**PROJECT**

**PROJECT DEFINITION:**

Build a breast cancer classifier on an IDC dataset that can accurately **classify** a histology image as benign or malignant using DENSENET.

**INTRODUCTION:**

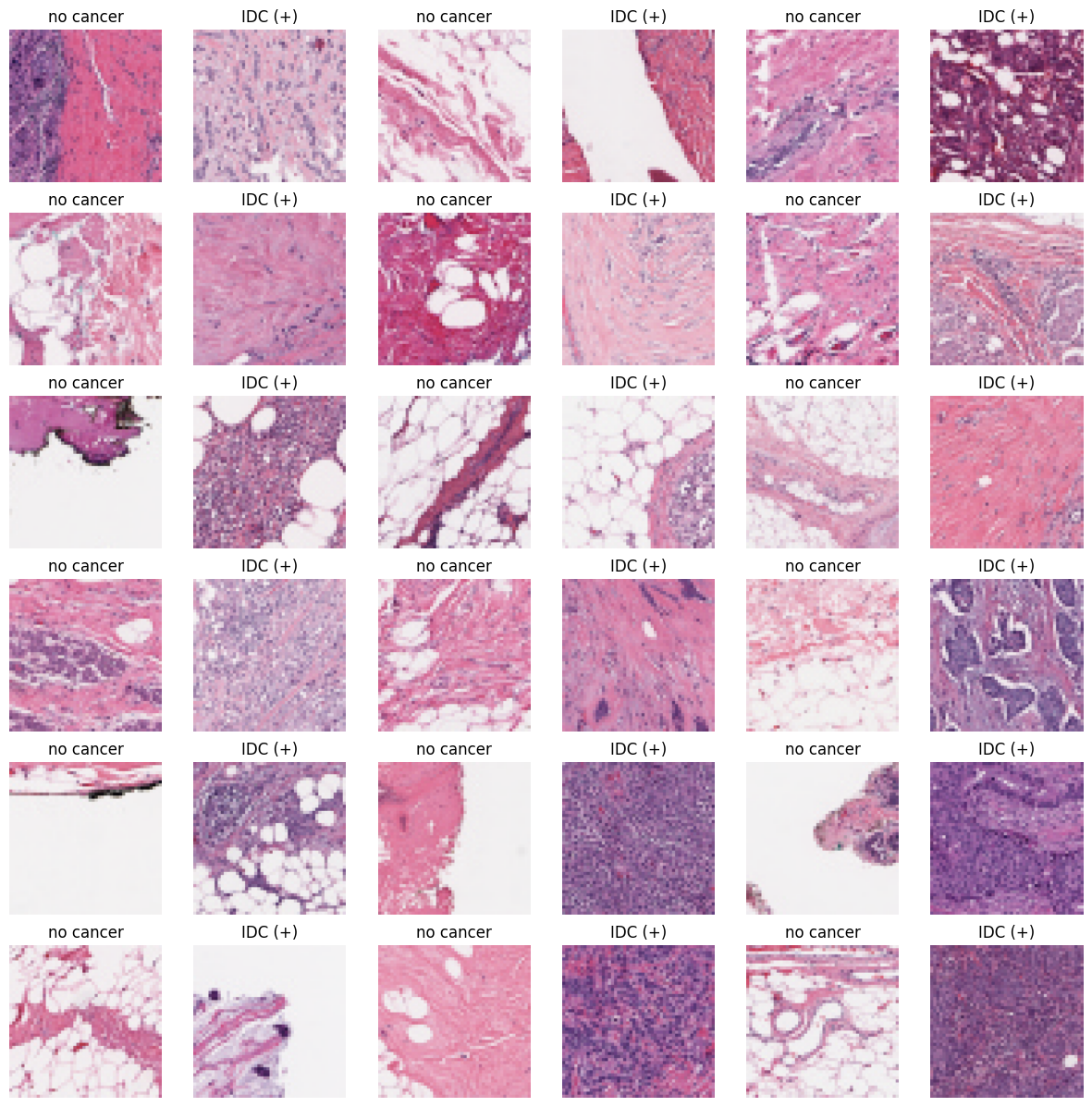
* Histopathology is the diagnosis and study of diseases of the tissues. They can reach a diagnosis by examining a small piece of tissue from the skin, liver, kidney or other organ. This is called a biopsy.
* Automatic analysis of histopathology images can help pathologists diagnose tumor and cancer subtypes, alleviating the workload of pathologists.
* IDC is Invasive Ductal Carcinoma; cancer that develops in a milk duct and invades the fibrous or fatty breast tissue outside the duct.
* It is the most common form of breast cancer forming 80% of all breast cancer diagnoses.

