### **Project TumorTrace: MRI-Based AI for Breast Cancer Detection**

### **Infosys SPRINGBOARD 5.0**

### 

Name : Shri Narayan Pandey

Company Name : Infosys

Role : AI/ML Intern

### 

### 

### 

### **AI/ML Internship Progress Notes**

**Day 1 - October 8, 2024**

**Introduction to TumorTrace Project**This project centers on utilizing artificial intelligence to enhance the early diagnosis of breast cancer through MRI image analysis.

**Key Dataset and Concepts**Dataset: [Link to Dataset](https://drive.google.com/file/d/1rKps09z1DEkfiICZlpIBOFek-LZ1vFAC/view)

**Day 2 - October 9, 2024**

**Establishing GitHub and Python Work Environments**

**Step 1: Installing Required Software**

1. Download and install Git from the official website.
2. Install Conda to create an ML environment**.**

**Step 2: Configuring GitHub Repository**

1. Create a new repository on GitHub (e.g., TumorTrace) and adjust its visibility according to your needs.

**Step 3: Cloning the Repository**

**Step 4: Setting Up the Machine Learning Environment**

1. Install Conda and create a Conda environment
2. Activate Conda environment
3. Install jupyter notebook and necessary libraries
4. Run jupyter notebook and start coding

**Day 3 - October 10, 2024**

**Deep Dive into Convolutional Neural Networks (CNN)**

**Reference:** [**CNN Explainer**](https://poloclub.github.io/cnn-explainer/)

CNN Explainer is an interactive tool that simplifies the understanding of Convolutional Neural Networks (CNNs), commonly used in image processing for machine learning tasks. Here are some key points:

**Main Concepts**

1. CNN Overview: CNNs excel at visual data processing. They consist of multiple layers, each designed to capture various features of the input image.
2. Interactive Visualization: This tool allows users to interact with different layers of CNNs to see how they process images and extract features.
3. Layer Breakdown:
   * Convolutional Layer: Extracts features from images using filters.
   * Activation Layer: Applies non-linear activation functions, like ReLU.
   * Pooling Layer: Reduces the size of the feature map while retaining important details.
   * Fully Connected Layer: Combines extracted features for classification.

**Day 4 - October 11, 2024**

**Conducting Exploratory Data Analysis (EDA)**

* Images were resized to 224x224 and transformed into tensors using PyTorch.
* The dataset was loaded using datasets.ImageFolder from PyTorch, which assumes images are organized by class.
* A DataLoader was used to shuffle and create batches for efficient data processing.
* Image visualization techniques were applied to display specific classes and process images as NumPy arrays via matplotlib.

**Implementing data augmentation techniques**

**Data Augmentation:**

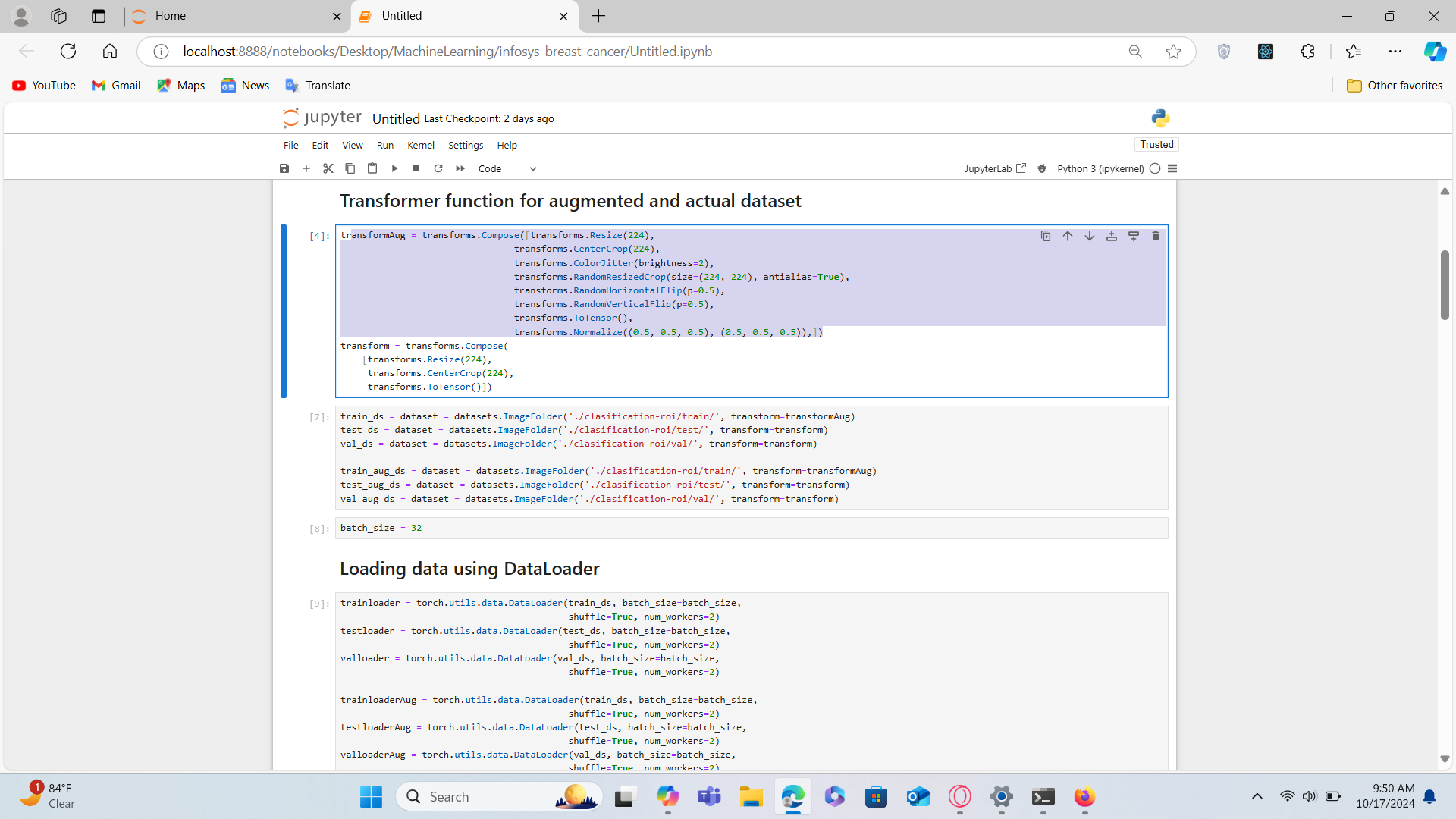
The code applies various transformations (resizing, flipping, rotation, zooming, and color adjustments) to augment the training images,enhancing dataset diversity for better model training.

