components that I used in my project in CNN architecture:

The figure you are seeing below represents a part of the model I use to classify mammography images.



Convolutional Neural Networks (CNNs) are particularly well-suited for image classification tasks due to their ability to effectively capture spatial features in images. CNNs leverage the convolutional layers to automatically learn and extract hierarchical representations from the image data. These layers utilize shared weights and local receptive fields, allowing the network to detect patterns and features at different scales and orientations, enabling robust and discriminative feature learning. Additionally, CNNs often incorporate pooling layers for downsampling and non-linear activation functions to introduce non-linearities, making them highly effective in handling the high-dimensional input of images and achieving state-of-the-art performance in image classification tasks. So decided to continue with cnn architecture for my project.

We can see (100,100,1) is the input shape of every picture in dataset (height, width, channel). Channel 1 means the input data is grayscale image.

The Conv2D layer is a fundamental building block in convolutional neural networks (CNNs) used for image processing tasks. It performs convolution operations on the input data, applying a set of filters to extract meaningful features. These filters slide across the input image, computing dot products to generate feature maps that capture different patterns and

Figure 10: My CNN Architecture representations within the image.

Activation function which is Relu in my model, **The Rectified Linear Unit (ReLU)** is commonly used in neural networks. It introduces non-linearity by setting negative values in the input to zero, while preserving positive values unchanged. ReLU helps the network learn complex representations by enabling the model to capture and amplify important features, improving its ability to generalize and make accurate predictions. It is computationally efficient and helps alleviate the vanishing gradient problem.

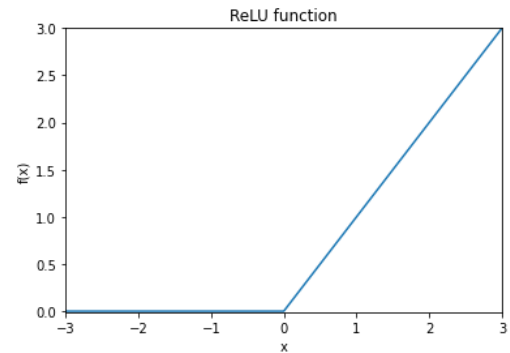


Figure 11: Relu function

MaxPooling2D is a popular operation used in convolutional neural networks for downsampling feature maps. It divides the input into non-overlapping rectangular regions and takes the maximum value within each region, discarding the rest. This helps to reduce the spatial dimensions of the input, extract the most salient features, and improve the computational efficiency of subsequent layers.

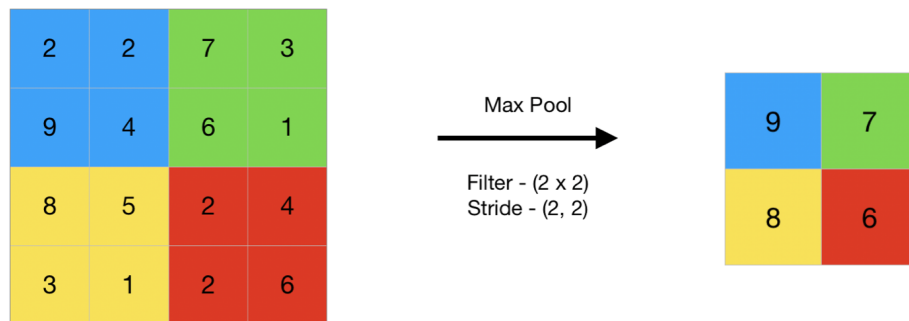
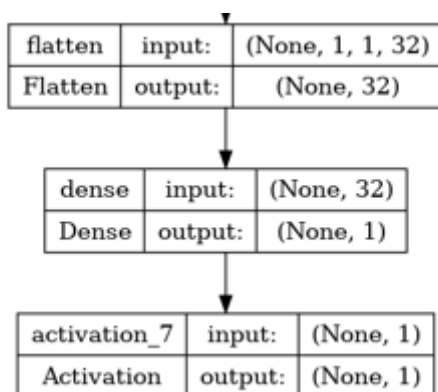