**DATASET DESCRIPTION:**

The property of extremely large size for a single image also makes a histopathology image dataset be considered large-scale, even if the number of images in the dataset is limited.

IDC\_regular dataset (the breast cancer histology image dataset) holds 9191 patches of size 50×50 extracted from 162 whole mount slide images of breast cancer specimens scanned at 40x.

Of these, 4769 test negative and 4422 test positive with IDC.

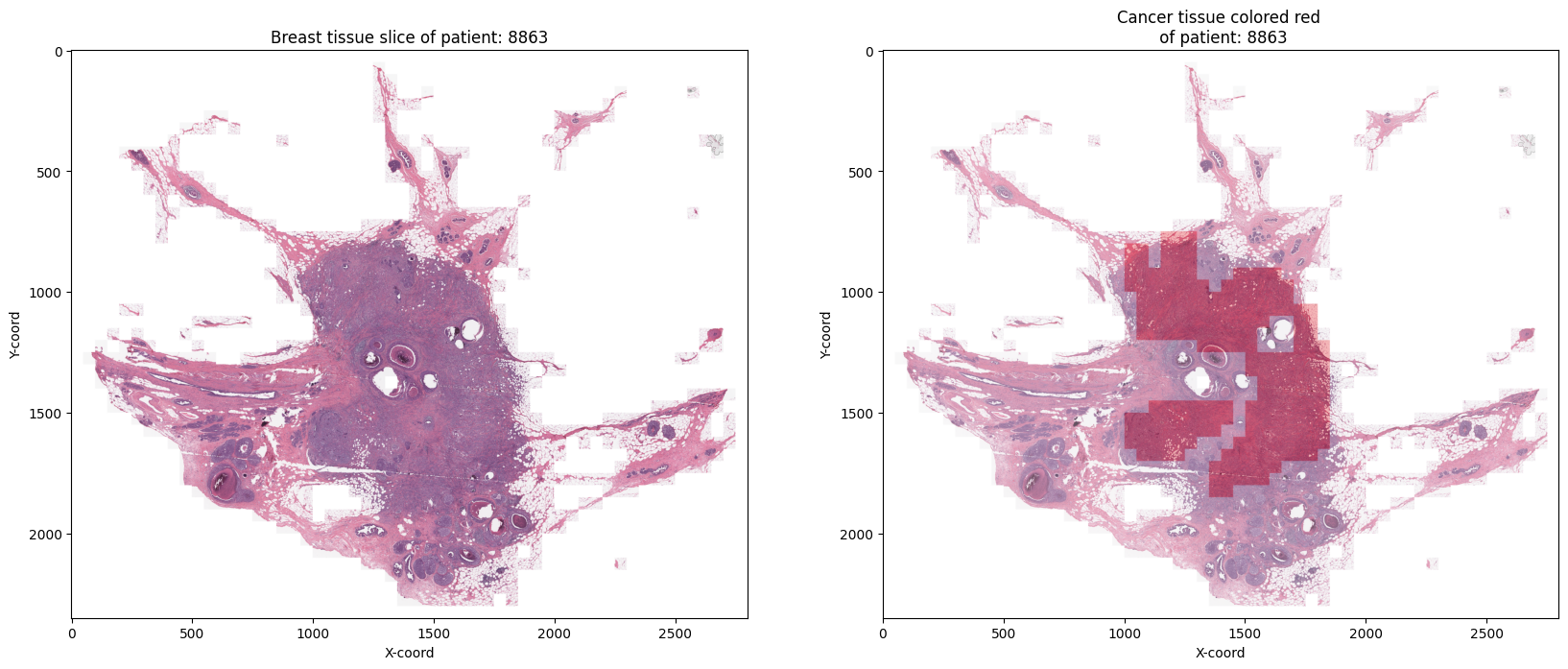