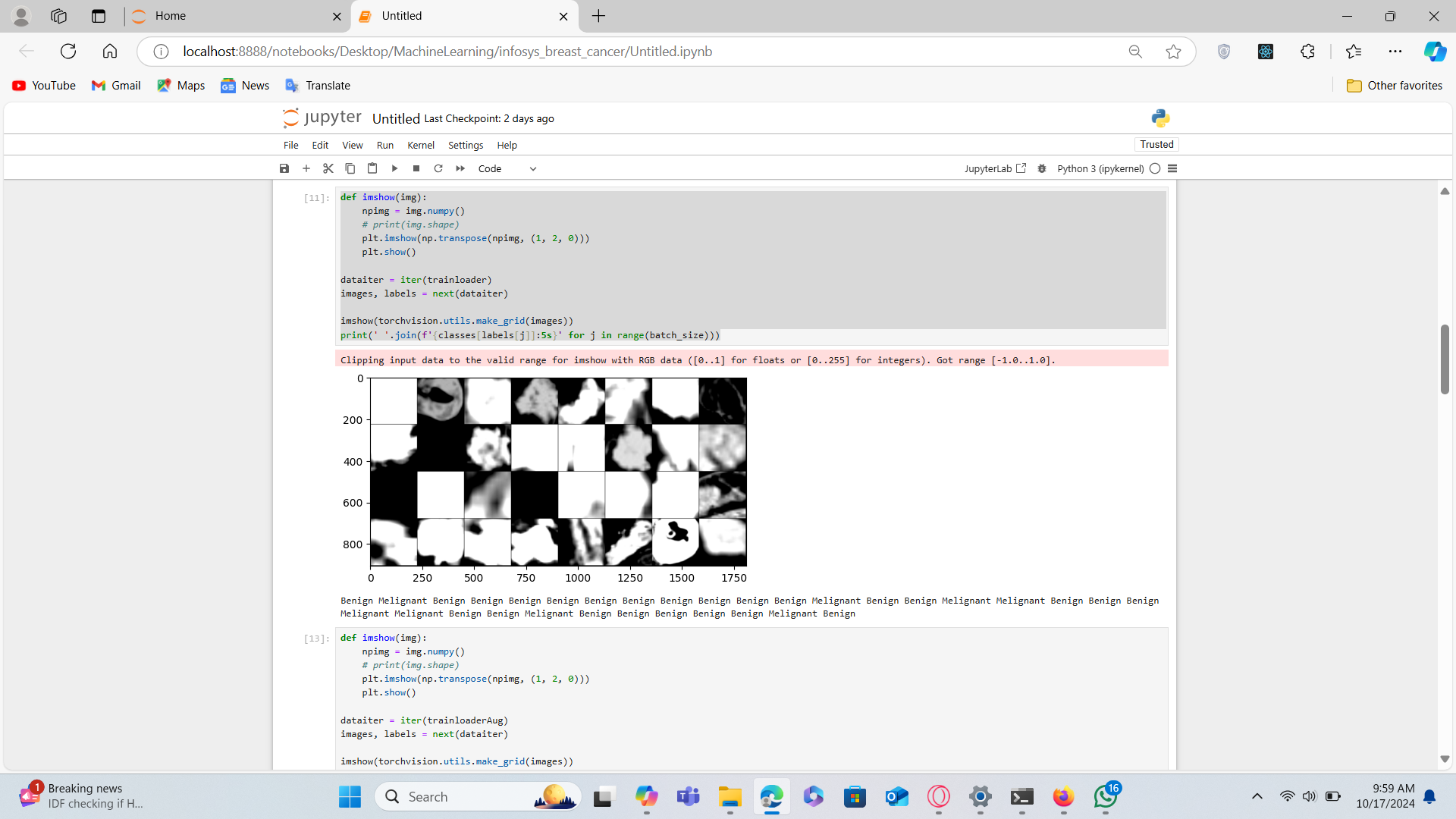
**Dataset Handling:**

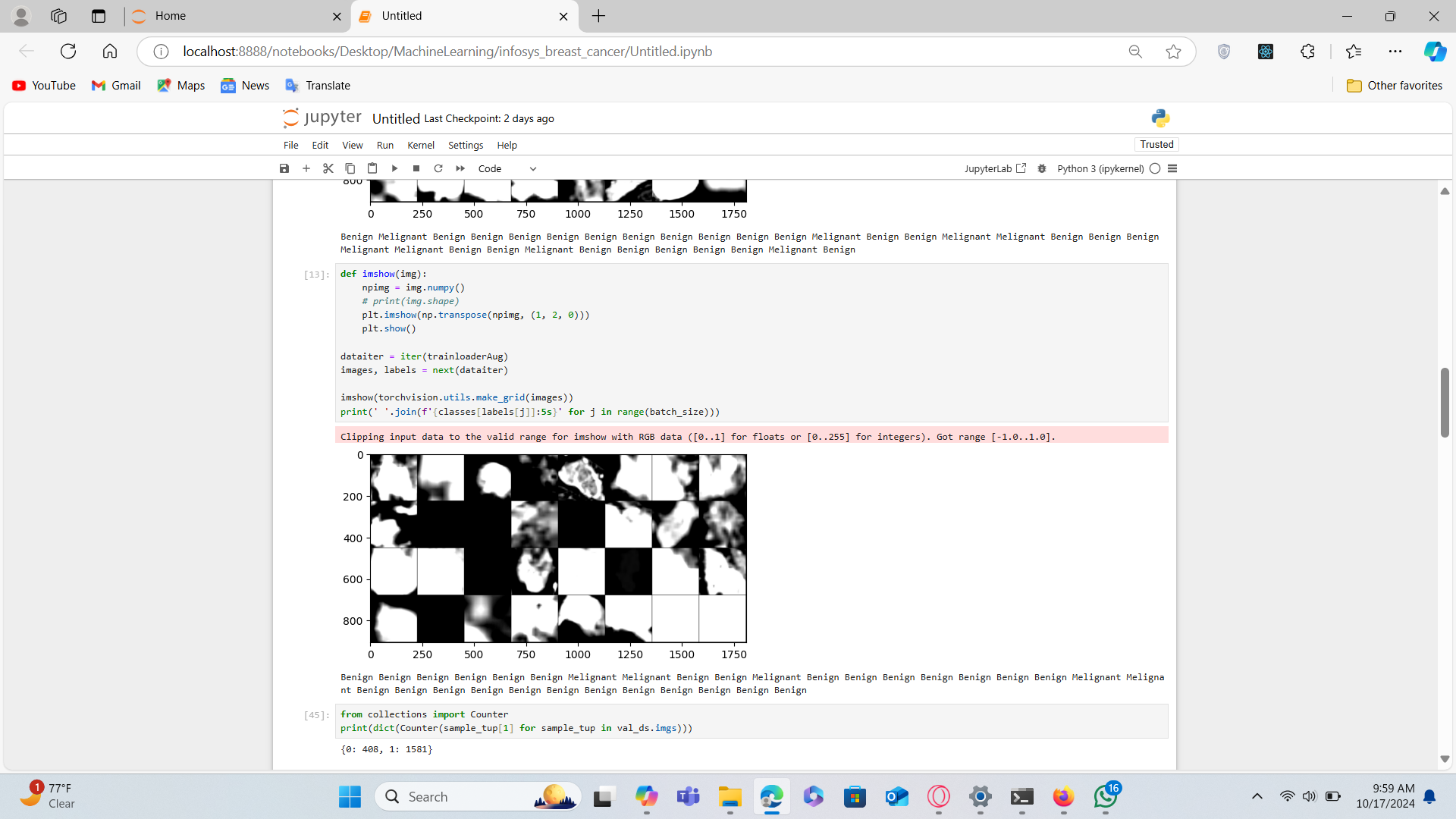
It utilizes torchvision.datasets.ImageFolder to load images organized by class from a specified directory, along with a DataLoader for efficient batch processing and shuffling**.**

**Visualization:**

A function is defined to display a specified number of augmented images, allowing for visual inspection of the transformations applied, which aids in understanding the data variations presented to the model

****

****

****

**Day 5 - October 14, 2024**

**ResNet and VGG16 Architectures**

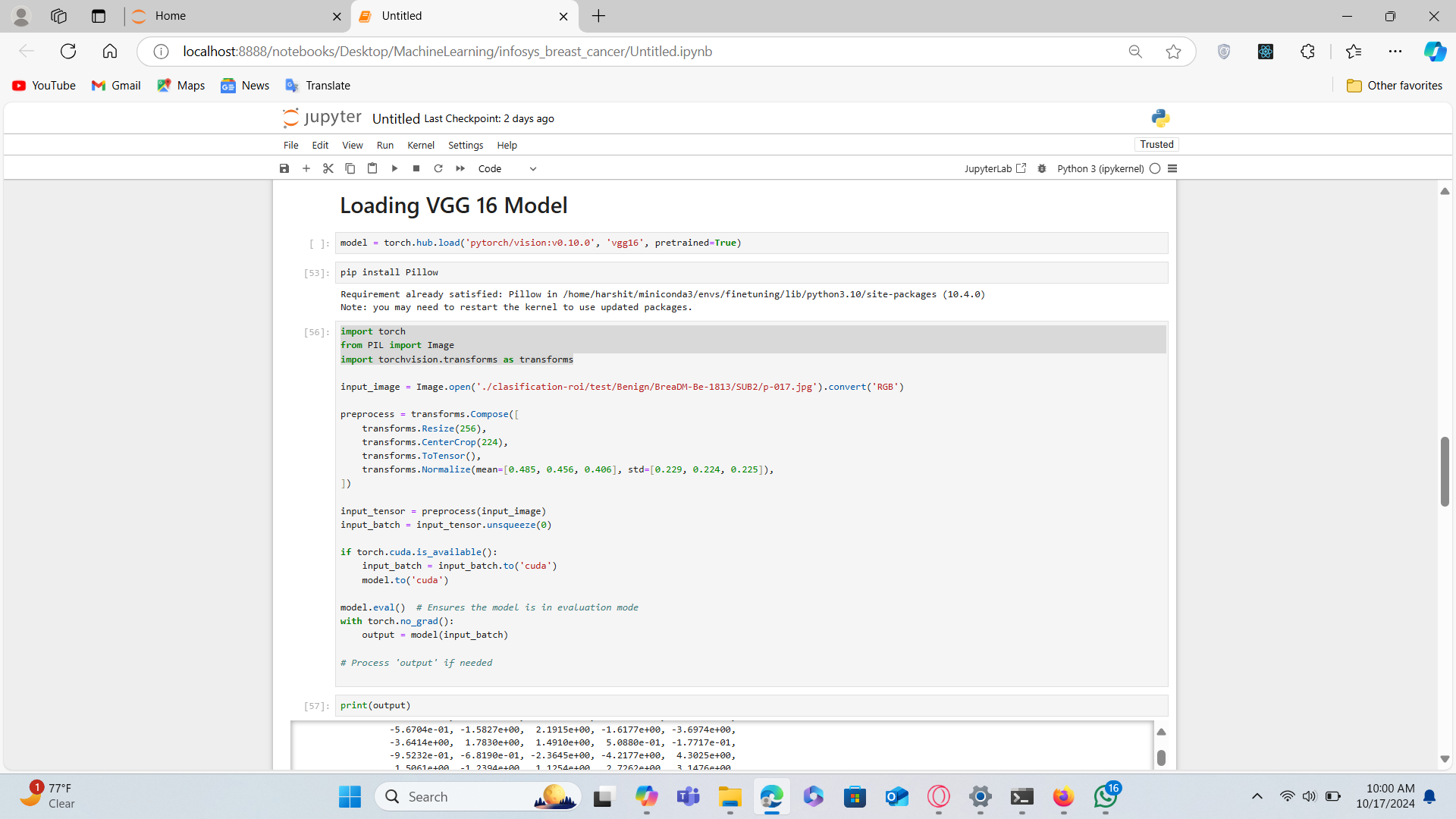
**ResNet Overview:**ResNet (Residual Network) is a CNN architecture designed to train very deep networks efficiently by introducing residual learning, which reduces the vanishing gradient problem. Skip connections allow inputs to bypass certain layers, leading to more effective training even in very deep models like ResNet-50 and ResNet-101.

**VGG16 Overview:**VGG16 is a deep learning architecture consisting of 16 layers with learnable parameters. It uses small convolution filters to learn detailed features while retaining high spatial resolution. Despite its effectiveness, it has a large number of parameters, making it resource-intensive.