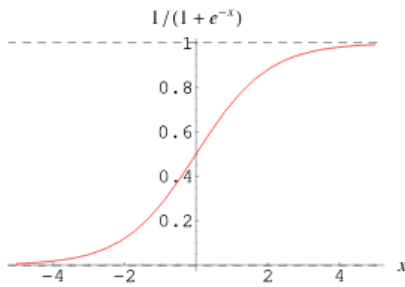
Figure 12: Max Pooling Process

Dropout is a regularization technique commonly used in neural networks to prevent overfitting. During training, it randomly sets a fraction of the input units to zero at each update, which helps to prevent the network from relying too heavily on any specific feature. This encourages the network to learn more robust and generalized representations, leading to improved performance on unseen data.



In the context of neural networks, a **dense layer**, also known as a fully connected layer, is a type of layer where each neuron is connected to every neuron in the previous layer. It performs a linear transformation on the input data followed by a non-linear activation function, allowing the network to learn complex relationships between the input and output.

Figure 13: Last layers of my CNN architecture



Activation function for last layer: The **sigmoid function** is a commonly used activation function in neural networks that maps the input to a value between 0 and 1. It has an S-shaped curve and is useful for models that need to produce binary classifications or probabilities.

Figure 14: Sigmoid Function

Machine Learning Classification Algorithm

I decided to try **Support Vector Machine** on Ultrasound Dataset Training process because SVM (Support Vector Machine) can be a good choice for ultrasound image classification with a small dataset due to its ability to handle high-dimensional data efficiently and its robustness against overfitting. SVM works well with small datasets as it focuses on finding the optimal hyperplane that maximally separates different classes, even with limited samples. Additionally, SVM has a solid theoretical foundation and performs well when the number of features is larger than the number of samples, making it suitable for ultrasound images, which typically have high dimensionality.

In machine learning, support vector machines are supervised models. A support vector machine creates a hyperplane when classifying the objects. A hyperplane is a line on a plane that distinguishes the two classes. Given a group of coaching examples, each marked as belonging to at least one or the opposite of two categories, an SVM training algorithm builds a model that assigns new examples to at least one category or the opposite, making it a non-probabilistic binary linear classifier (although methods like Platt scaling exist to use SVM during a probabilistic classification setting). New examples are then mapped into that very same space and predicted to belong to a category supported the side of the gap on which they fall.

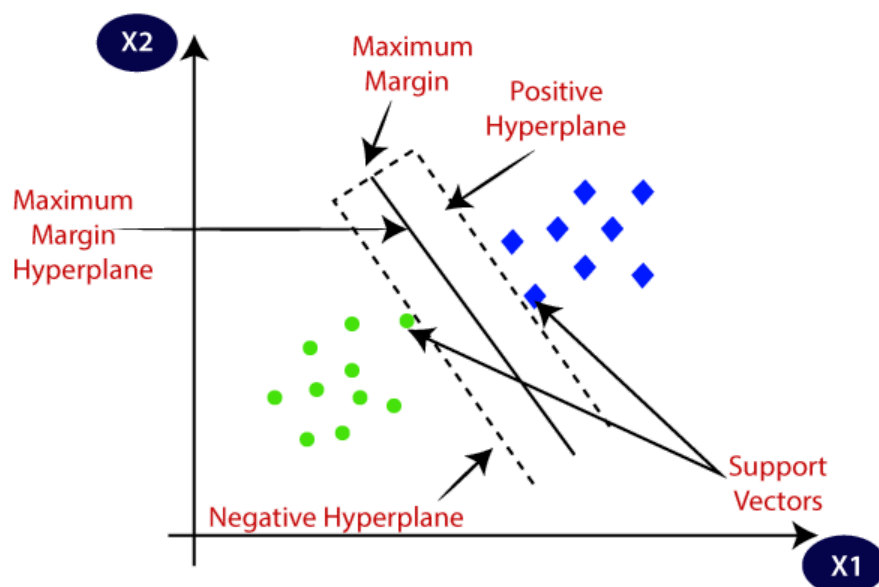


Figure 15: Support Vector Machine Training Algorithm

5 Technologies and Tools

Machine Learning Frameworks/Libraries That Used In Breast Cancer Prediction Project

