**CancerVision: Advanced Breast Cancer Prediction With Deep Learning**

Breast cancer is one of the main causes of cancer death worldwide. Computer-aided diagnosis systems showed potential for improving the diagnostic accuracy. But early detection and prevention can significantly reduce the chances of death. It is important to detect breast cancer as early as possible. The goal is to classify images into two classifications of malignant and benign. As early diagnostics significantly increases the chances of correct treatment and survival. In this application we are helping the doctors and patients to classify the Type of Tumour for the specific image given with the help of Neural Networks.

**Anaconda Navigator :**

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning-related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross-platform,  package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook,

QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using a Jupyter notebook and Spyder

To install the Anaconda navigator and to know how to use Jupyter Notebook & Spyder using Anaconda watch the video

[link](https://www.youtube.com/watch?v=5mDYijMfSzs&feature=emb_logo)

**Python packages:**

**NumPy:** NumPy is a Python package that stands for 'Numerical Python. It is the core library for scientific computing, which contains a powerful n-dimensional array of objects.

**Pandas:** pandas is a fast, powerful, flexible, and easy-to-use open-source data analysis and manipulation tool, built on top of the Python programming language[.](https://pandas.pydata.org/)

**Matplotlib:**It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits

**Keras:**Keras is an open-source library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Up until version 2.3, Keras supported multiple backends, including TensorFlow, Microsoft Cognitive Toolkit, R, Theano, and PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.

**TensorFlow:** TensorFlow is just one part of a much bigger, and growing ecosystem of libraries and extensions that help you accomplish your machine learning goals. It is a free and open-source software library for data flow and differentiable programming across a range of tasks. It is a symbolic math library and is also used for machine learning applications such as neural networks.

**Flask:** Web framework used for building Web applications

Watch the video to install packages:

* Python packages:
  + open anaconda prompt as administrator
  + Type “pip install numpy” and click enter.
  + Type “pip install pandas” and click enter.
  + Type “pip install scikit-learn” and click enter.
  + Type “pip install tensorflow==2.3.2” and click enter.
  + Type “pip install keras==2.3.1” and click enter.
  + Type “pip install Flask” and click enter.

* **Deep Learning Concepts**

* **CNN**: a convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery.

[CNN Basic](https://towardsdatascience.com/basics-of-the-classic-cnn-a3dce1225add)

* **Flask**: Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications.

[**Flask Basics**](https://www.youtube.com/watch?v=lj4I_CvBnt0)

If you are using Pycharm IDE, you can install the packages through the command prompt and follow the same syntax as above.

### Prior Knowledge

You must have prior knowledge of following topics to complete this project.

* Deep Learning Concept
  + [CNN](https://towardsdatascience.com/basics-of-the-classic-cnn-a3dce1225add)

* Flask: Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications.
* **Neural Networks:**

**Link :**[**click**](https://www.youtube.com/watch?v=vpOLiDyhNUA)

### Project Objectives

By the end of this project you will:

* Know fundamental concepts and techniques of Convolutional Neural Network.
* Gain a broad understanding of image data.
* Know how to pre-process/clean the data using different data preprocessing techniques.
* know how to build a web application using the Flask framework.

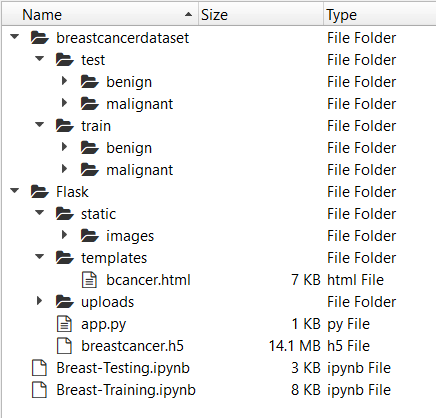
### Project Flow

* The user interacts with the UI (User Interface) to choose the image.
* The chosen image analyzed by the model which is integrated with flask application.
* CNN Models analyze the image, then prediction is showcased on the Flask UI.