**DATA PREPROCESSING:**

Visualize the cancer tissue separately from the given histopathology image.

Preprocessing such as Image resizing,Image normalization, and Data augmentation techniques applied to the images to increase the diversity of the training data and improve the generalization of the model.

Tissue segmentation The images are segmented to separate the regions of interest (ROIs) from the background.



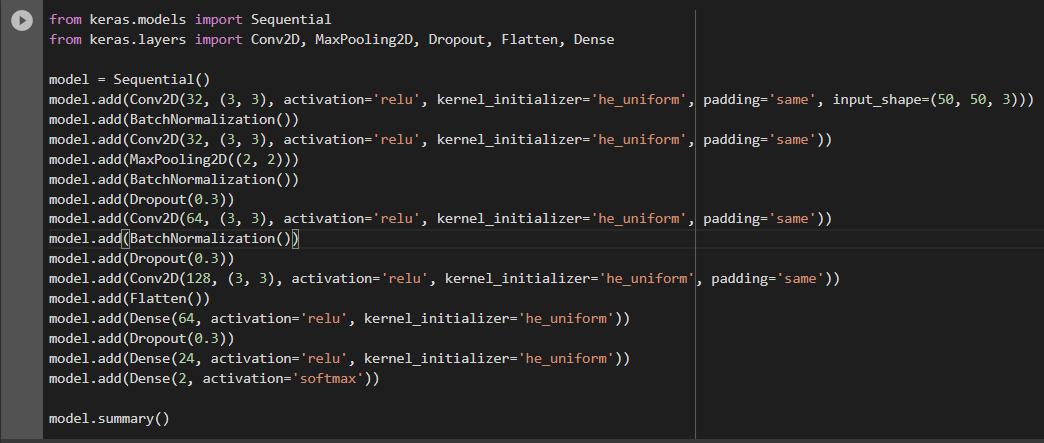
By applying these preprocessing steps, the histopathology images can be prepared for classification using DenseNet.