**Day 6 - October 15, 2024**

**Loading the VGG16 Model**

**The VGG16 model was successfully loaded for further experimentation and application in image classification tasks.**

****

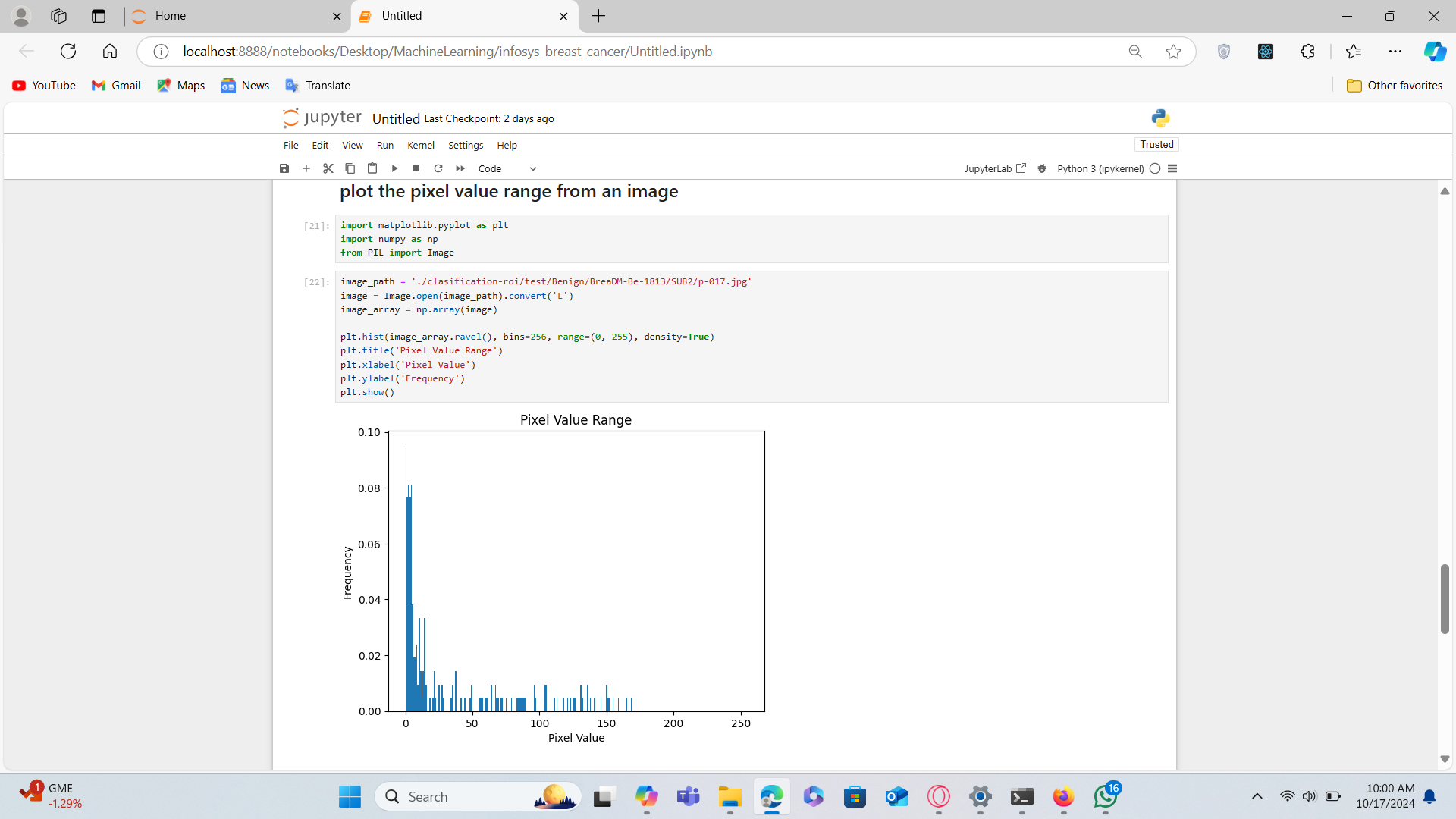
**October 16-October 17, 2024**

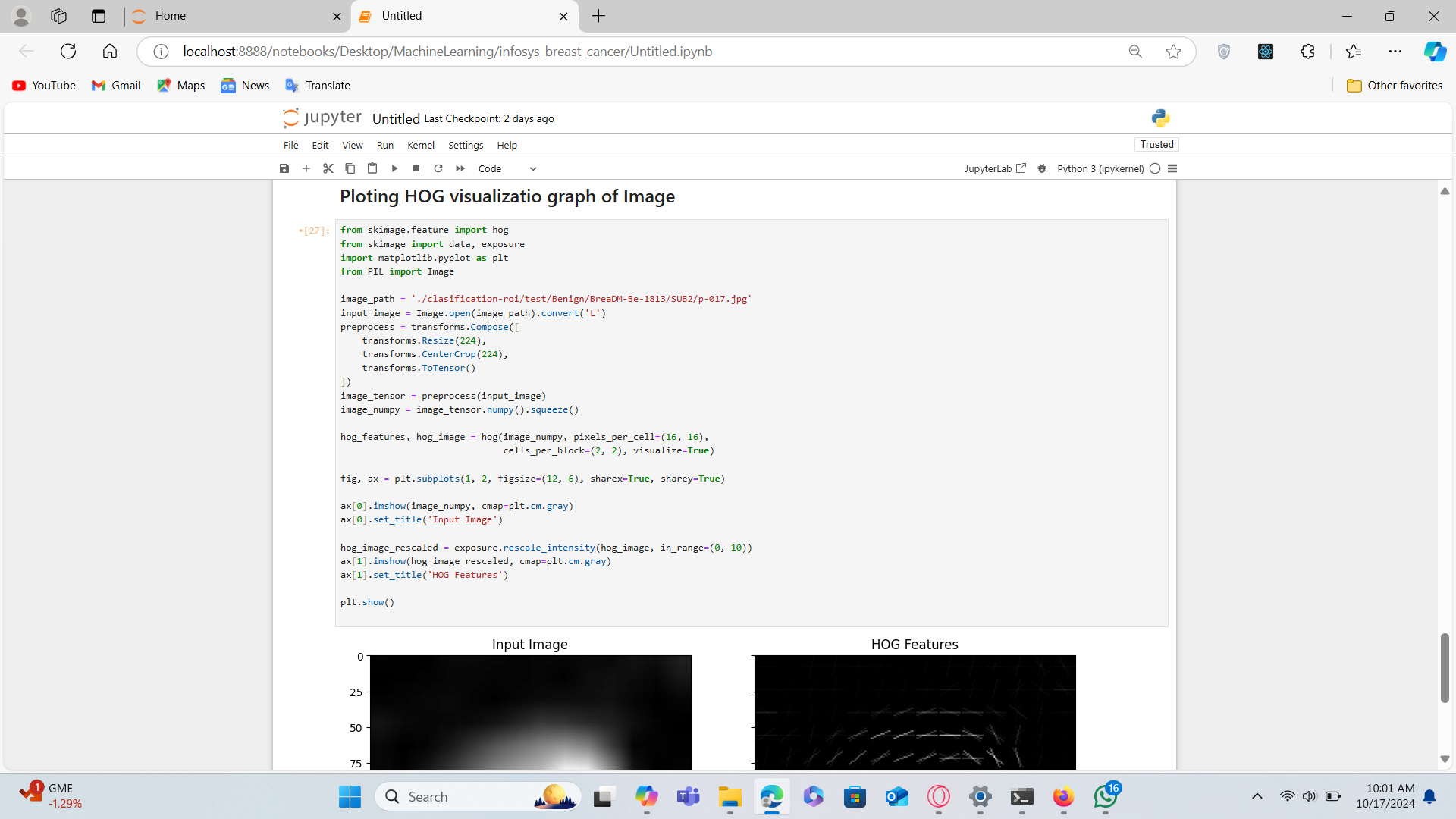
**Histogram of Oriented Gradients (HOG)**

**Overview:**HOG is a feature extraction technique widely used in object detection. It captures the shape and structure of objects by analyzing the gradient orientations in localized parts of an image.

**Steps to Compute HOG:**

1. Preprocessing: Resize the image and normalize it.
2. Gradient Calculation: Compute the image gradient using methods like the Sobel operator.
3. Cell Division: Divide the image into cells (e.g., 8x8 pixels).
4. Histogram Creation: Create a histogram of gradient directions for each cell.
5. Block Normalization: Group neighboring cells into blocks for normalization.
6. HOG Descriptor: Concatenate the histograms from all blocks into a feature vector.