To accomplish this, we have to complete all the activities and tasks listed below

* Data Collection.
  + Create Train and Test Folders.
* Data Preprocessing.
* Import the ImageDataGenerator library
* Configure ImageDataGenerator class
* ApplyImageDataGenerator functionality to Trainset and Testset
* Model Building
  + Import the model building Libraries
  + Initializing the model
  + Adding Input Layer
  + Adding Hidden Layer
  + Adding Output Layer
  + Configure the Learning Process
  + Training and testing the model
  + Save the Model
* Application Building
  + Create an HTML file
  + Build Python Code

**Project Structure**

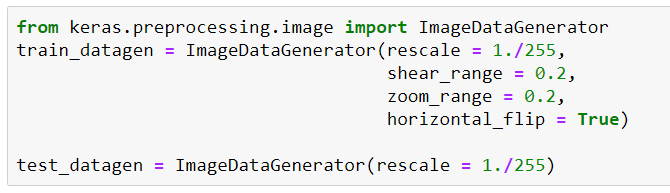


* The dataset folder (breastcancerdataset) contains two folders test and train, each of them having benign and malignant tumor images.
* Flask folder has all the files necessary to build the flask application.
* static folder has the images that are needed in building the web page.
* templates folder has the HTML page.
* uploads folder has the uploads made by the user.
* app.py is the python script for server side computing.
* .h5 file is the model file which is to be saved after model building.
* Breast-Training.ipynb and Breast-Testing.ipynb are the training and testing notebooks.
* **Data Collection**
* There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc
* **Download The Dataset**
* The dataset used for this project is in this [link](https://www.kaggle.com/paultimothymooney/breast-histopathology-images).  Please refer to the link to download the dataset. In this dataset, there are two classes of images people having breast cancer( represented with folder 1)  people not having a Breast cancer(represented with 0).In our project according to project structure, create train test folders and in them, place "0" folder images in benign and place "1" folder images in Malignant in train and test folders respectively as shown in project structure.
* 
* 198,738 IDC(-) image patches; 78,786 IDC(+) image patches..
* 198,738 IDC(-) image patches; 78,786 IDC(+) image patches..
* <https://www.kaggle.com/datasets/paultimothymooney/breast-histopathology-images>
* **Image Preprocessing**
* In this milestone we will be improving the image data that suppresses unwilling distortions or enhances some image features important for further processing, although perform some geometric transformations of images like rotation, scaling, translation, etc.

### Import ImageDataGenerator Library And Configure It

##### Activity 1: Import ImageDataGenerator Library and Configure it

ImageDataGenerator class is used to load the images with different modifications like considering the zoomed image, flipping the image and rescaling the images to range of 0 and 1.



### Apply ImageDataGenerator Functionality To Train And Test Set

Specify the path of both the folders in flow\_from\_directory method.

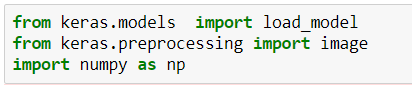
### Model Building

The neural network model is to be built by adding different network layers like convolution, pooling, flattening, dropout and neural layers.

### Test The Model

The model is to be tested with different images to know if it is working correctly.

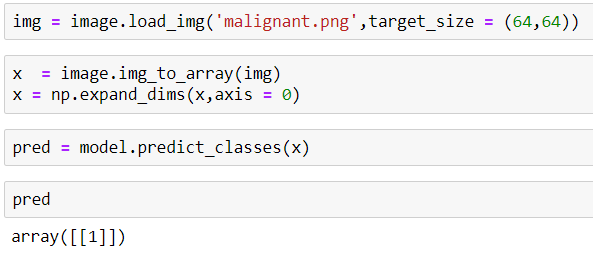
### Import The Packages And Load The Saved Model





### Load The Test Image, Pre-Process It And Predict

Pre-processing the image includes converting the image to array and resizing according to the model. Give the pre-processed image to the model to know to which class your model belongs to.