- **Keras:** Keras provides a high-level and user-friendly API for building and training neural networks. It simplifies the process of constructing complex architectures, such as convolutional neural networks (CNNs), which are commonly used for image classification. Keras offers pre-built layers, optimizers, and loss functions specifically designed for deep learning tasks, making it easier to create and experiment with different models.
- **TensorFlow:** TensorFlow is a powerful machine learning library that serves as the backend for Keras. It provides efficient computation on GPUs and distributed systems, making it suitable for training large-scale image classification models. TensorFlow offers a wide range of operations and tools for tensor manipulation, model deployment, and performance optimization. It also supports automatic differentiation, enabling the training of deep learning models through gradient-based optimization.
- **OpenCV:** OpenCV is a widely used computer vision library that offers a diverse set of functions for image and video processing. It provides numerous algorithms for image manipulation, feature extraction, and preprocessing, which are crucial steps in image classification pipelines. OpenCV allows you to perform tasks like resizing, cropping, and augmenting image data, as well as extracting meaningful features for training machine learning models.
- **Flask:** Flask is a lightweight web framework written in Python that allows developers to build web applications quickly and efficiently. It provides a simple and flexible architecture for creating web APIs and handling HTTP requests and responses. I used Flask to create a web application for my project usage.

Breast Cancer Image Classification

Patient ID:

1111111111

Classification Type Of Image:


Ultrasound

Upload Your Image :

Dosya Seç

benign (2).png

Submit



Prediction For This Image: *Benign*

Figure 16: My Flask Web Application For Breast Cancer Image Classification

I used these tools for my project because by combining the capabilities of Keras, TensorFlow, and OpenCV, you can build an end-to-end image classification pipeline. You can use OpenCV for data preprocessing, feature extraction, and augmentation. Then, with Keras and TensorFlow, you can design and train deep learning models, leveraging their efficiency, flexibility, and advanced functionalities. Overall, these tools provide a comprehensive and powerful toolkit for developing accurate and efficient image classification systems.

Platforms

- **Kaggle** is an online community and platform for data science and machine learning competitions. It provides a vast repository of datasets and a collaborative environment for data scientists to explore, analyze, and build predictive models. By participating in Kaggle competitions, you can gain valuable experience, learn from other experts, and showcase your skills to potential employers, making it an ideal platform to enhance your data science expertise. Kaggle is used by data scientists and machine learning practitioners for several reasons. Firstly, it offers access to a wide variety of real-world datasets, enabling practitioners to work on diverse and challenging problems. Secondly, Kaggle provides a platform for knowledge sharing and collaboration, allowing participants to learn from each other's approaches and improve their skills. I used Kaggle for developing my notebooks and collecting datasets.
- **Colab**, short for Google Colaboratory, is a cloud-based development environment provided by Google. It allows users to write and execute Python code, particularly for data science and machine learning tasks, in a browser-based notebook interface. Colab offers free access to GPUs and TPUs, making it suitable for training and running computationally intensive models without the need for powerful hardware. Colab is widely used for data science and machine learning projects due to its several advantages. Firstly, it provides a convenient and collaborative environment for coding and experimentation, allowing multiple users to work together on the same notebook in real-time. Secondly, Colab integrates seamlessly with other Google services and libraries, such as Google Drive and TensorFlow, making it easy to import and export data and leverage additional tools and frameworks.
- **Firebase** is a versatile platform that simplifies backend development and provides essential services for building and deploying web and mobile applications. Its real-time database, authentication, and hosting services make it a popular choice among developers for building scalable and interactive applications. I used Firebase in the breast cancer prediction information storage part.