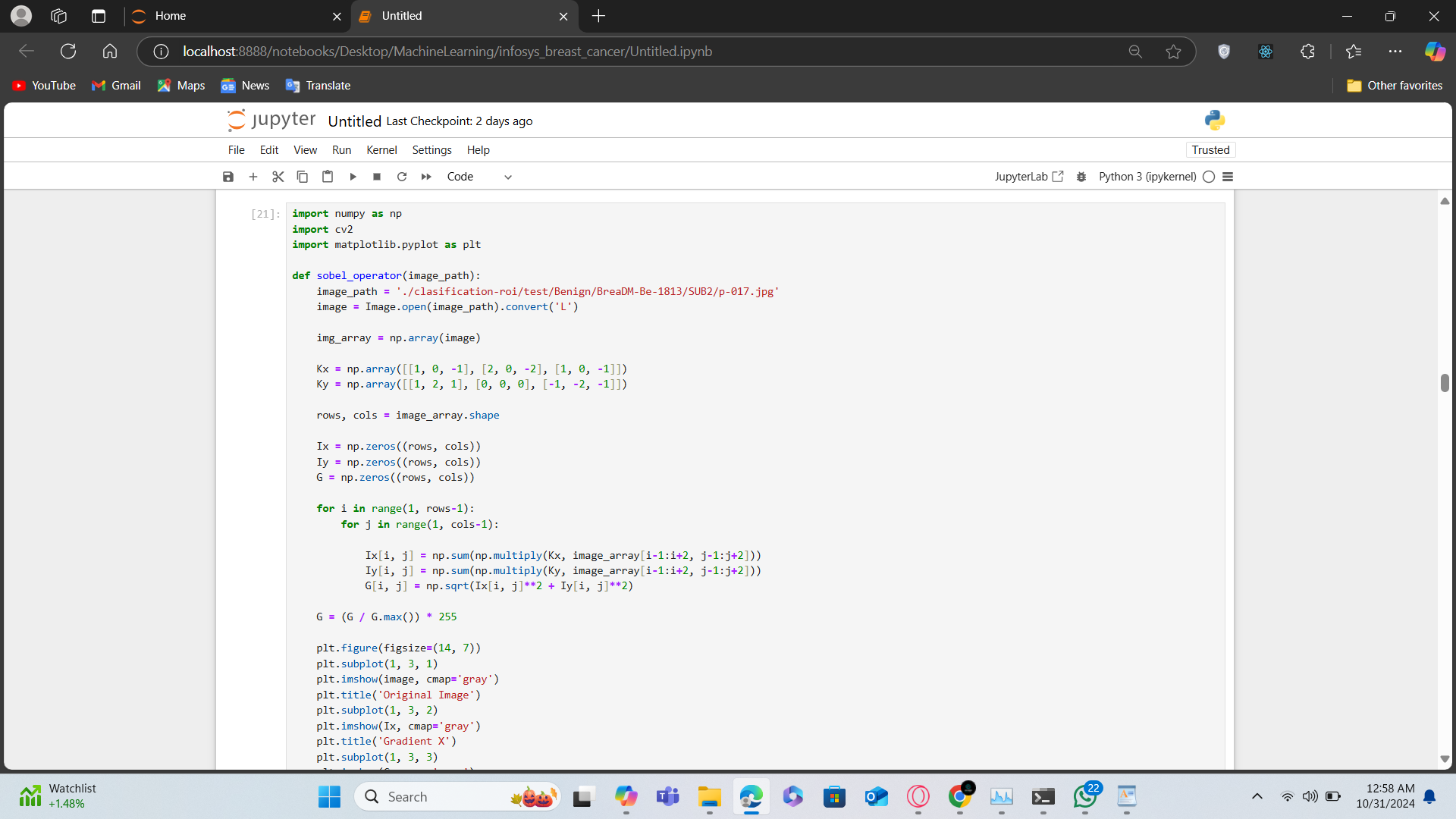
****

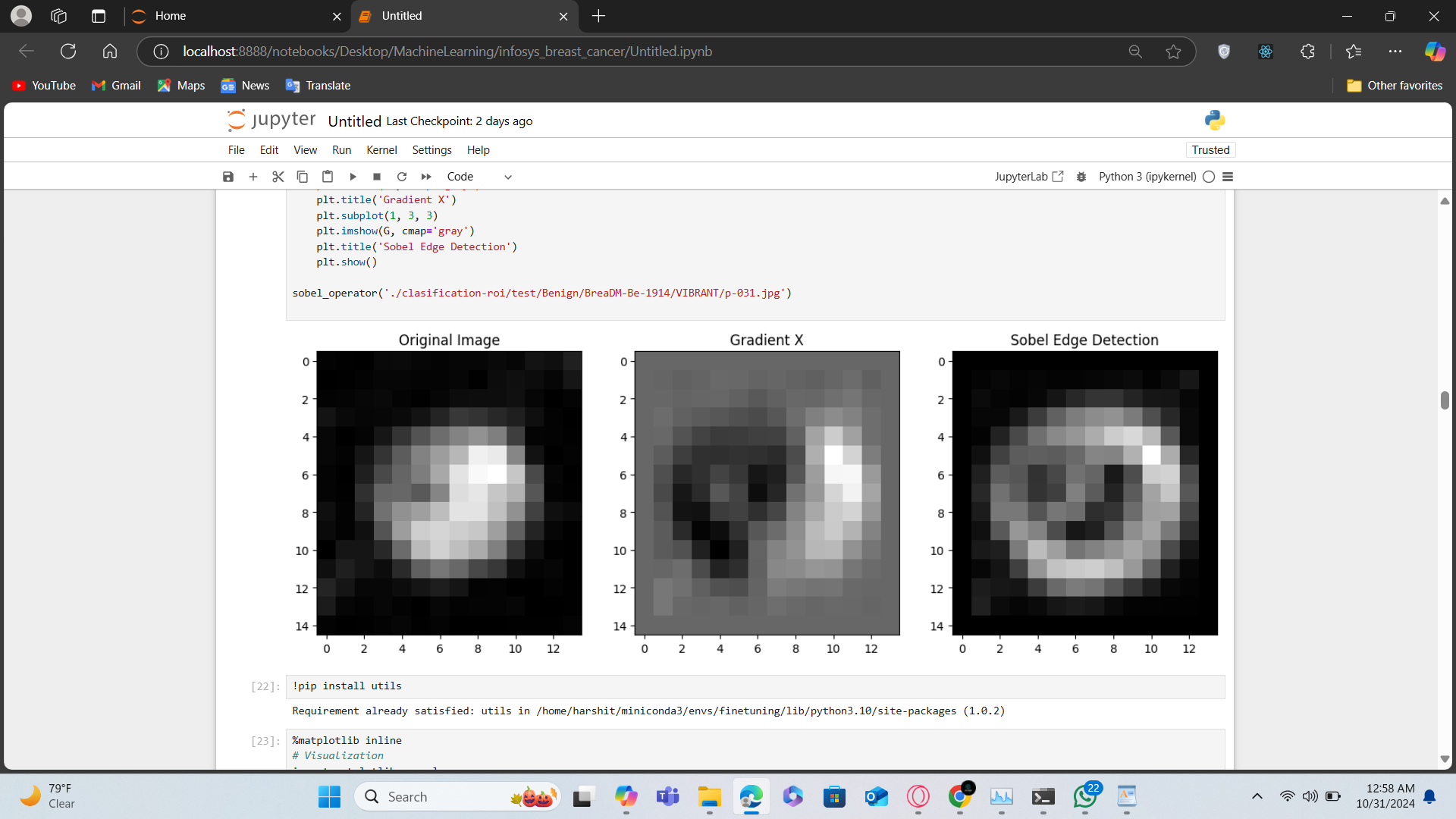


**October 21, 2024**

**Edge detection via Sobel Edge Detection**

**Overview:**Studied various edge extraction techniques and learned about sobel edge detection and implemented it





**October 22, 2024**

**Learned about Local Binary Points**

**Overview:**

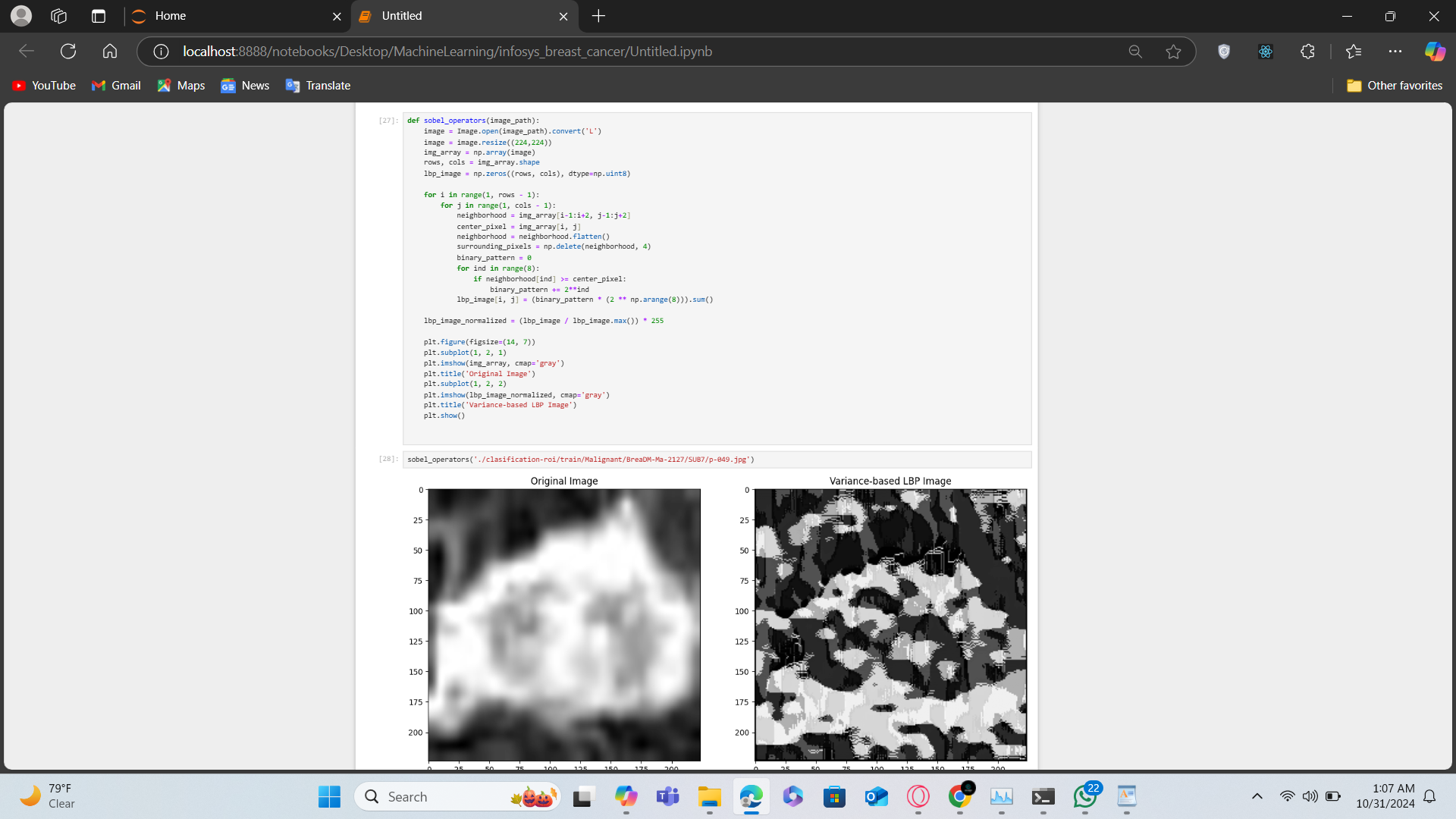
Local Binary Pattern (LBP): Overview of LBP as a texture analysis technique, encoding spatial structure to enhance feature extraction

