IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

ULTRASOUND IMAGE CLASSIFICATION OF THYROID NODULES BASED ON DEEP LEARNING

[1]V. Sai Vedanth, [2]M. Sai Rohan, [3]S. Vineeth Sai, [4]Dr.T. Ramaswamy, [5]Dr.Y. Sreenivasulu,
[1,2,3]Student, [4]Assocaiate Professor, [5]Professor
[1,2,3,4,5]Department of Electronics and Communication Engineering,
[1,2,3,4,5] Sreenidhi Institute of Science and Technology, Hyderabad, India.

Abstract: Thyroid nodules, which are characterised as aberrant thyroid cell growth, might be an indicator of an excessive iodine intake, thyroid degeneration, inflammation, and other diseases. Despite the fact that thyroid nodules are mostly benign, the likelihood that they are malignant rises noticeably every year. The labour of medical practitioners is reduced and unnecessary fine needle aspiration and surgical excision are avoided and thanks to a number of studies that have been done to identify thyroid nodules using deep-learning-based image recognition analysis.

A novel deep learning architecture is proposed in this study to reliably identify benign and malignant thyroid nodules. We imported this data from Algeria's hospitals exactly in Setifcity, and it was labelled by volunteer doctors. There are 1546 images belonging to three different classes namely normal thyroid, benign and malignant nodules where benign referring to less harmful and malignant being highly dangerous. First, a ResNet50 model pretrained on ImageNet database was applied to a collection of ultrasound thyroid images.The trained model has now able to classify the thyroid nodules into different categories like normal, benign and malignant. Overall, the proposed model demonstrated that ultrasound images and deep learning may be used to distinguish between benign and cancerous thyroid nodules.

***Index Terms* - Thyroid, Benign, Malignant, ResNet50, Deep Learning.**

I. INTRODUCTION

Thyroid disease is a part of endocrinology, which is one of the most important poorly understood and under detected diseases. Diabetes and disorders of the thyroid gland are the two endocrine illnesses that are most prevalent globally, according to the World Health Organization. Hypothyroidism and hyperfunction hyperthyroidism affect, respectively, 2% and 1% of people. Men outnumber women by an amount that is around one eighth. Potential causes of hyper- and hypothyroidism include disorders of the thyroid gland, pituitary gland, and hypothalamus. Due to dietary iodine deficiency, goitre or active thyroid nodules may become prevalent, with a prevalence of up to 15%, in some regions. Endogenous antibodies can mess with the thyroid gland, which can house a variety of cancers and be dangerous (autoantibodies). According to specialists, limiting disease progression and even death depends on early disease identification, diagnosis, and management. The possibility of effective therapy for a range of disorders is increased by early detection and differential diagnosis. Despite many attempts, clinical diagnosis is usually seen as a very hard task.

The thyroid organ is a butterfly-formed organ organized at the underpinning of the throat. It contains two powerful thyroid synthetic compounds, levothyroxine (T4) and triiodothyronine (T3), which are locked in with mind works, for instance, inner intensity level control, circulatory strain the leaders, and heartbeat rule also thyroid disorder is quite possibly of the most unavoidable sickness all over the planet, and it is generally achieved by an absence of iodine, but it could similarly be achieved by various factors. The thyroid organ is an endocrine organ that secretes synthetics and goes them through the flow framework. It is organized toward the front of the body. Thyroid organ synthetic compounds are obligated for assisting with ingestion as well as staying aware of the body wet, changed, and so on. Thyroid organ prescriptions like T3, T4 and TSH are used to study thyroid activity (thyroid fortifying substance). Thyroid disturbance is requested into two sorts: hypothyroidism and hyperthyroidism. Data mining is a semi-mechanized technique for looking for associations in enormous datasets.

AI and profound learning calculations are one of the most mind-blowing answers for some issues that are challenging to settle Grouping is an information extraction method (AI) used to foresee and distinguish numerous illnesses, for example, thyroid sickness, which we explored and ordered here since AI calculations assume a critical part in characterizing thyroid infection and on the grounds that these calculations are high performing and productive and help in characterization. Albeit the utilization of PC learning and computerized reasoning in medication traces all the way back to the beginning of the field, there has been another development to consider the requirement for AI driven medical care arrangements. Accordingly, examiners anticipate that AI will become typical in medical care sooner rather than later.

However, ultrasound diagnosis mainly depends on the knowledge and judgement of doctors, which could lead to misdiagnosis. The ResNet model was utilized to classify harmless and dangerous thyroid knobs utilizing move learning. Second, the model's emphasis on thyroid knob pictures was portrayed utilizing a heatmap. The first photographs were then recovered, and the heatmaps' featured regions were analysed. At long last, we found that the characteristics of thyroid knobs with different characteristics shifted a seriously little (P 0.05 means factual importance). All in all, the order of thyroid knob pictures utilizing our strategy is fruitful.