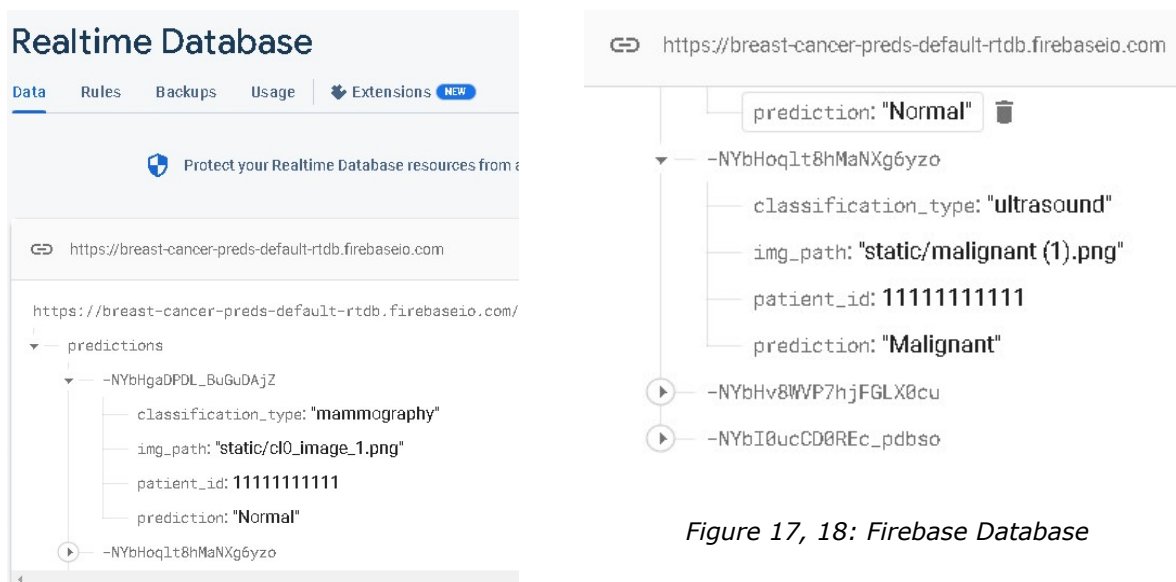


Figure 17, 18: Firebase Database

6 Project Implementation

Senior Desing Project Diagram

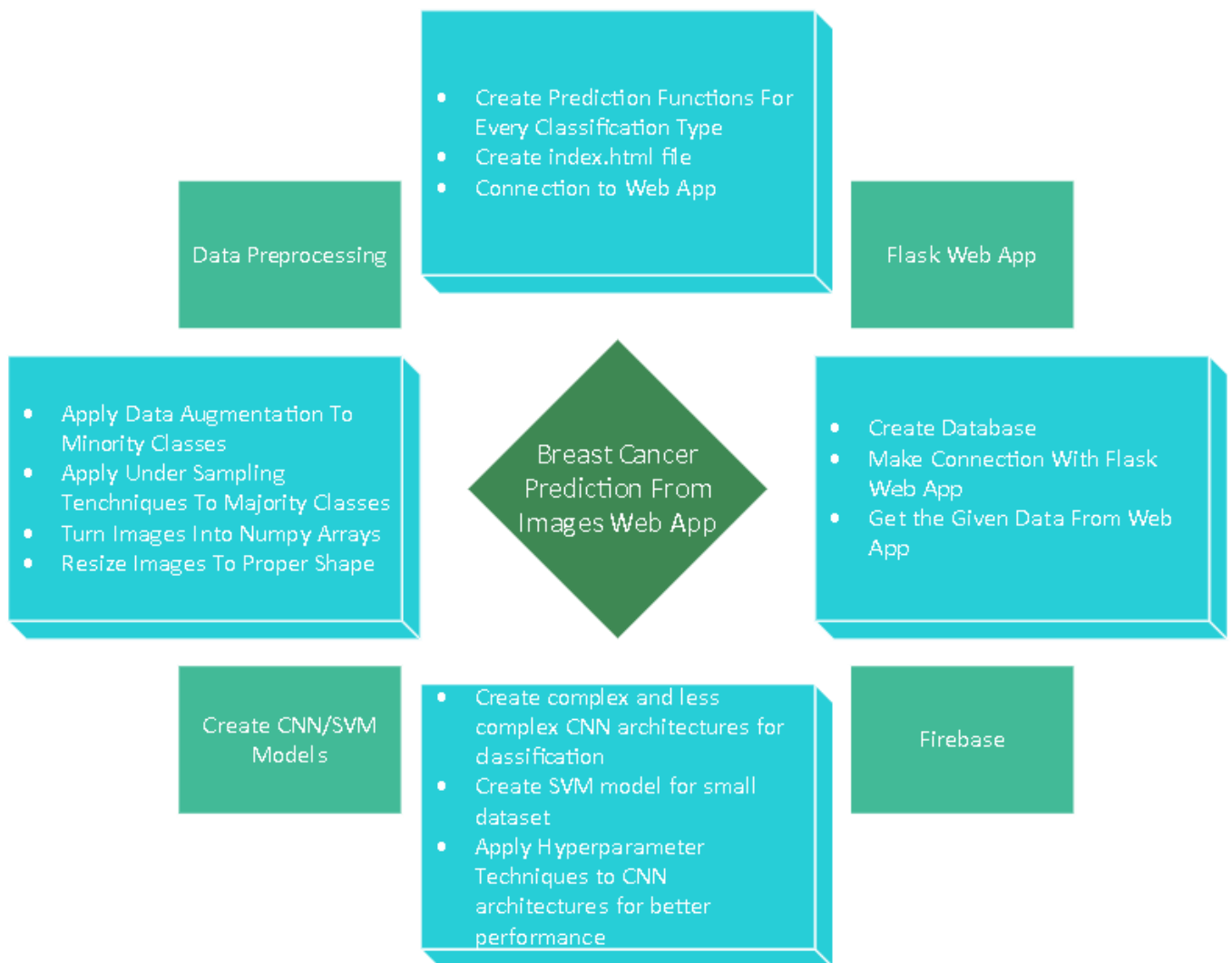


Figure 17: Breast Cancer Detection Web Application

The first step in my project flow is to collect the data, which involves gathering mammogram images and converting them into arrays and tensors. The collected data is then appropriately resized, normalized and split into training and testing datasets. To enhance the robustness of the models, data augmentation techniques and undersampling can be applied. Various models are trained using this dataset, and the best-performing models are identified for further use in the web application.

Moving on to the web application development, Flask is utilized to create an interface for users. The index HTML file contains input fields for the patient ID number, which is a unique 11-digit identifier, and a select tag allowing the user to choose the classification type (mammography, histopathology, or ultrasound). Additionally, the user can select an image file for prediction. Upon submitting the form, the Flask application renders the prediction based on the chosen model and displays it to the user. The prediction results are automatically saved to Firebase for future reference.

The saved predictions are stored as JSON files, with each file containing information such as the patient ID, image path, classification type, and the predicted result. This storage format ensures easy access to the predictions and facilitates further analysis or retrieval of specific information as needed.

By organizing the prediction data in a structured manner, it becomes possible to track and monitor the predictions made by the web application, enabling efficient management and analysis of the collected results.