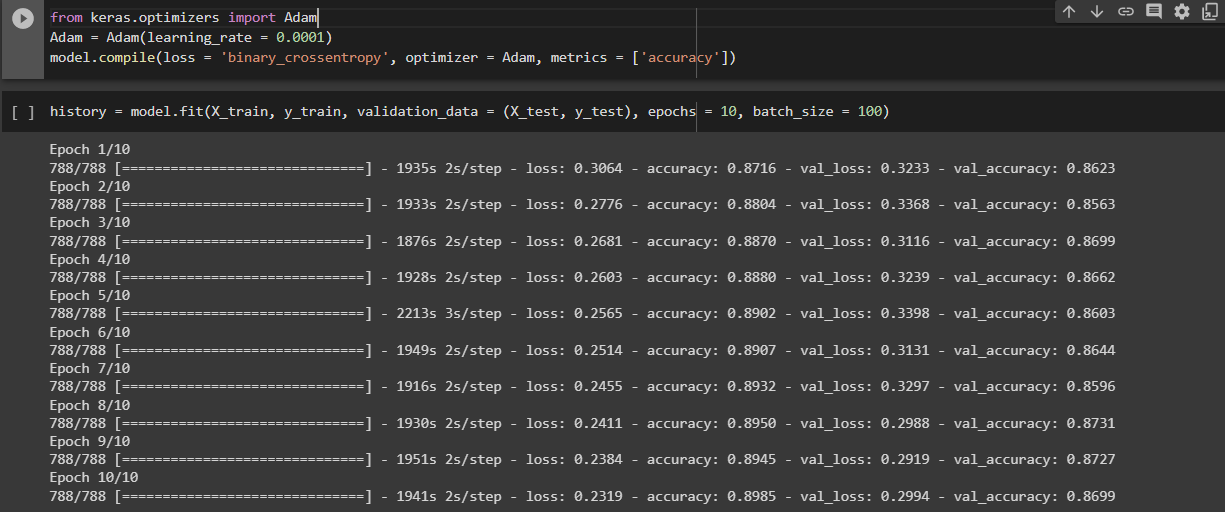
**MODELS USED:**

After splitting the dataset to train and test sets, 3 models were built on the histology dataset and their results are compared.

1. **CNN MODEL**

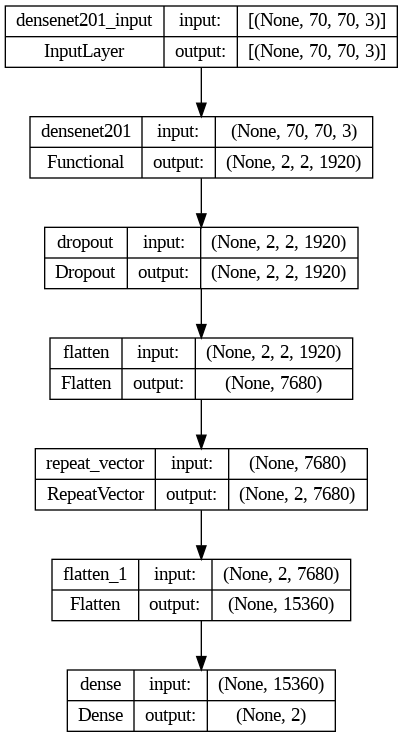
Model designed for image classification with two output classes. It has multiple convolutional layers with varying filter sizes and numbers of filters, followed by MaxPooling2D layers and BatchNormalization layers. Dropout layers are also included to prevent overfitting. The model has a flattened layer followed by two fully connected layers with ReLU activation functions, and a final output layer with a softmax activation function.