II. Literature Survey

1) Various approaches and previous works towards the treatment of Thyroid disease are briefly described here. Presence of nodule in the neck region is first symptom of Thyroid nodule. Thyroid growth has high incidence rates of all malignancies and has grown at the quickest rate over the past years. Ultrasonography is a accurate method for locating the nodules. There are many methods for computer-aided diagnosis that can be used to examine thyroid ultrasound images. Many of these methods will perform diagnosis by only one image instead of all the images for purpose of examining, such that it loses information regarding the thyroid nodules in the process. It is recommended to use many ultrasound images taken during an examination to diagnose thyroid nodules

2) Another approach is through attention dependent aggregation of features network, to directly combine the characters that are extracted from large set of images. It performs it in a single examining in order to improve the accuracy of recognizing cancerous nodules. These methods are used to extract and combine many features manually so that it would be difficult to perform the task.

3) Guan et al. proposed a model based on inception v3 for detecting the thyroid nodule images from the given dataset. But he used his own dataset which comprised of less training samples. Another approach is using the study regarding, computer-aided diagnostics research that automatically classifies benign, malignant nodules from Thyroid nodule images. After going through several processing, it is used to analyze and differentiate the benign and malignant thyroid nodules. Computer-aided diagnosis contains several qualities that are categorized as two groups. One is sonographic quality and the other is non-clinical quality. Both obtained from ultrasound images by using data mining and statistical techniques. classifiers that are often used in ultrasound-based (CAD) systems plays a major role. The studies that classified thyroid nodules using qualities from ultrasound pictures are then examined, and their flaws are drawn out. Additionally, reviewed and examine the techniques used in studies that classified thyroid nodules using non-clinical criteria.
.

4) Gender and racial/ethnic differences in thyroid cancer incidence also have an impact on how people can access and use healthcare. Another approach is that, demographics and tumor characteristics makes possible to look into thyroid cancer incidence rates. For this purpose, it uses data that is collected from thyroid cases nearly about 48,403 (diagnosed in 1980 up to 2005). The rates varied Based on histologic type, race and sex. At this time there were about papillary thyroid cancers nearly 39,706 were identified and were the focus of further analyses. Females experienced the fastest growth in papillary carcinoma rates. However, they did so more slowly than individuals with a starting stage and small sized tumors. Increasing rates of very tiny (2 cm) papillary carcinoma between 1992 and 1995 were responsible for half of the overall increase in papillary carcinoma rates. Medical surveillance or more accurate detection methods cannot fully explain the reported increase in papillary thyroid cancer rates. Therefore, it's critical to hunt for further justifications. All these scenarios, motivated for present algorithm that is Ultrasound image classification that is based on Thyroid nodules. This strategy has certain shortcomings even though it produced a favorable outcome. First off, there aren't enough images accessible for training and testing, and there aren't any multicenter data. Second, several algorithms may be used to classify thyroid nodules in a more thorough manner. Clinically, thyroid nodules can be separated into benign nodules as well as follicular adenomas, while thyroid nodules that are malignant can be separated into Medullary carcinomas that are papillary, follicular, etc. finally results are regarding ultrasound scans (static), which is a crucial final point, and we should think about ways to help doctors more effectively in a real clinical setting.

III. PROPOSED SYSTEM

3.1. Implementation Modules:

Resnet is the classical convolutional neural network which was the champion of 2015 ImageNet database large scale the competition in 2015. The ResNet50 architecture consists of a convolutional layer stack followed by a sequence of residual blocks. The convolutional layer stack is composed of a stack of 3×3 convolutions, batch normalization, and ReLU activations. The residual blocks each contain two 3x3 convolutions, a batch normalization layer, and a ReLU activation, followed by a shortcut connection that adds the input from the previous layer to the output of the residual block. The final layer has a global average pooling layer, a fully connected layer, and a SoftMax activation for outputting class probabilities.

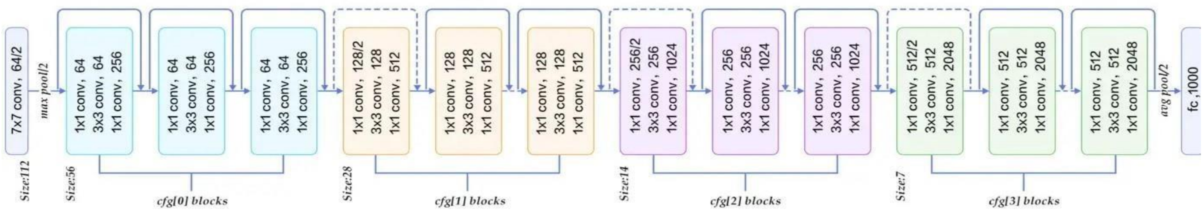


Figure 3.1 Architecture of ResNet50

The VGG19 deep learning network is composed of a total of 1,000 layers, and each layer is composed of a total of 16,000 neurons. The first layer is composed of 224 units, the second layer is composed of 1,024 units, the third layer is composed of 4,096 units, the fourth layer is composed of 16,384 units, and the fifth layer is composed of 100,000 units.