7 Results Of Models

Breast Histopathology Images CNN Architecture Models							
Model Name	Model Description	Validation Accuracy	Validation Loss	Test Accuracy	Test Loss	F1 score (class0)	F1 score (class1)
model_or_1	*Original Data *Complex	0,8568	0,3570	0,8263	0,3961	0,8400	0,8500
model_1_nor	*Normalized Data *Complex	0,8838	0,2835	0,8702	0,3221	0,87	0,87
model_or_2	*Original Data *Less Complex	0,845	0,4378	0,8263	0,5102	0,82	0,83
model_2_nor	*Normalized Data *Less Complex	0,8373	0,4997	0,8209	0,4806	0,81	0,83
model_1_nor_ht	*Normalized Data *Complex *Hyperparameter Tuning	0,8528	0,3422	0,8402	0,368	-	-
Mammography ROI Images CNN Architecture Models							
model_1_mam	*Unbalanced Data *Complex	0,901	0,2906	0,9012	0,3029	0,93	0
model_2_mam	*Balanced Data *Complex	0,8061	0,4037	0,81	0,3851	0,7	0
model_3_mam	*Unbalanced Data *Less Complex	0,9345	0,234	0,9	0,68	0,93	0
model_4_mam	*Balanced Data *Less Complex	0,3055	0,9121	0,8451	0,595	0,7	0
model_5_mam	*Balanced Data *Learning rate schedule	0,8583 (lr:0.025)	0,3112	0,8457	0,4657	0,7	0
Ultrasound Images Models							
ult_model_1_a	*Augmented Data *CNN	0,7948	1,82	0,97	-	0,95	0,98
ult_model_1	*Original Data *CNN	0,7094	0,6996	0,75	-	0,47	0,85
ult_svm_model	*Original Data *SVM	-	-	0,67	-	0,14	0,72
ult_svm_model	*Augmented Data *SVM	-	-	0,69	-	0,2	0,78

We can see all finished models results in my project. Of course I try more models but I dont save all of them because of their problems.

Breast Histopathology Images CNN Architecture Models:

- **model_or_1:** Original data with a complex CNN architecture. Achieved a validation accuracy of 0.8568 and a validation loss of 0.3570. The test accuracy is 0.8263 with a test loss of 0.3961. It shows a high F1 score for both classes (0.84 for class 0 and 0.85 for class 1).
- **model_1_nor (Chosen Model For Web App):** Normalized data with a complex CNN architecture. It achieved a higher validation accuracy of 0.8838 and a lower validation loss of 0.2835. The test accuracy is 0.8702 with a test loss of 0.3221. It maintains a consistent F1 score of 0.87 for both classes.
- **model_or_2:** Original data with a less complex CNN architecture. It achieved a validation accuracy of 0.845 and a validation loss of 0.4378. The test accuracy is 0.8263 with a test loss of 0.5102. The F1 score is 0.82 for class 0 and 0.83 for class 1.
- **model_2_nor:** Normalized data with a less complex CNN architecture. It achieved a validation accuracy of 0.8373 and a validation loss of 0.4997. The test accuracy is 0.8209 with a test loss of 0.4806. It shows a consistent F1 score of 0.81 for class 0 and 0.83 for class 1.
- **model_1_nor_ht:** Normalized data with a complex CNN architecture and hyperparameter tuning. It achieved a validation accuracy of 0.8528 and a validation loss of 0.3422. The test accuracy is 0.8402 with a test loss of 0.368. The F1 scores are not provided for this model.

Mammography ROI Images CNN Architecture Models:

- **model_1_mam:** Unbalanced data with a complex CNN architecture. Achieved a validation accuracy of 0.901 and a validation loss of 0.2906. The test accuracy is 0.9012 with a test loss of 0.3029. It shows a high F1 score of 0.93 for class 0 and 0 for class 1.
- **model_2_mam(Chosen Model For Web app):** Balanced data with a complex CNN architecture. It achieved a validation accuracy of 0.8061 and a validation loss of 0.4037. The test accuracy is 0.81 with a test loss of 0.3851. The F1 scores are 0.7 for class 0 and 0 for class 1.

- **model_3_mam:** Unbalanced data with a less complex CNN architecture. It achieved a higher validation accuracy of 0.9345 and a lower validation loss of 0.234. The test accuracy is 0.9 with a test loss of 0.68. The F1 scores are 0.93 for class 0 and 0 for class 1.
- **model_4_mam:** Balanced data with a less complex CNN architecture. It achieved a lower validation accuracy of 0.3055 and a higher validation loss of 0.9121. The test accuracy is 0.8451 with a test loss of 0.595. The F1 scores are 0.7 for class 0 and 0 for class 1.
- **model_5_mam:** Balanced data with a learning rate schedule. Achieved a validation accuracy of 0.