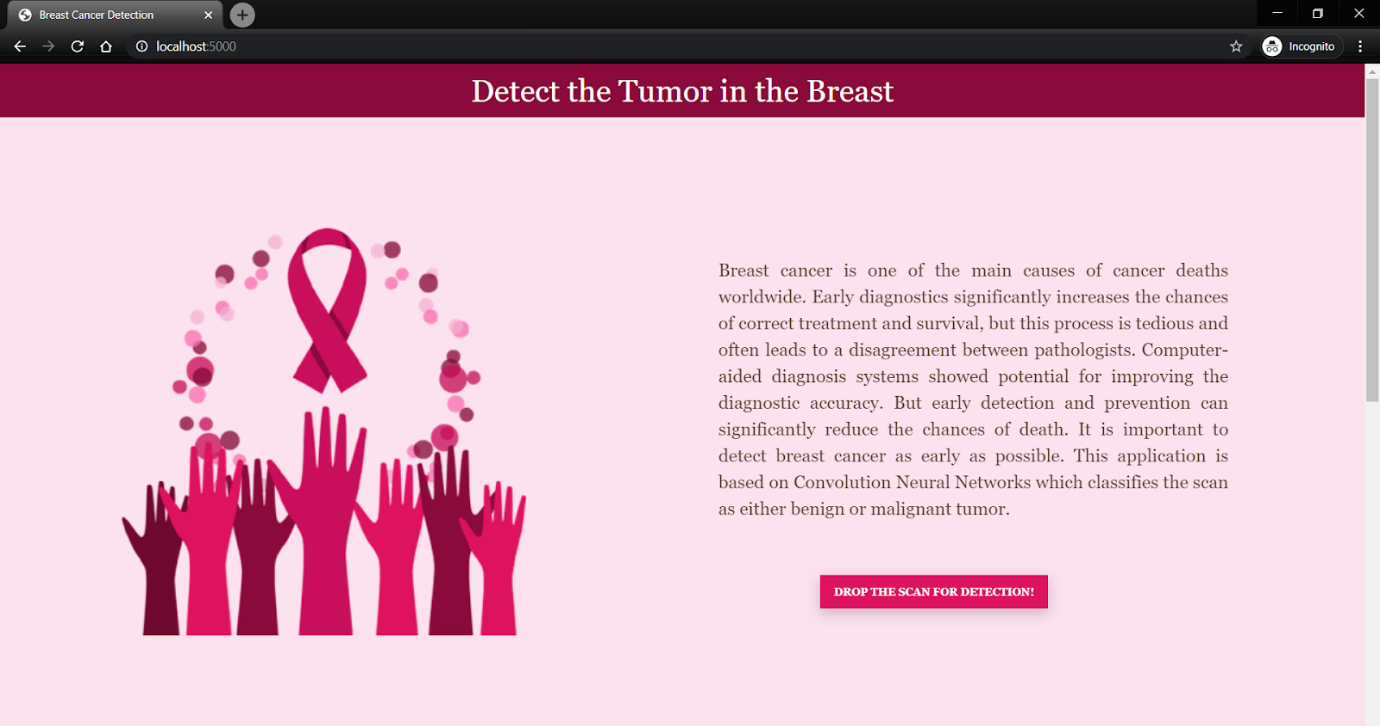
### Application Building

After the model is built, we will be integrating it to a web application so that normal users can also use it. The users need to give the microscopic image of the tissue/tumor to know the predictions.