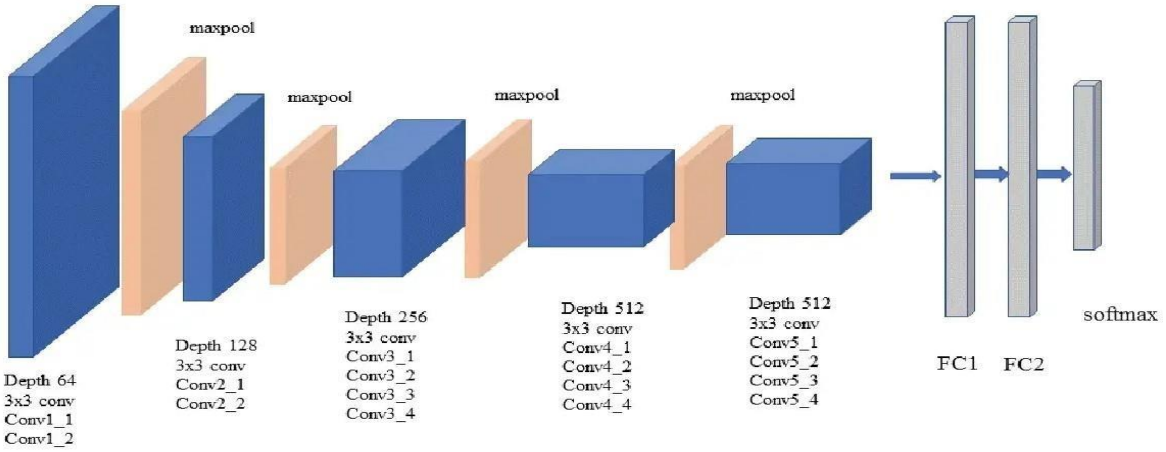


Figure 3.2 Architecture of VGG19

In this project, we were gathered 1617 ultrasound images of thyroid nodules of JPG format from the Kaggle and afterward pre-trained them for feasible and optimal classification. Among them, 1546 images of the dataset are saved in a folder and named as train which should be treated as the training set and 71 images are saved in a folder named as test which should be treated as the testing set. Between the training and testing datasets, we made sure that there was no overlapping. If any overlapping occurs, the accuracy increases so there is no point in training the dataset and that's not proper way to train the model. The principal work is as per the following. In the first place, the ResNet50 model, joined with move learning, was utilized to arrange typical(normal), benign which are harmless and malignant which are dangerous thyroid knobs. Next, a heatmap was utilized to envision the model's functioning on thyroid knob pictures. Then, the featured areas communicated by heatmaps were distinguished and broke down from distinct images utilizing Grad-CAM algorithm. At last, we observed that the qualities of thyroid knobs with various properties were very unique. In rundown, our technique is compelling in the characterization of thyroid knob pictures.

The work process of thyroid nodule image dataset classification is done with ResNet50. The Layers and the adaptive pooling depict the course of image by feature extraction and exploration with ResNet50. With the addition of layers, the key features extricated by the algorithm which turned out to be more dynamic. AUC, the acronym for area under the curve and other examination indexes were utilized to test the effect of training model classification. Additionally, a heatmap was utilized from Grad-CAM algorithm to depict the classified results, by which we segregated and examined the featured portions.