Ultrasound Images CNN and SVM Models:

- **ult_model_1_aug (Chosen Model For Web App) :** This model used augmented data with a CNN architecture. It achieved a validation accuracy of 0.7948 and a validation loss of 1.82. The test accuracy is 0.97, and the F1 score is 0.95 for class 0, 0.98 for class 1, and 0.95 for class 2. However, the test loss value is not provided.
- **ult_model_1:** This model used original data with a CNN architecture. It achieved a lower validation accuracy of 0.7094 and a validation loss of 0.6996. The test accuracy is 0.75. The F1 score is 0.47 for class 0, 0.85 for class 1, and 0.66 for class 2. The test loss value is not provided.
- **ult_svm_model_1:** This model used original data with an SVM classifier. The validation accuracy and validation loss values are not provided. However, it achieved a test accuracy of 0.67. The F1 score is 0.14 for class 0, 0.72 for class 1, and 0.61 for class 2.
- **ult_svm_model_2:** This model used augmented data with an SVM classifier. The validation accuracy and validation loss values are not provided. It achieved a test accuracy of 0.69. The F1 score is 0.2 for class 0, 0.78 for class 1, and 0.65 for class 2.

8 Future Work

In future work, there are several areas of development that can be explored to enhance the project. Firstly, expanding the database can provide a larger and more diverse set of data for training and evaluation, leading to improved model performance and generalization. Additionally, further development and refinement of the models can be carried out, experimenting with different architectures, hyperparameters, and techniques like Transfer Learning to leverage pre-trained models and boost performance.

Moreover, creating a user web page in Flask can greatly enhance the project's usability and accessibility. This web page can serve as an interface for users to interact with the system, allowing them to input queries, retrieve information from the database, and view model predictions or analysis results. Flask's flexibility and extensibility make it an ideal choice for building a user-friendly web page that seamlessly integrates with the existing functionality.

By incorporating these future work aspects, the project can be taken to the next level, enabling scalability, improved model performance, and enhanced user experience. It opens up opportunities for further research, experimentation, and practical deployment, ultimately leading to a more robust and valuable solution.

9 References

- 1 Rane, Nikita, et al. "Breast cancer classification and prediction using machine learning." International Journal of Engineering Research and Technology 9.2 (2020): 576-580.
- 2 Agarap, Abien Fred M. "On breast cancer detection: an application of machine learning algorithms on the wisconsin diagnostic dataset." Proceedings of the 2nd international conference on machine learning and soft computing. 2018.

- 3 Naji, Mohammed Amine, et al. "Machine learning algorithms for breast cancer prediction and diagnosis." *Procedia Computer Science* 191 (2021): 487-492.
- 4 JASTI, V., et al. Computational technique based on machine learning and image processing for medical image analysis of breast cancer diagnosis. *Security and Communication Networks*, 2022, 2022.
- 5 Ragab, Dina A., et al. "A framework for breast cancer classification using multi-DCNNs." *Computers in Biology and Medicine* 131 (2021): 104245.
- 6 Alzubaidi, Laith, et al. "Optimizing the performance of breast cancer classification by employing the same domain transfer learning from hybrid deep convolutional neural network model." *Electronics* 9.3 (2020): 445.
- 7 Houssein, Essam H., et al. "Deep and machine learning techniques for medical imaging-based breast cancer: A comprehensive review." *Expert Systems with Applications* 167 (2021): 114161.
- 8
"https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33"
- 9 "https://www.edureka.co/blog/keras-vs-tensorflow-vs-pytorch/"
- 10
"https://towardsdatascience.com/which-is-better-for-your-machine-learning-task-opencv-or-tensorflow-ed16403c5799"
- 11
"https://www.earthdatascience.org/courses/intro-to-earth-data-science/python-code-fundamentals/use-python-packages/introduction-to-python-conda-environments/"
-

Figures from:

<https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm>

<https://www.xenonstack.com/blog/artificial-neural-network-applications>