**October 23 and October 25, 2024**

**Local Binary Points implementation**

**Overview:**

Local Binary Pattern (LBP): Overview of LBP as a texture analysis technique, encoding spatial structure to enhance feature extraction and implemented LBP

****

**October 28 , October 29 and October 30 , 2024**

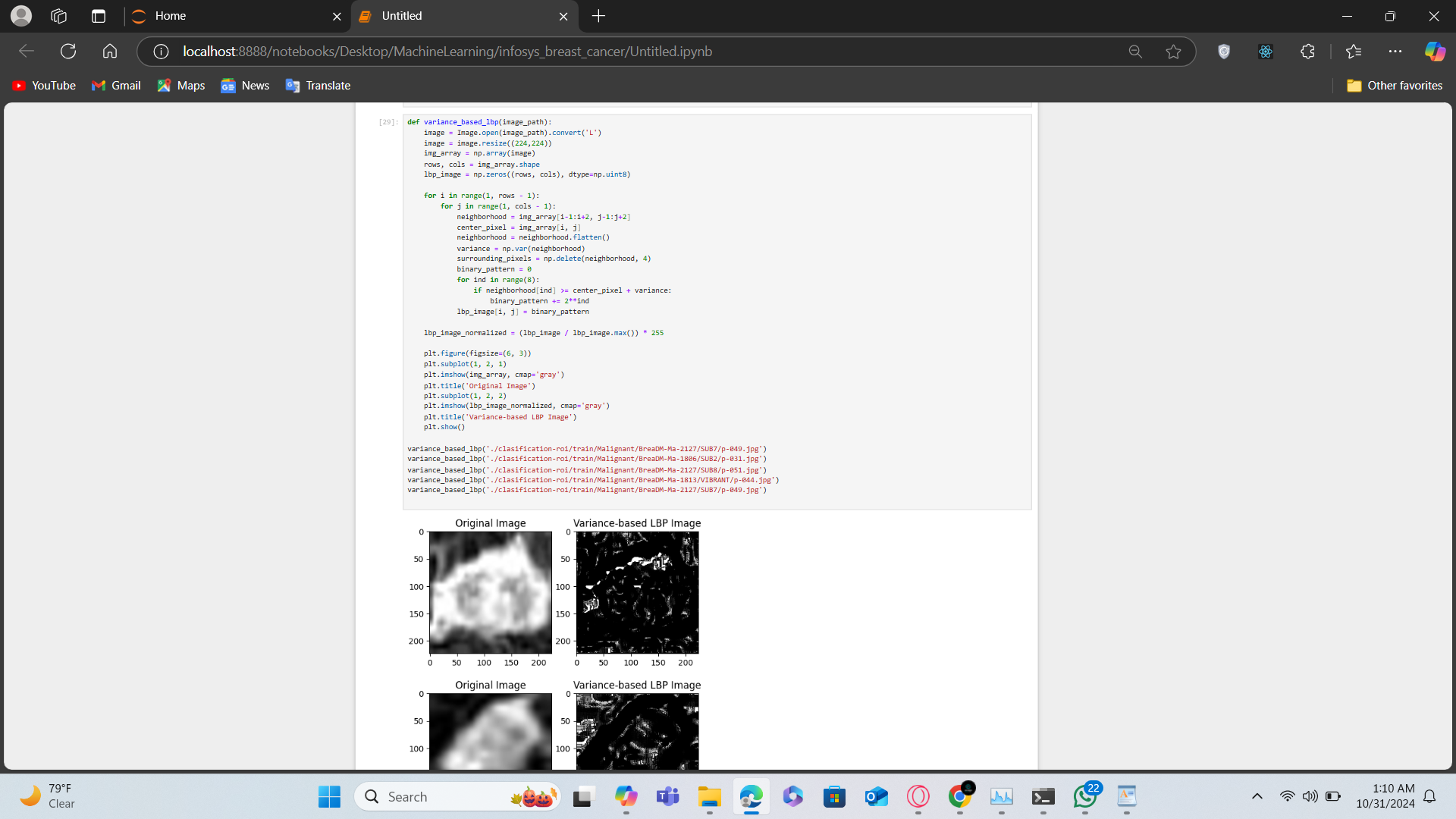
**Mean Variance Median Binary Points**

**Overview:**

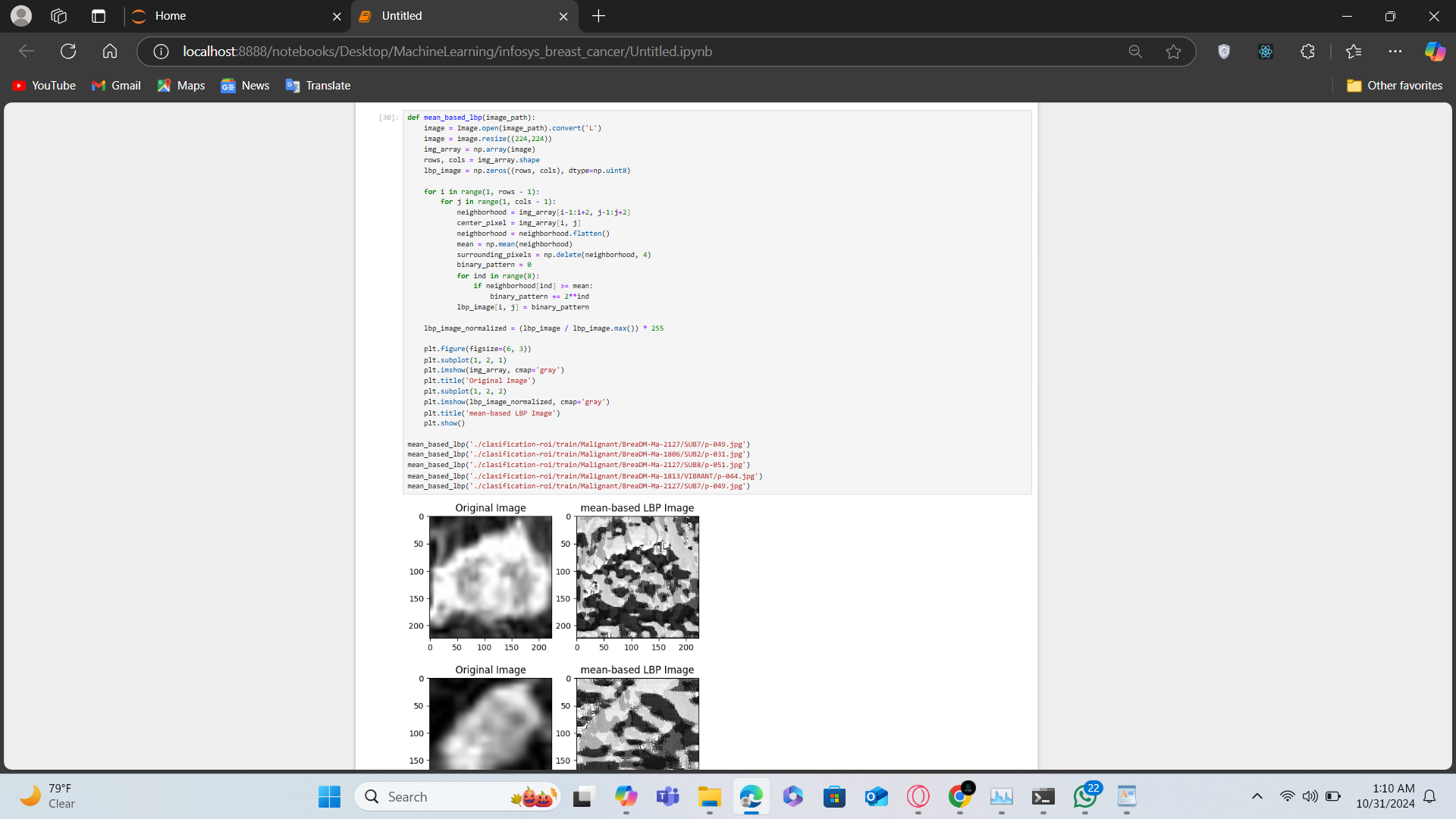
**A traditional technique used for texture classification and feature extraction in face recognition. This method computes the mean, variance, and median of the pixel intensities in a local neighborhood.**