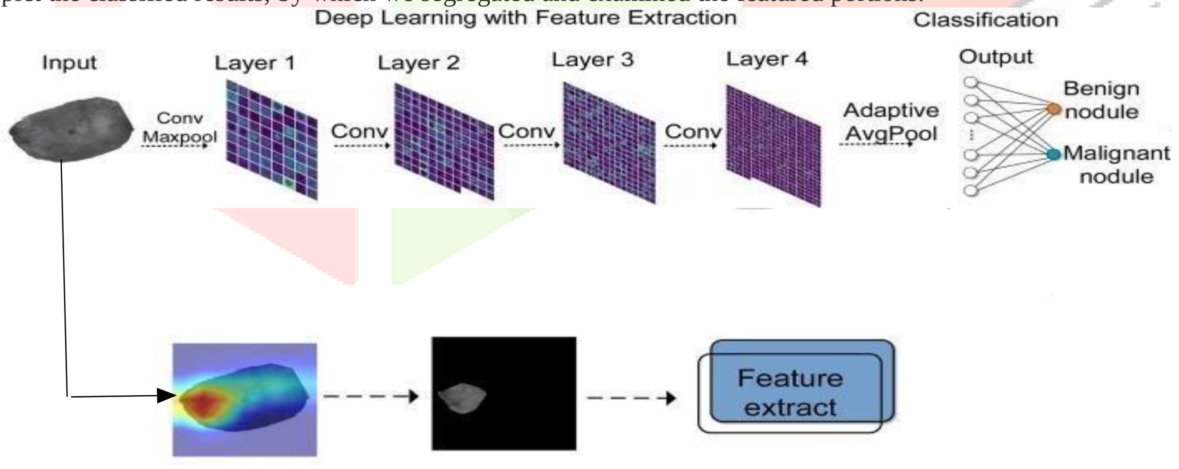
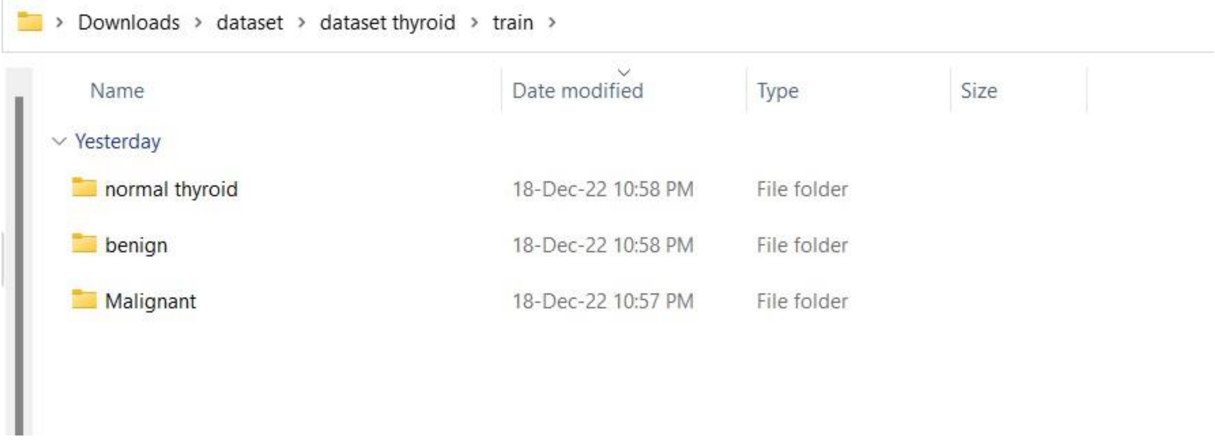


Figure 3.3 Pictorial Representation of Proposed System

3.2. Data Collection:

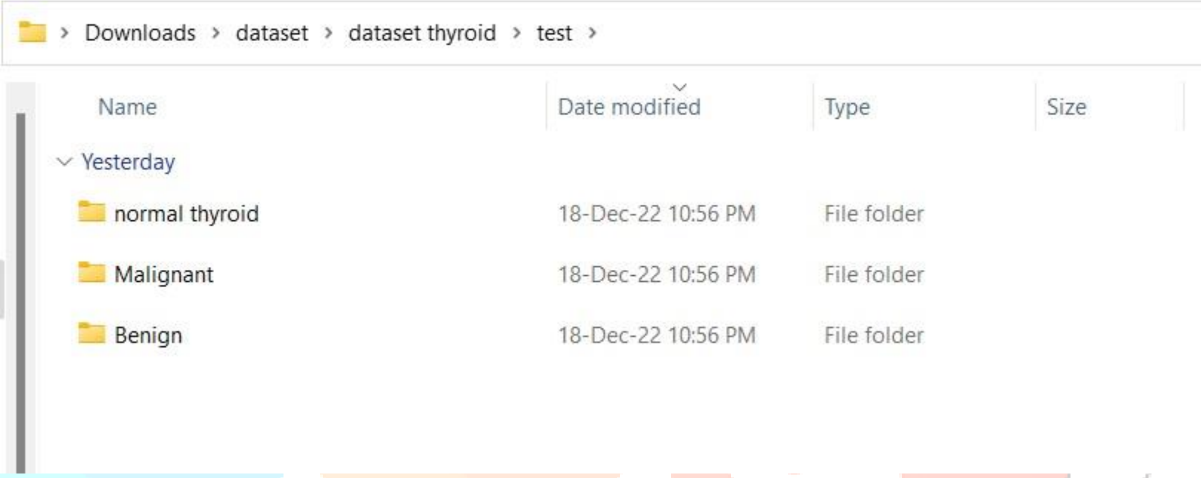
We have collected the thyroid nodules ultrasound images from the Kaggle website. The mode of this image dataset is categorical. The whole dataset is categorized into three different folders named normal, benign and malignant.



› Downloads › dataset › dataset thyroid › train ›

Name	Date modified	Type	Size
Yesterday			
normal thyroid	18-Dec-22 10:58 PM	File folder	
benign	18-Dec-22 10:58 PM	File folder	
Malignant	18-Dec-22 10:57 PM	File folder	

Figure 3.4 Training Dataset classes.