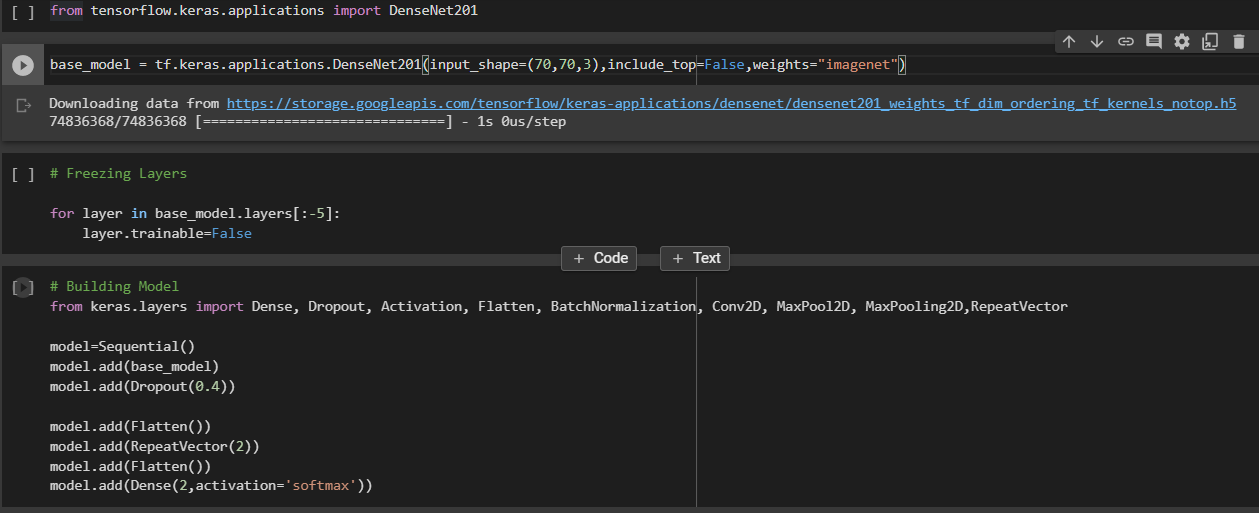


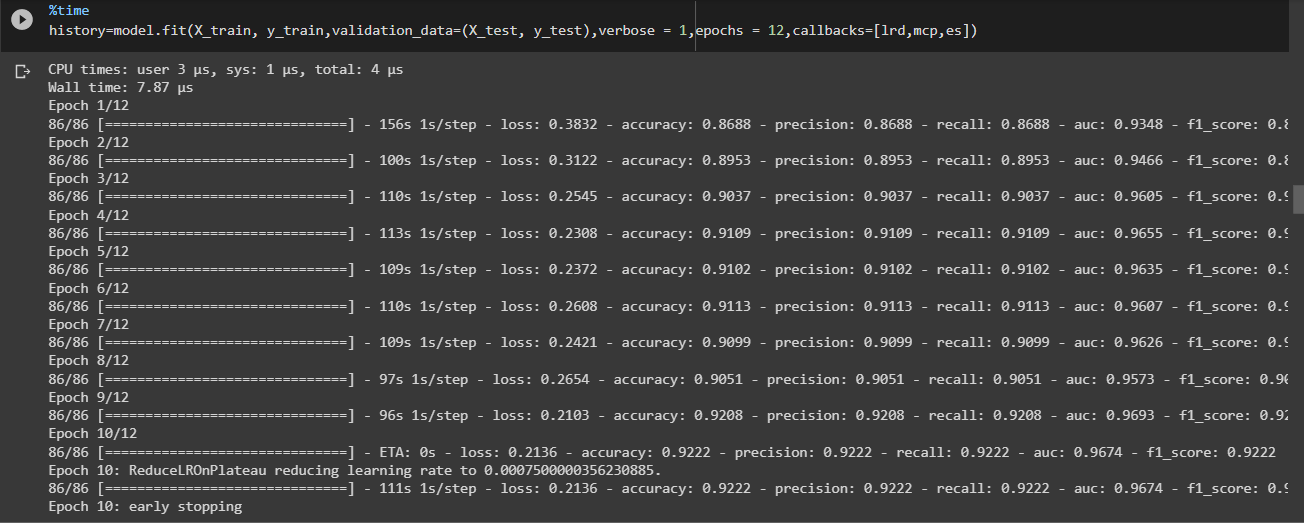


1. **DENSENET MODEL (FINE TUNING):**

Transfer learning-based model using the DenseNet201 architecture pre-trained on the ImageNet dataset. The model's top layers are removed, and a new classification layer is added to the end. The pre-trained layers are frozen, and only the last five layers are made trainable. The model also includes dropout layers to prevent overfitting and a dense layer with softmax activation to produce the final classification output. The input image size is 70x70x3, and the output has two classes.