**We Implemented MVMBP including Mean Binary Point , Variance binary Point , Binary Median Point**

**Variance Based Binary Points:**

****

**Mean Based Binary Points**

****

**Variance Based Binary Points:**

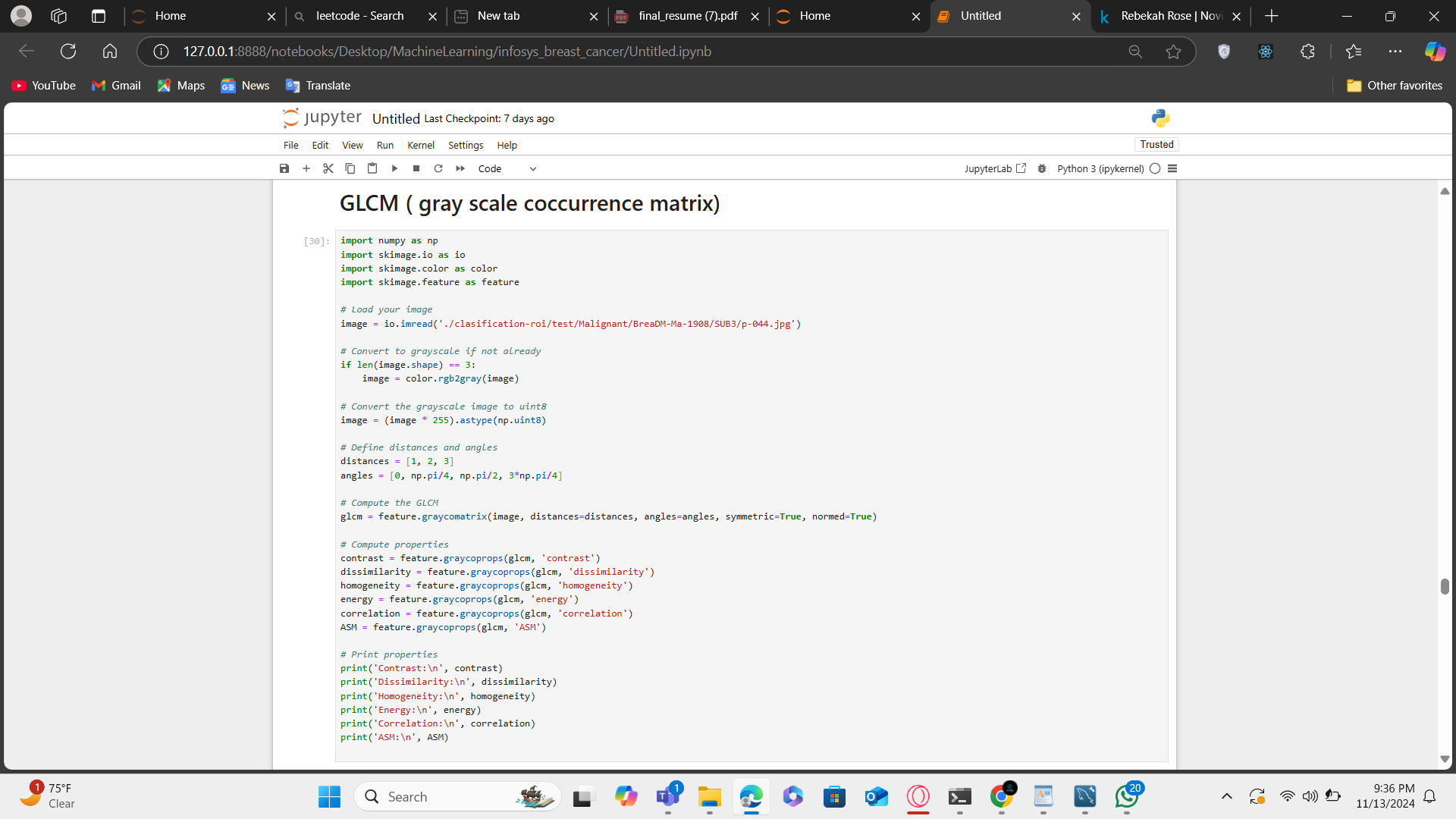
****

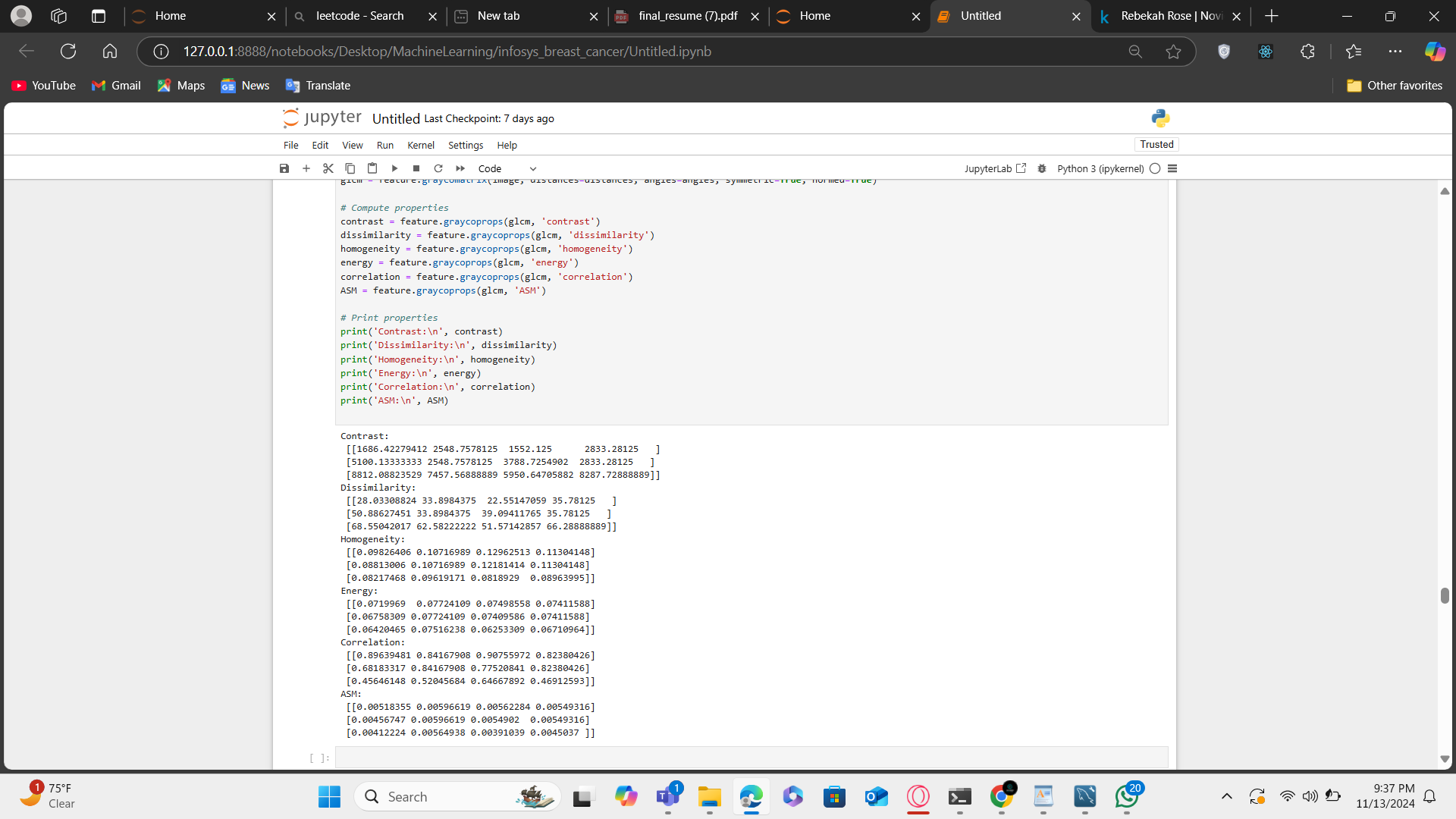