› Downloads › dataset › dataset thyroid › test ›

Name	Date modified	Type	Size
Yesterday			
normal thyroid	18-Dec-22 10:56 PM	File folder	
Malignant	18-Dec-22 10:56 PM	File folder	
Benign	18-Dec-22 10:56 PM	File folder	

Figure 3.5 Testing Dataset classes.

Some of the images from dataset that comes under the normal category of thyroid nodule ultrasound images are:

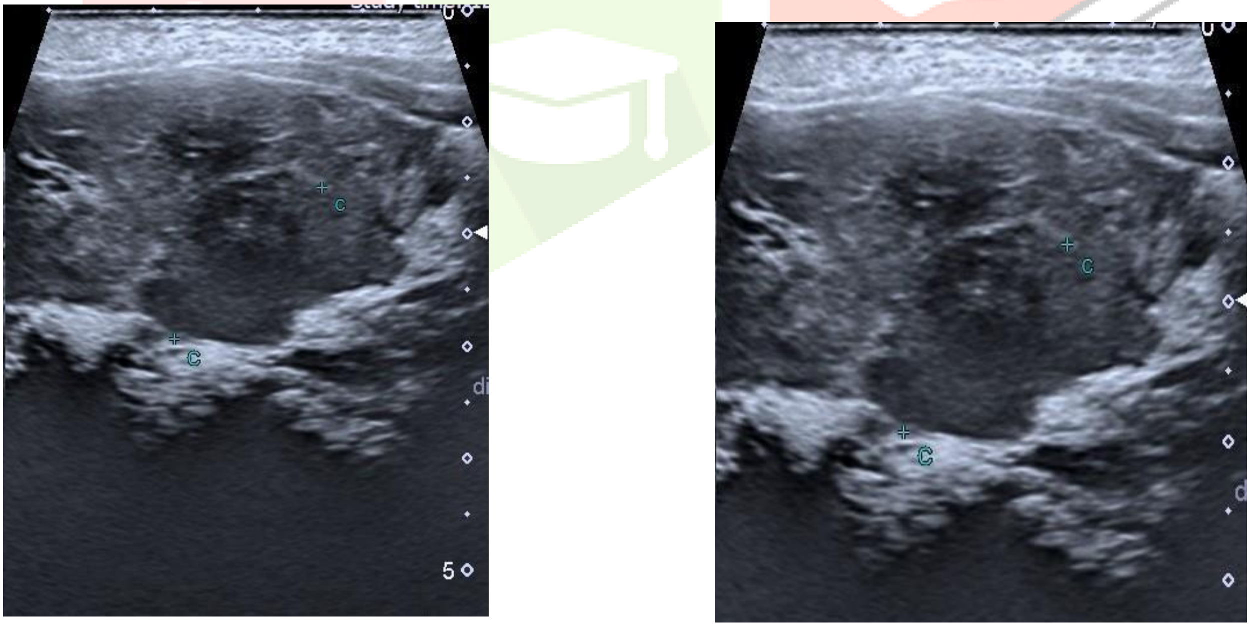


Figure 3.6 Normal thyroid nodule ultrasound images

Some of the image from the dataset that comes under the benign category of the thyroid nodules ultrasound images are:

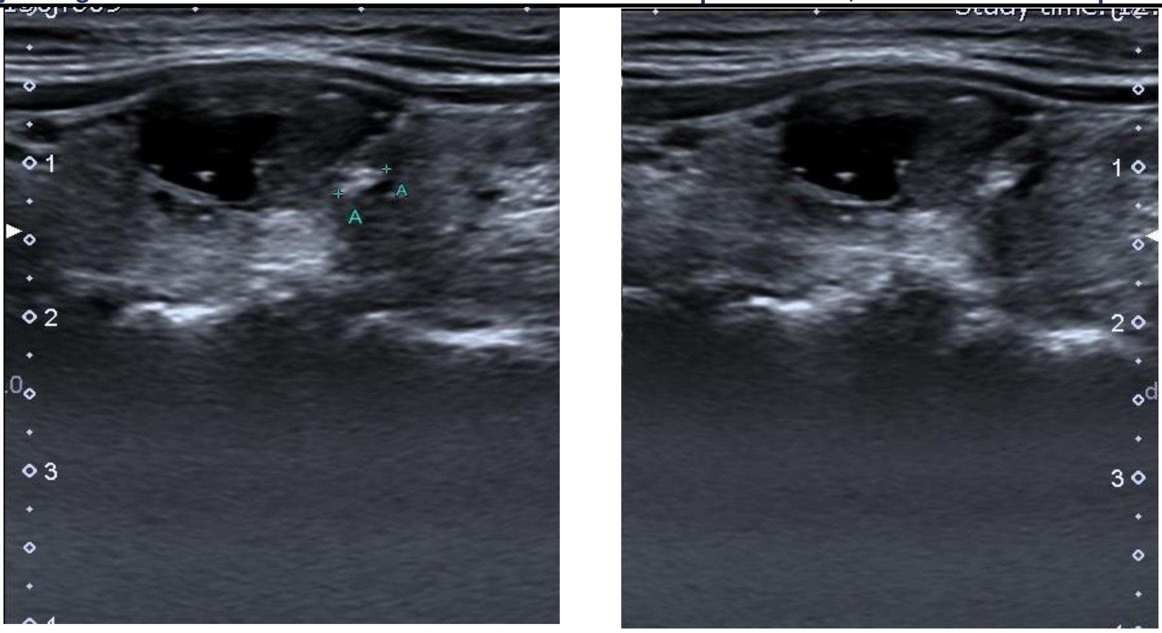


Figure 3.7 Benign thyroid nodule ultrasound images.

Some of the images from dataset that comes under the malignant category of thyroid nodules ultrasound images:

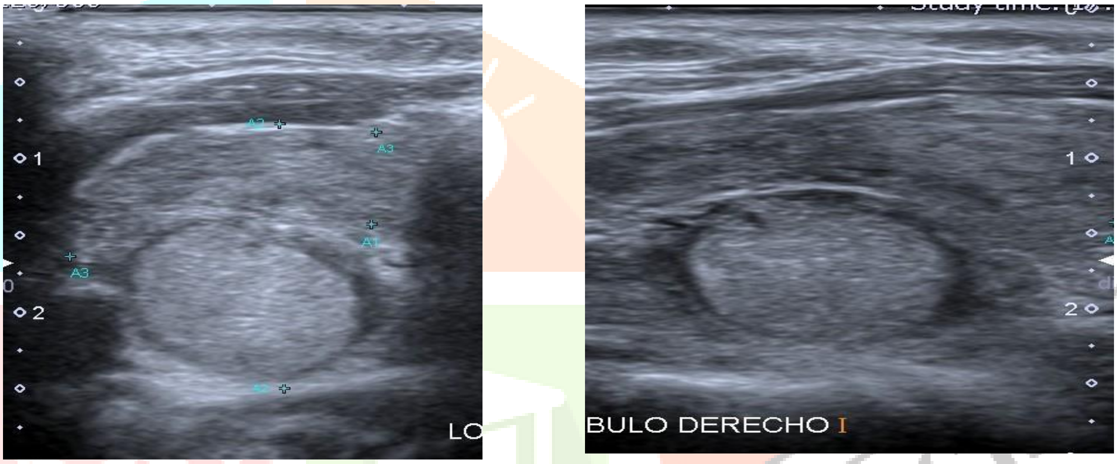


Figure 3.8 Malignant thyroid nodule ultrasound images.

3.3. Data Pre-processing:

Data Pre-processing include removal of unwanted frames, portions and text in the images for better classification using some pre-defined functions and built-in modules of Python.

3.4. Splitting of Data:

As we mentioned earlier that 1617 images were gathered from Kaggle, in those images 1546 have used for training the model and rest of the images used for testing purpose. Since the dataset mode is categorical, the splitting of data happened at the directory level by placing different categorical images in their respective labelled folder or directory and thus splitting had not done at the code implementation level.

3.5. Model Training:

Firstly, we have used VGG19 model an existing model which is not mentioned in the proposed system. In order prove that the proposed algorithm gives better accuracy compared to existing model, we have used VGG19. Then we have trained the dataset using ResNet which is a proposed algorithm.

3.6. Evaluation

In this Evaluation phase, we calculate the accuracies of different models that we have used to classify our dataset. Based on the evaluation results, we prove that the proposed model works better than the existing model.

IV. REQUIREMENT ANALYSIS

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

REQUIREMENT SPECIFICATION:

Functional Requirements

- Graphical User interface with the User.

Software Requirements

For developing the application the following are the Software Requirements:

1. Operating system : Windows 10 or 11/ Linux/ Mac
2. Coding Language : Python
3. Libraries Used : Kera's, Tensor flow, NumPy, Mat plot
4. Platform : Google Collab / Jupyter Notebook
5. Front-End : Html , CSS
6. Designing : Html, CSS, javascript

Hardware Requirements

For developing the application the following are the Hardware Requirements:

- Processor: Intel core i5 processor or higher
- RAM: 8 GB or higher
- Space on Hard Disk: 50 GB
- Core: GPU- NVIDIA if possible

V. SYSTEM DESIGN

5.1. UML Diagram:

The UML diagrams are categorized into structural diagrams, behavioural diagrams, and also interaction overview diagrams. In this project, we've used Sequence diagram which comes under behavioural diagrams to represent our proposed system.