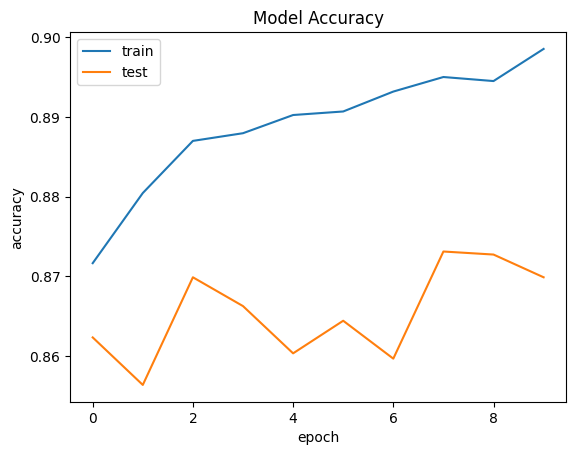


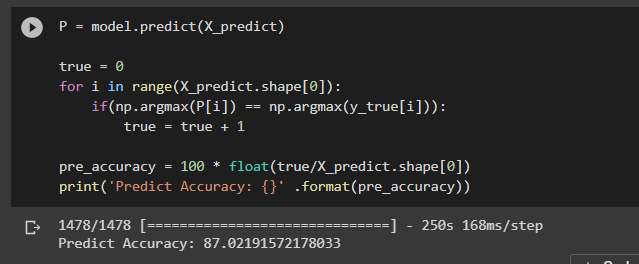




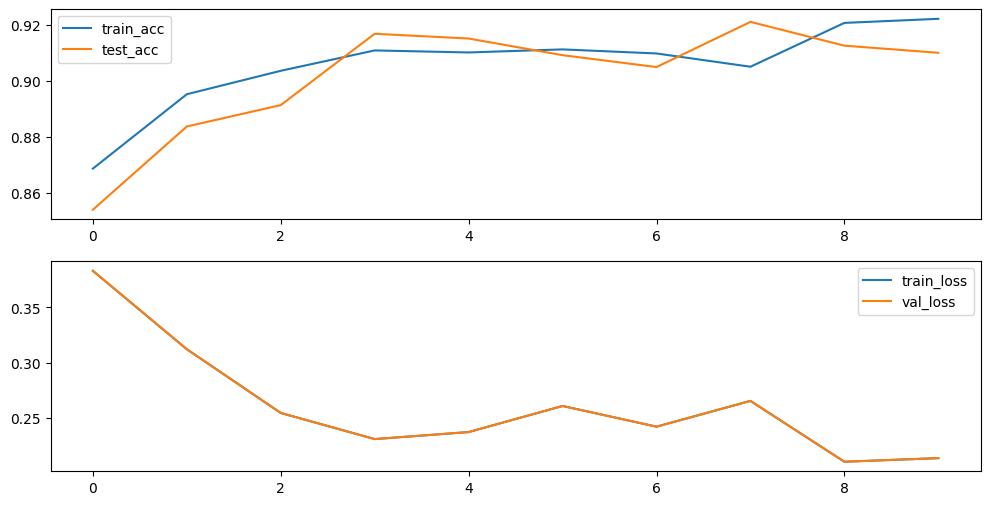
**INFERENCE:**

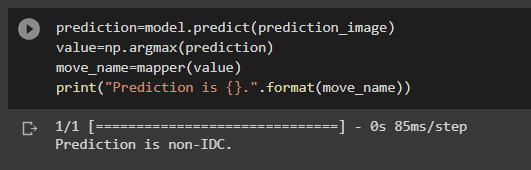
**CNN MODEL:**

****

****

**DENSENET MODEL (FINE TUNING):**

****

****

CNN model was trained to get the accuracy of **87%** and densenet with fine tuning of **90%**

And the image loaded from the dataset was predicted correctly to have IDC cancer or not.

* From the above accuracy plots it is found that for this problem CNN model has huge difference between training and testing accuracy, that is, the model is overfitted.
* But for the **densenet models** both training and testing accuracies are similar. Thus a best model is trained. And **no overfitting** occured.

**DEPLOYMENT:**

The trained model was deployed using **Streamlit**, a Python web application framework. The user can upload an image and the model will make predictions on the uploaded image. The predicted class and probability are displayed to the user.