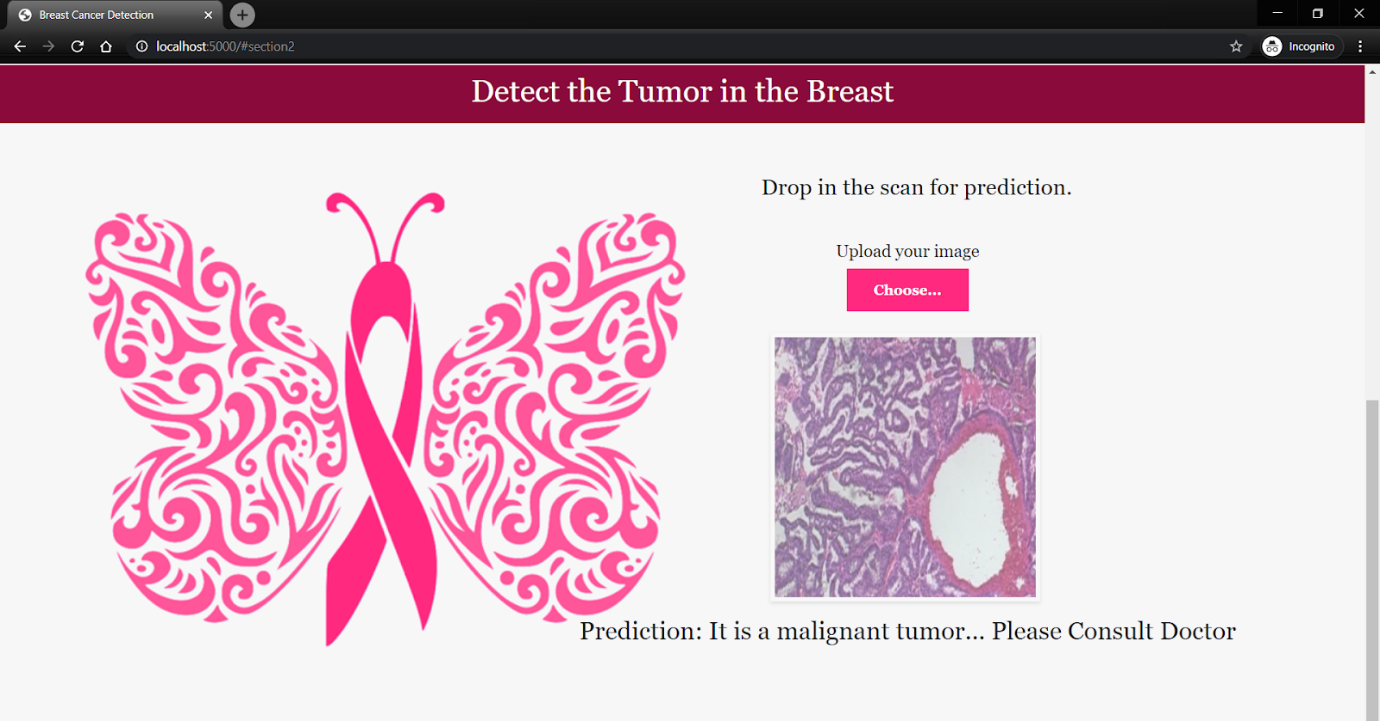
### Building Html Pages

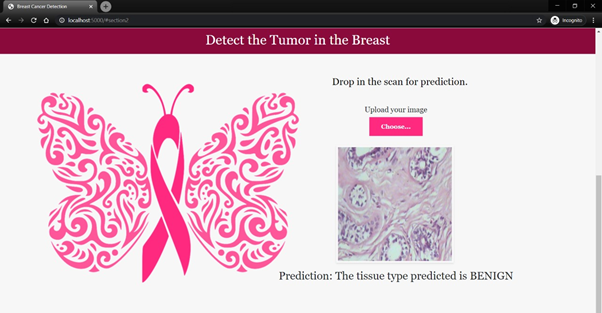
* Intro.html displays an introduction about the project
* upload.html gives the emergency alert
* We also use JavaScript-main.js and CSS-main.css to enhance our functionality and view of HTML page
* Link :[CSS](https://www.w3schools.com/css/) , [JS](https://www.w3schools.com/js/DEFAULT.asp)

The home page looks like this. When you click on the button “Drop the scan”, you’ll be redirected to the predict section.



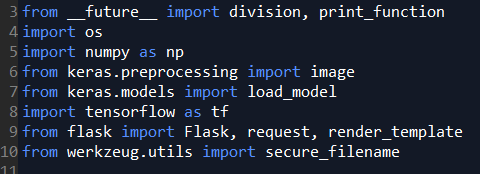
In this section you can browse and choose the image you want to predict and then click on predict to get the predictions.





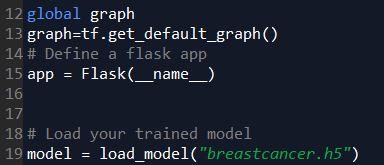
### Build Python Code

###### Step 1: Load the required packages

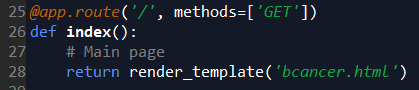


###### Step 2: Initialize graph, load the model, initialize the flask app and load the model

Graph elements are required to work with tensorflow. So, graph elements are created explicitly. Instance of Flask is created and the model is loaded using load\_model from keras.

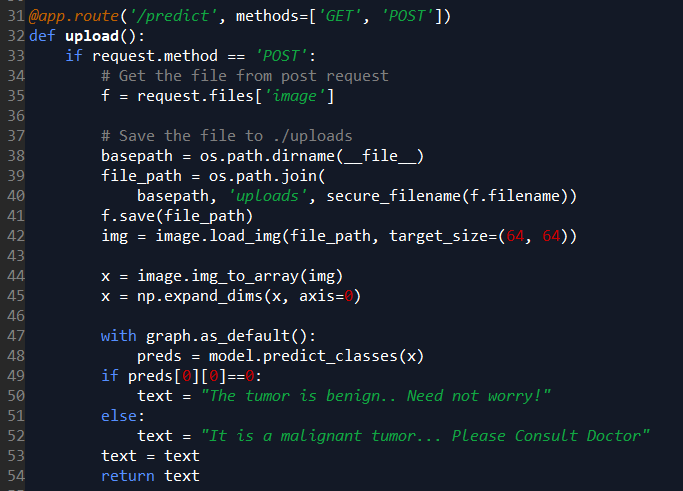


###### Step 3: Configure the home page



###### Step 4: Pre-process the frame and run

Pre-process the captured frame and give it to the model for prediction. Based on the prediction the output text is generated and sent to the HTML to display.