**MVMBP:**

**October 31 , November 1**

**GLCM (Gray-Level Co-occurrence Matrix)**

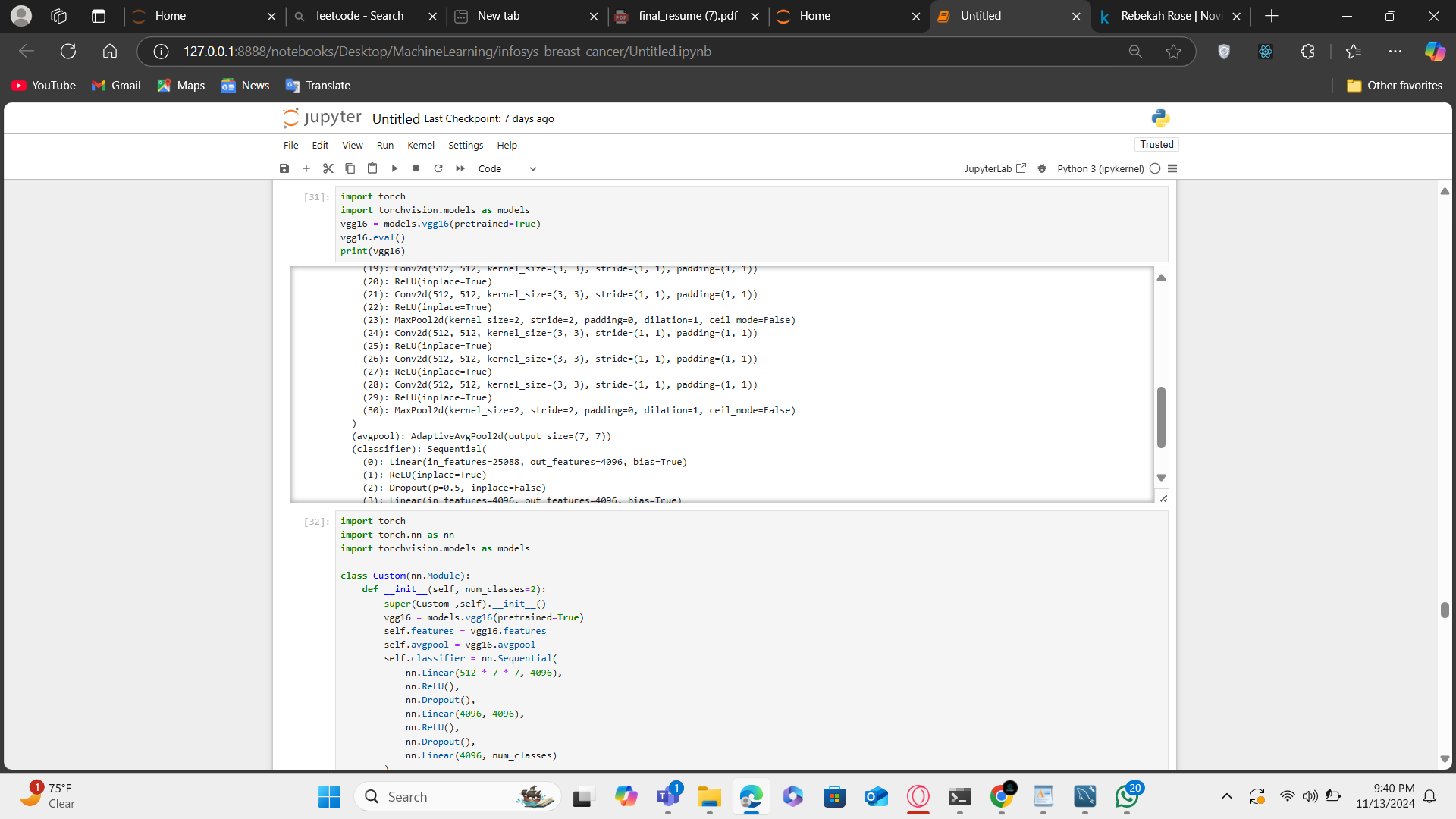
**Overview:**

****

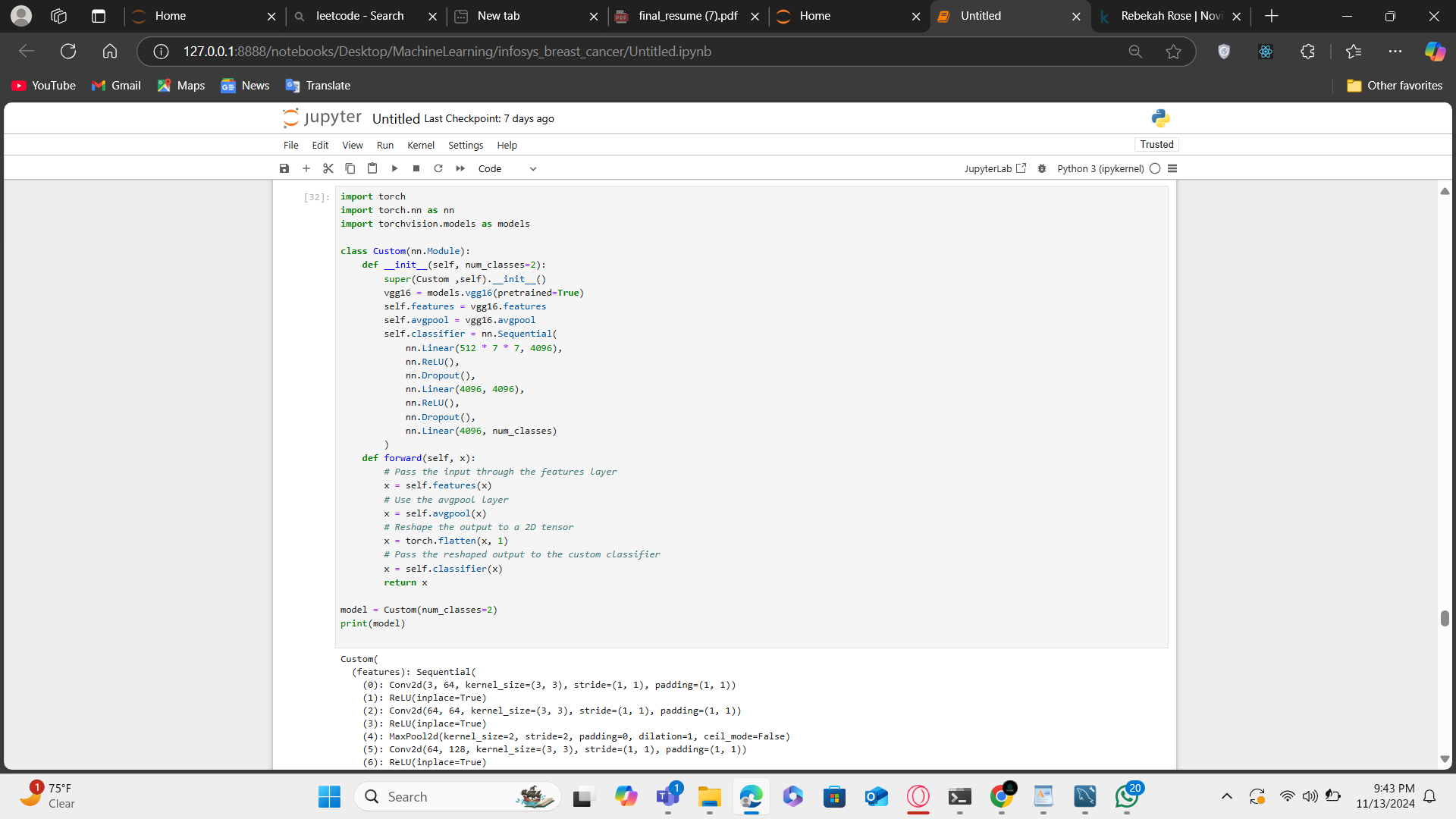
****

**November 4 - November 6:**

**Loading And visualizing Vgg16**

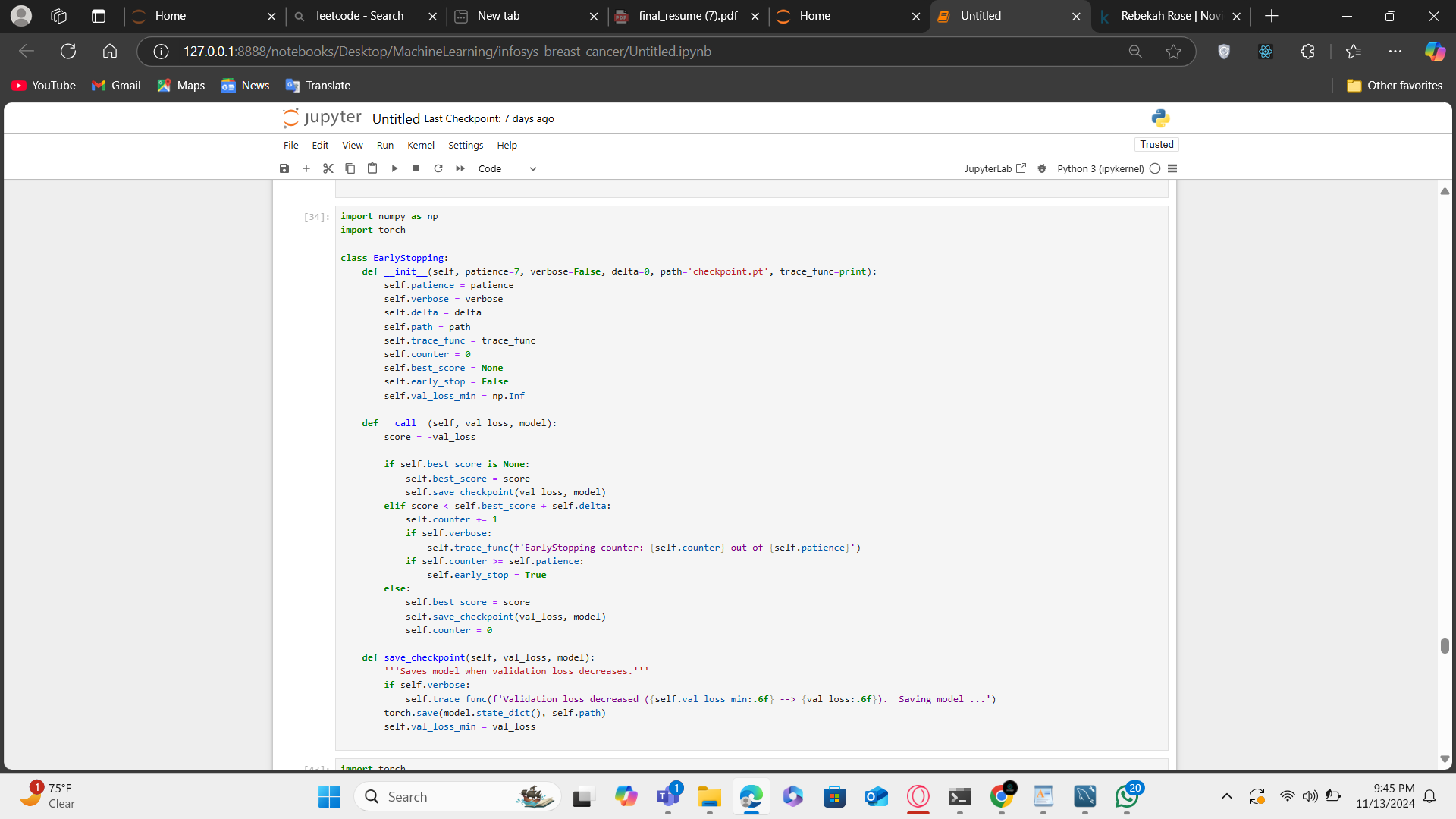
****

**Loading custom Vgg16**

****

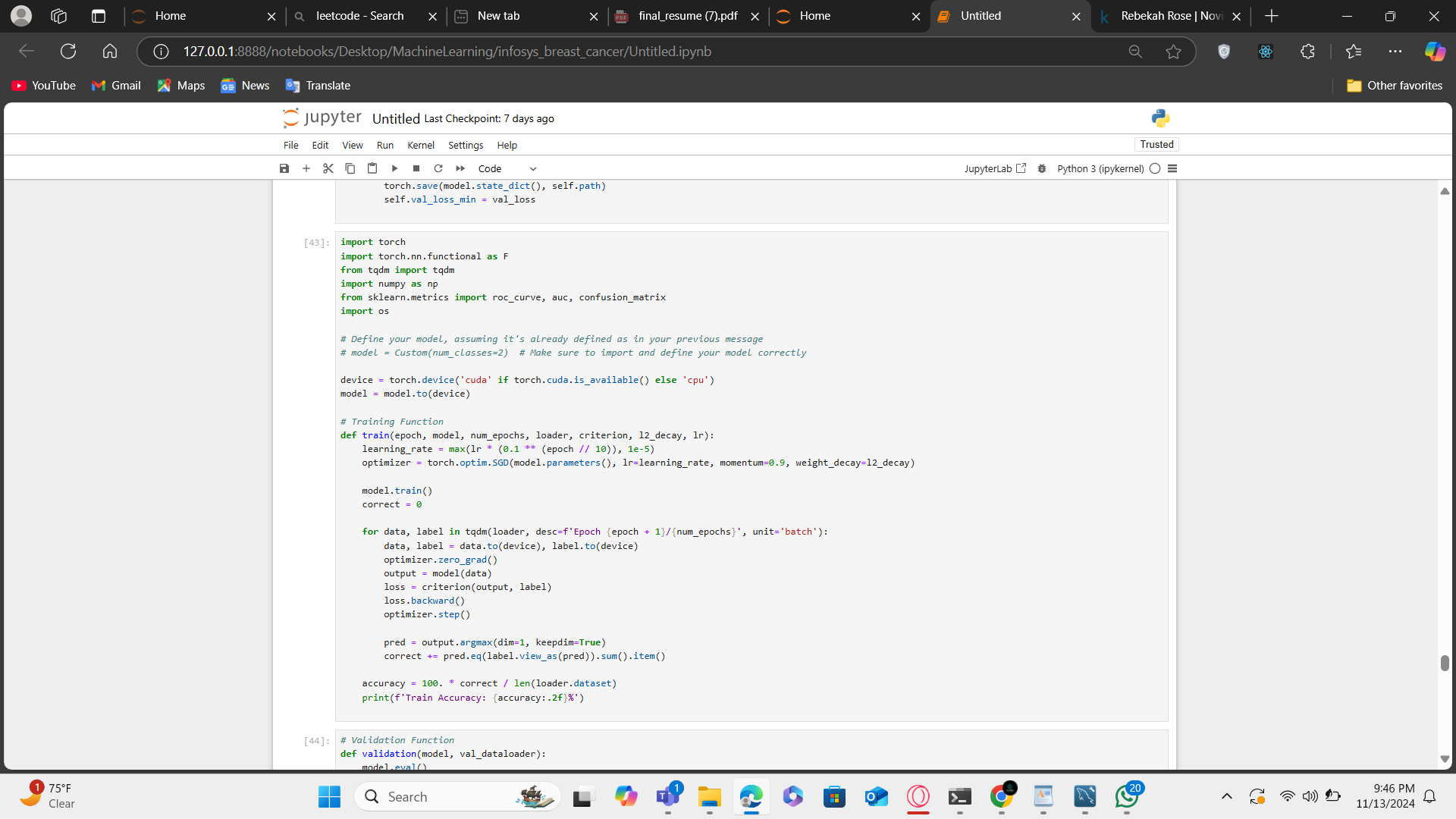