**Code:**

import streamlit as st

import tensorflow as tf

from PIL import Image

# Load pre-trained model

model = tf.keras.models.load\_model('densemodel\_3\_code.h5')

# Define interface for user to upload image

st.set\_option('deprecation.showfileUploaderEncoding', False)

uploaded\_file = st.file\_uploader("Choose an image...", type="png")

# Define function to make predictions on uploaded image

@st.cache\_data

def predict(image):

# Load and preprocess image

img = Image.open(image).resize((70, 70), resample=Image.BILINEAR)

x = tf.keras.preprocessing.image.img\_to\_array(img)

x = tf.expand\_dims(x, axis=0)

# Make prediction

y\_pred = model.predict(x)[0]

# Return predicted class as a string

class\_names = ['Non-cancer', 'cancer'] # Replace with your own class names

return class\_names[y\_pred.argmax()]

# Run Streamlit app

st.title("Breast Cancer Classification using Histopathology images")

st.write("")

if uploaded\_file is not None:

st.write("Image uploaded!")

st.image(uploaded\_file, caption='Uploaded Image.', use\_column\_width=True)

st.write("")

if st.button("Classify"):

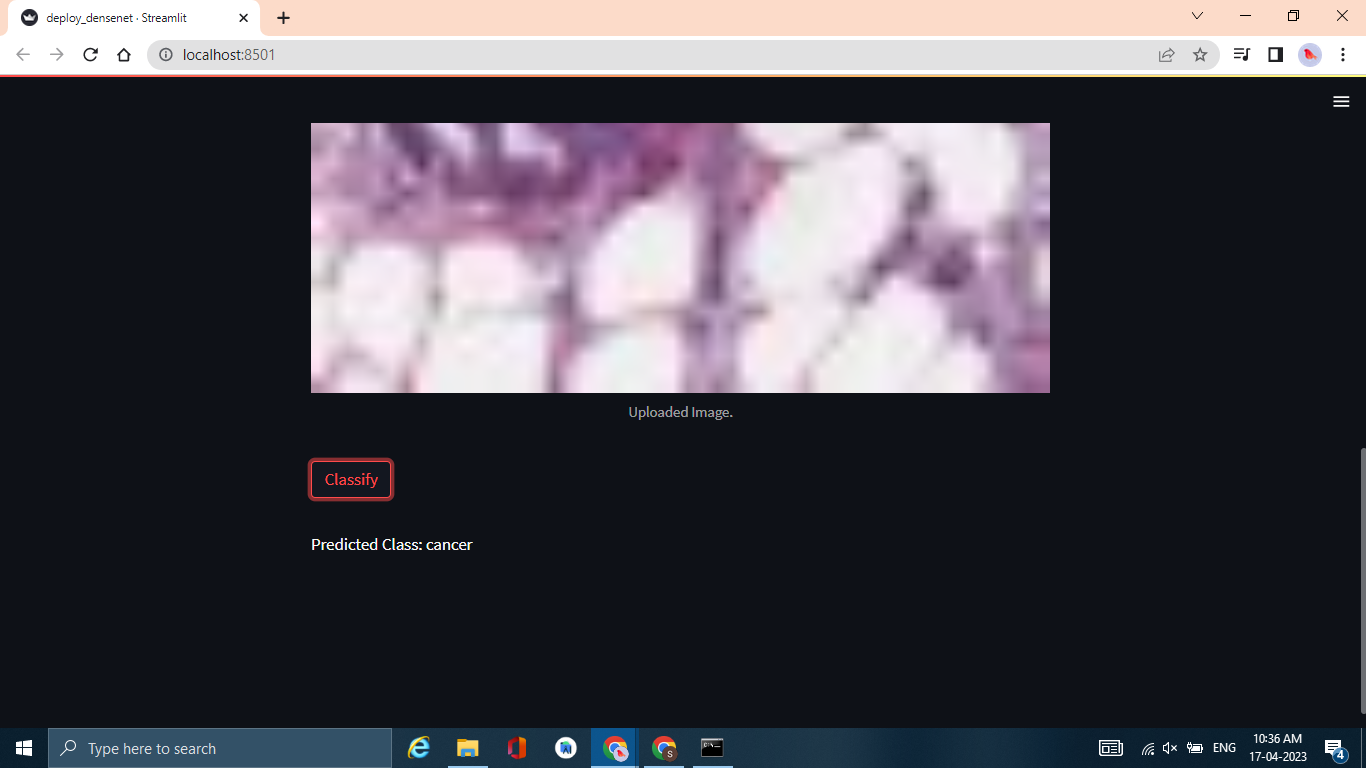
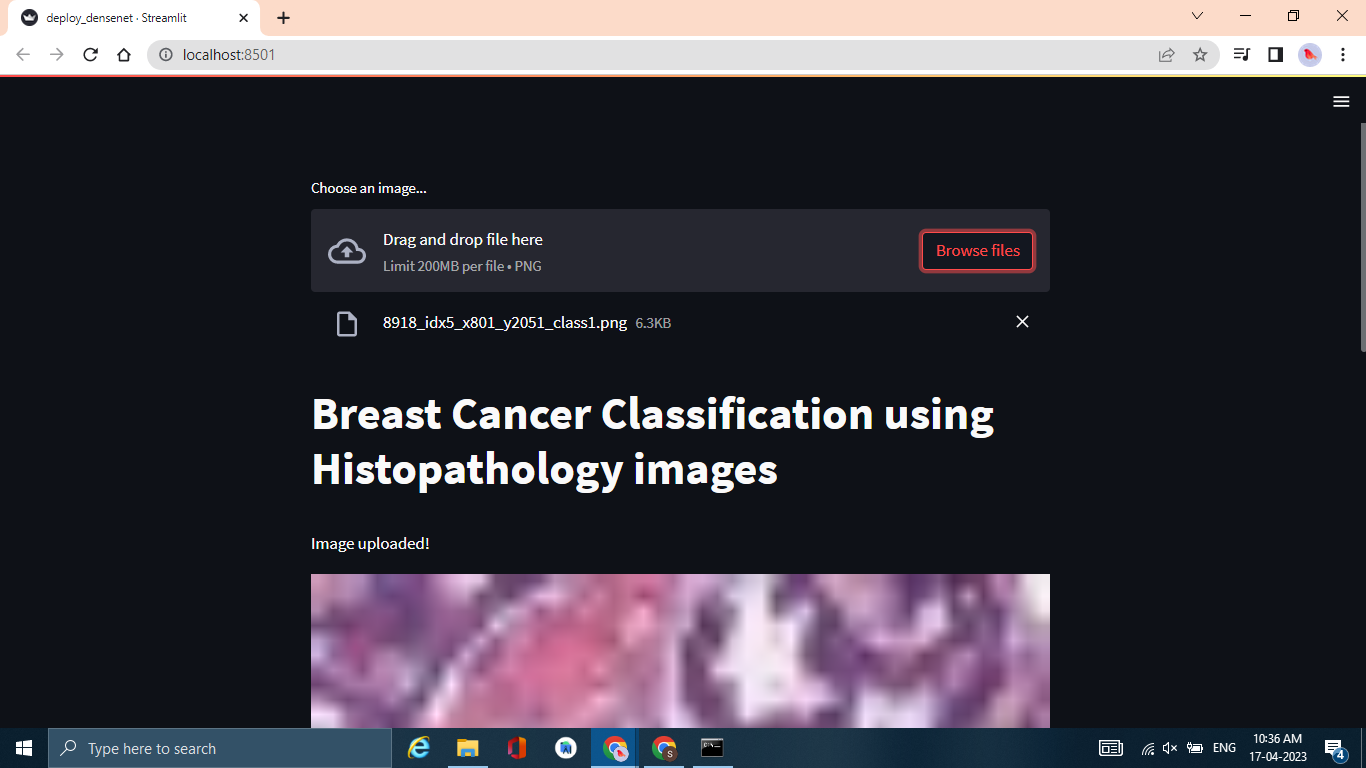
predicted\_class = predict(uploaded\_file)

st.write("")

st.write(f"Predicted Class: {predicted\_class}")

else:

st.write("Please upload an image.")



**CONCLUSION:**

This project demonstrates the effectiveness of deep learning for the detection of IDC cancer in histopathological images. The developed model achieved high accuracy on the validation and test sets, indicating its potential for use in real-world scenarios.