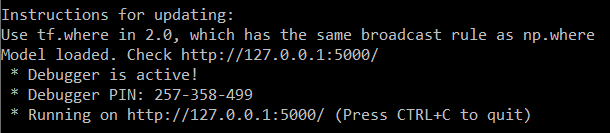
Run the flask application using the run method. By default the flask runs on 5000 port. If the port is to be changed, an argument can be passed and port can be modified.



### Run The Application

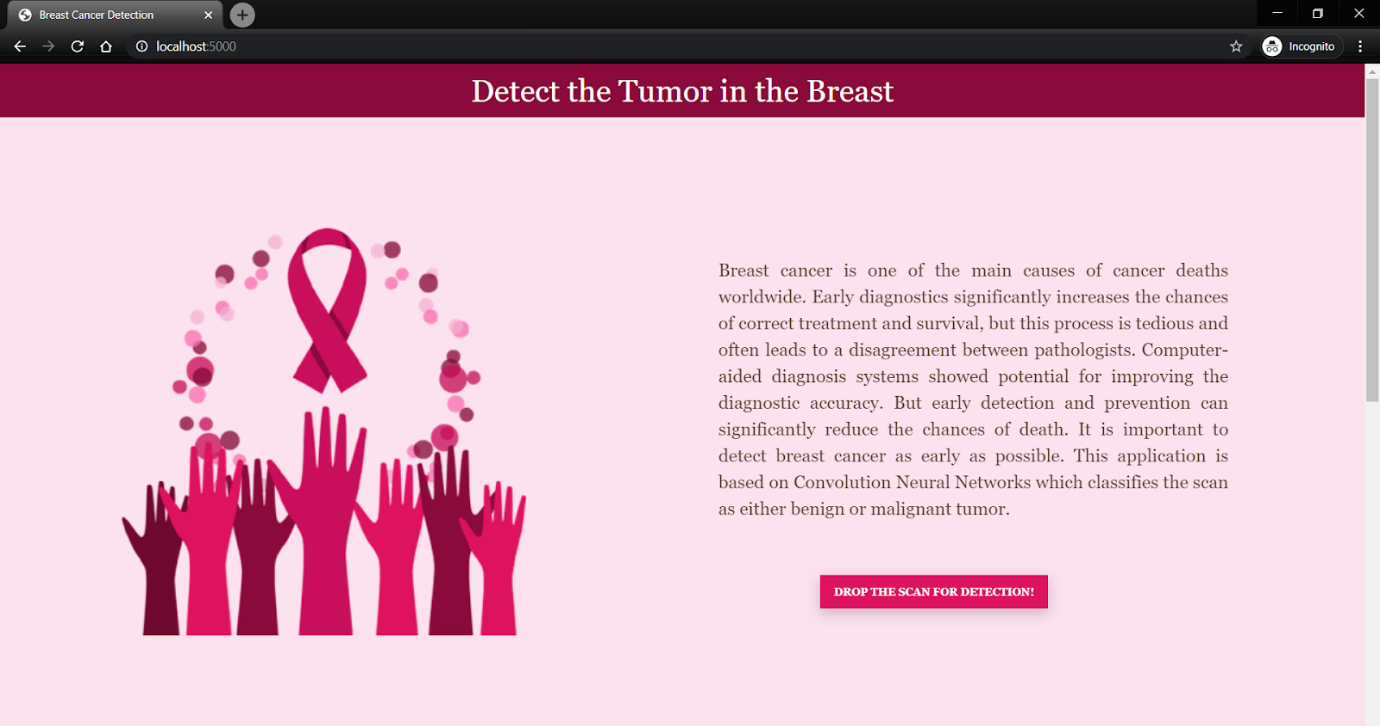
In anaconda prompt, navigate to the folder in which the flask app is present. When the python file is executed the localhost is activated on 5000 port and can be accessed through it.





###### **Open the browser and navigate to localhost:5000 to check your application**

The home page looks like this. When you click on the button “Drop the scan”, you’ll be redirected to the predict section.



In this section you can browse and choose the image you want to predict and then click on predict to get the predictions.