A sequence diagram is the most normally interaction diagrams. Interaction diagram is utilized to show the intuitive way of behaving of a framework. Since picturing the communications in a framework can be an unwieldy undertaking, we utilize various kinds of interaction diagrams to catch different elements and parts of cooperation in a framework. Sequence diagrams – A grouping chart just portrays interaction between objects in a consecutive request for example the request where these connections occur. Sequence diagrams generally comprises of Objects, links, messages and boxes. Grouping charts portray how and in what request the items in a framework capability. These diagrams are broadly utilized by finance managers and programming designers to report and grasp necessities for new and existing frameworks.

There are four objects in the sequence diagram that we have drawn for our project and they are developer, Classification Model i.e., ResNet, Visualization model i.e., Grad-CAM and shape feature analysis. The line with forward arrow represents the request from an object and the line with backward represents the response from an object. The text written over these lines briefly describes about the request and response. The thin rectangular boxes under the objects represents the timespan or the duration of the process between the request and the response. Longer the rectangle, higher the duration. At top of our diagram, we have placed the objects that involve in the interaction horizontally across the X-axis so that the diagram gets formed. Generally, at the left we place the object that starts the interaction and then followed by other objects that respectively involve in the interaction to the right. Along the Y-Axis we have placed the small notes or messages that these objects request and receive response from top to bottom in increasing order. This provides the client or any other project viewer a clear insight to the flow of control over period of time.

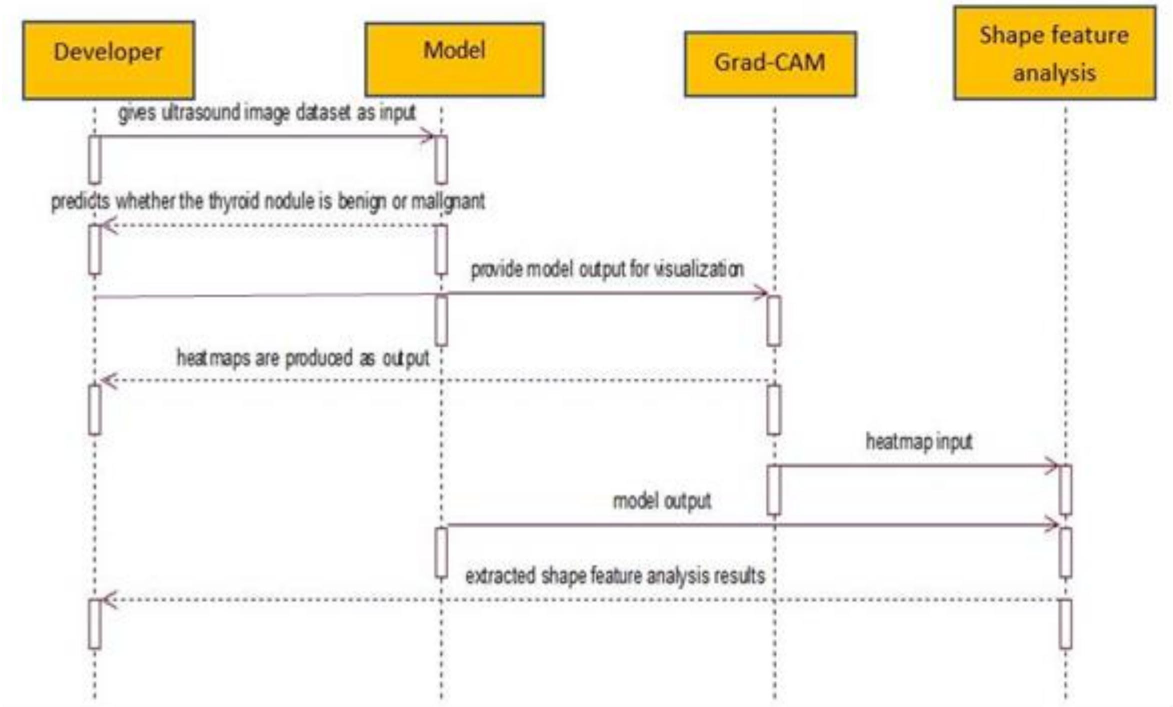

Developer
Model
Grad-CAM
Shape feature analysis
gives ultrasound image dataset as input
predicts whether the thyroid nodule is benign or malignant
provide model output for visualization
heat maps are produced as output
heatmap input
model output
extracted shape feature analysis results

Figure 5.1 Sequence diagram of the proposed system.

5.2. Data Flow Diagram:

An understanding of the information flow for any software process or system is provided by a data flow diagram (DFD). It uses preset symbols like rectangles, circles, and arrows as well as short text labels to indicate data inputs, outputs, storage places, and paths between each destination. Data flow diagrams can range from simple, one-level DFDs that briefly summarize the data handling process to more intricate, multi-level DFDs that gradually delve deeper into the process.