**November 7**

**Creating EarlyStopping class for model:**

****

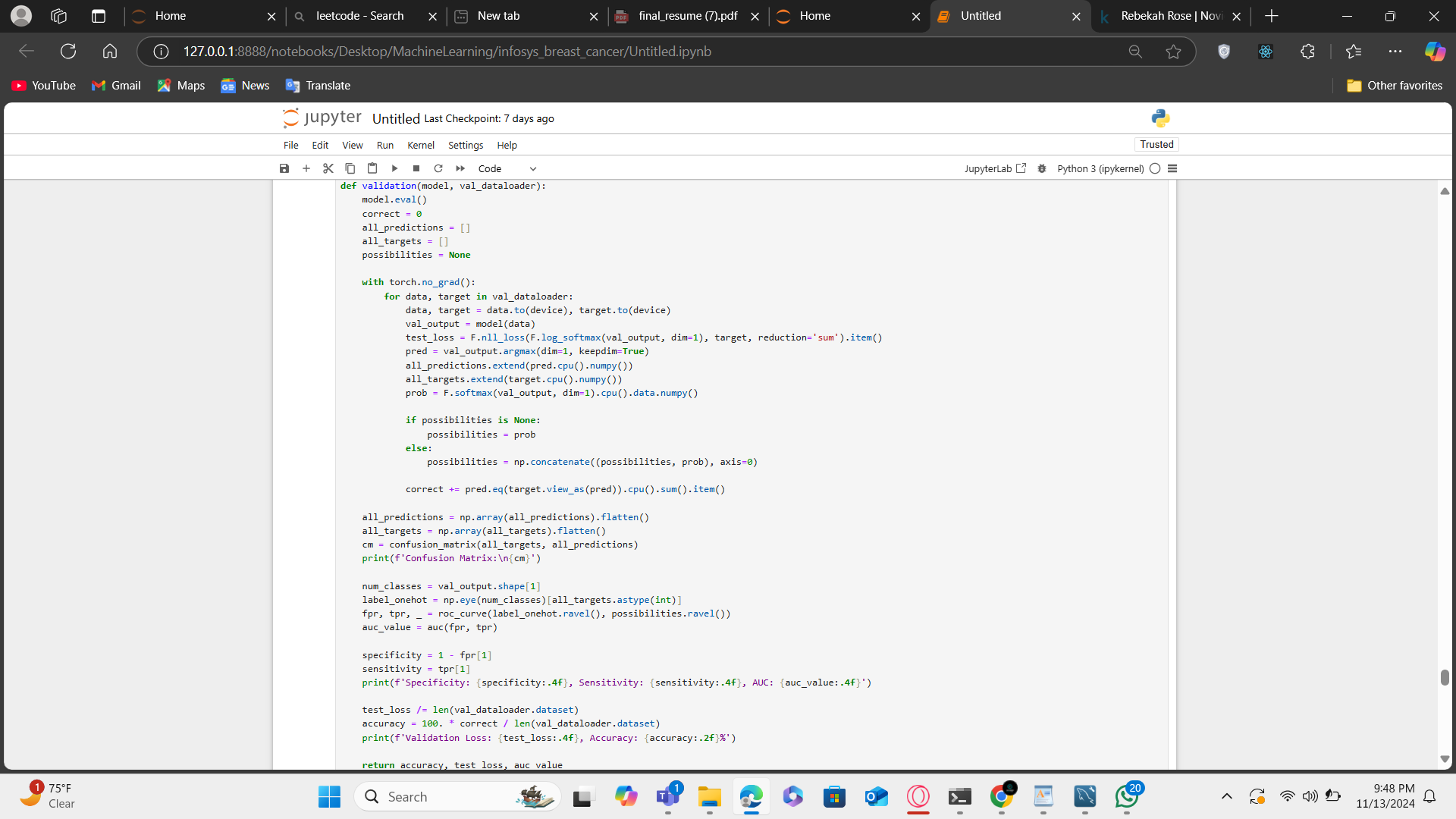
**November 7**

**Creating Training class for model:**

****

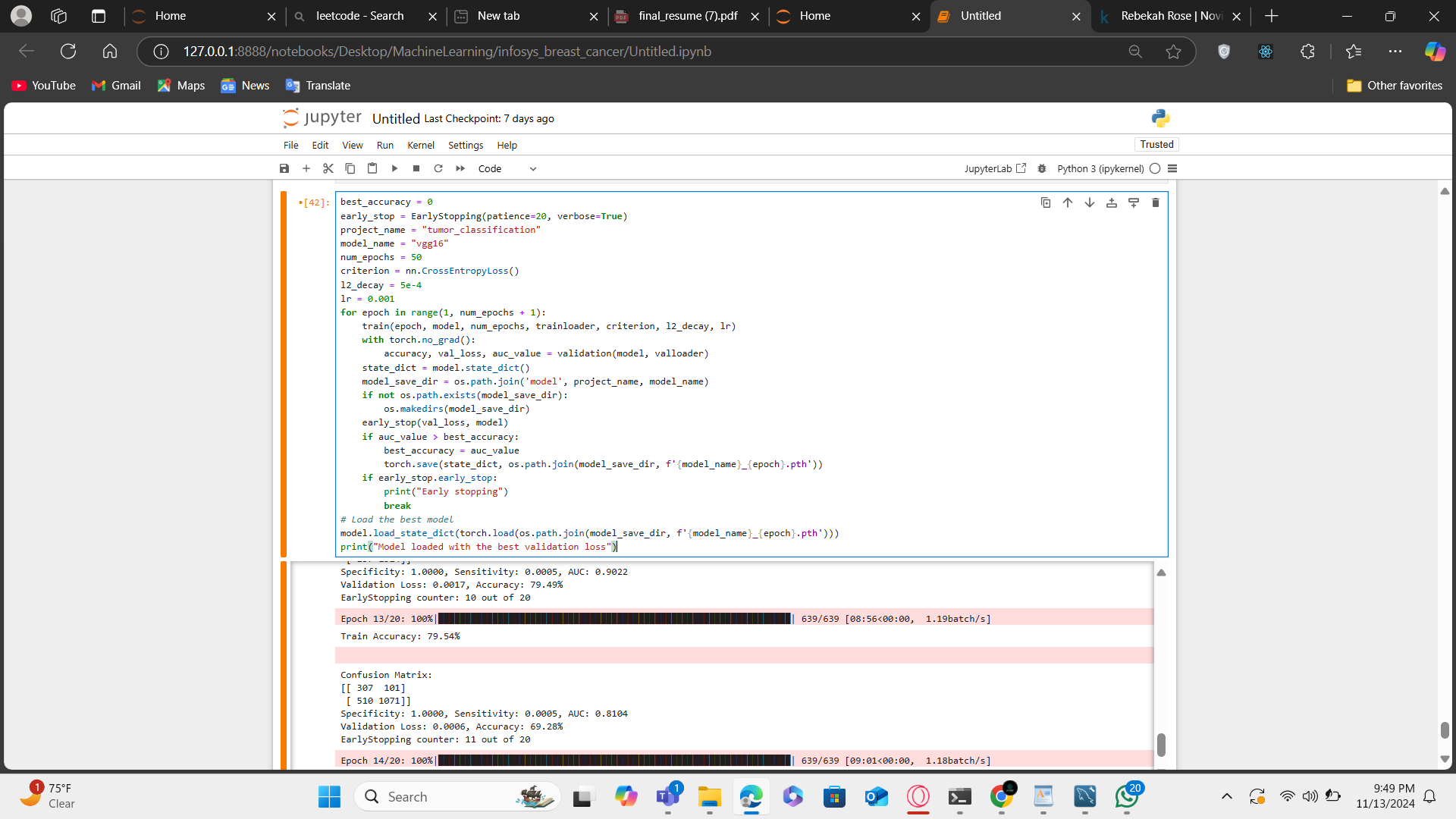
**November 8**

**Creating Validation class for model:**

****

**November 9 - 12 :**

**Training and Validation of model**

****