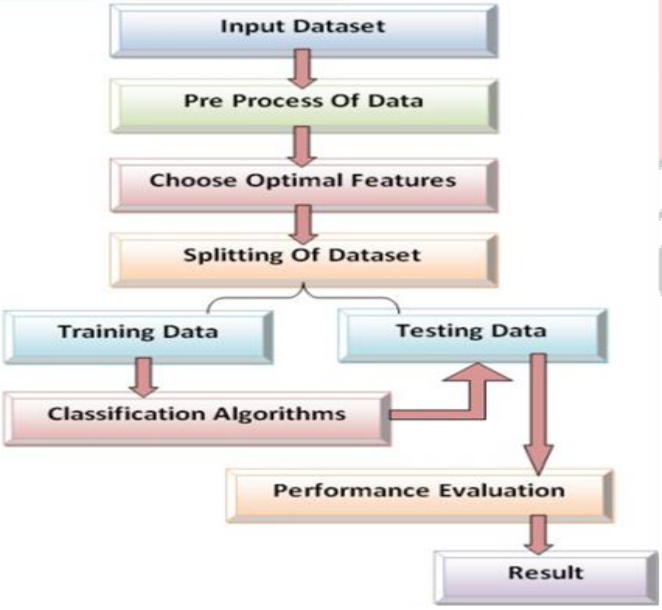


Figure 5.2 Data Flow Diagram of the proposed system.

VI. EVALUATION DETAILS

6.1. Evaluation metrics

The Evaluation is the process of measuring the quality of a model we have built by using some evaluation metrics like accuracy of the model and ROC curve. We have built two models by using VGG19 and ResNet50 respectively. The performance of both the models are evaluated based on the evaluation metrics and the best one is selected and used for further classification of thyroid nodules.

Now in an orderly we go through each and every one of the algorithms by evaluating the model using the conventional sklearn module in which we use the metrics function to evaluate the model.

6.1.1. Evaluation of VGG19

After training the model with the VGG19 algorithm the model loss and the model AUC has been as follows. The graphs are plotted by using the Matplotlib library available in Python packages. The model loss graph is plotted to analyse the loss of the model and while training the model it is ensured that the loss of the model is decreased while training.

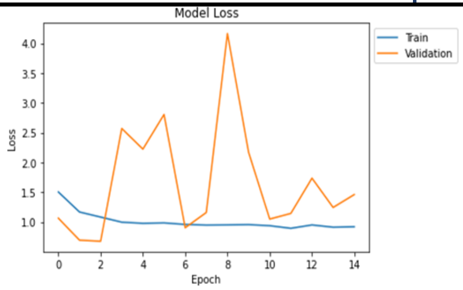


Figure 6.1 VGG19 Model Loss Graph.

The AUC of the model is plotted and as follows. The refers to the area under the ROC curve. It measures the two dimensional area underneath the graph. The model with less prediction accuracy has the less AUC and vice-versa.

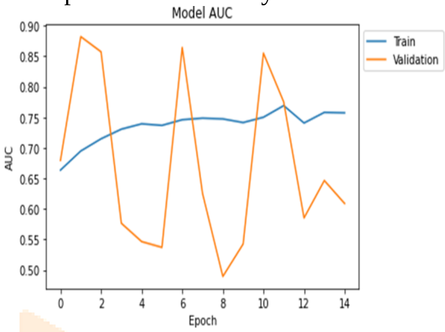
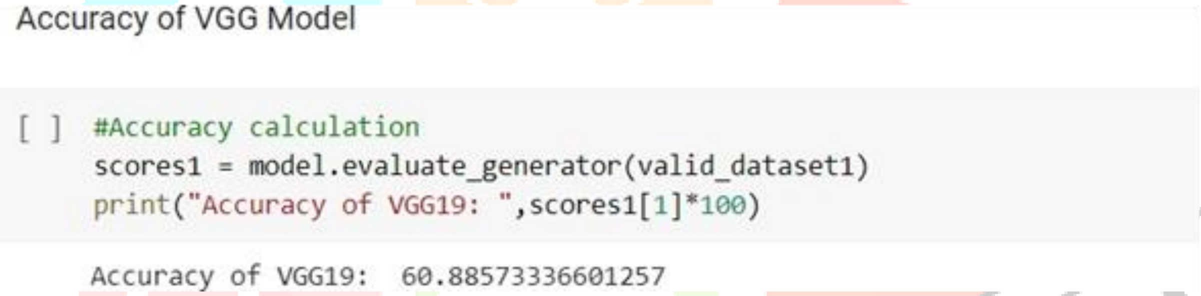


Figure 6.2 VGG19 Model AUC graph

After the training of the model, the accuracy of the trained model is calculated by providing the explicit testing data set which consists of three classes namely normal thyroid, benign and malignant.



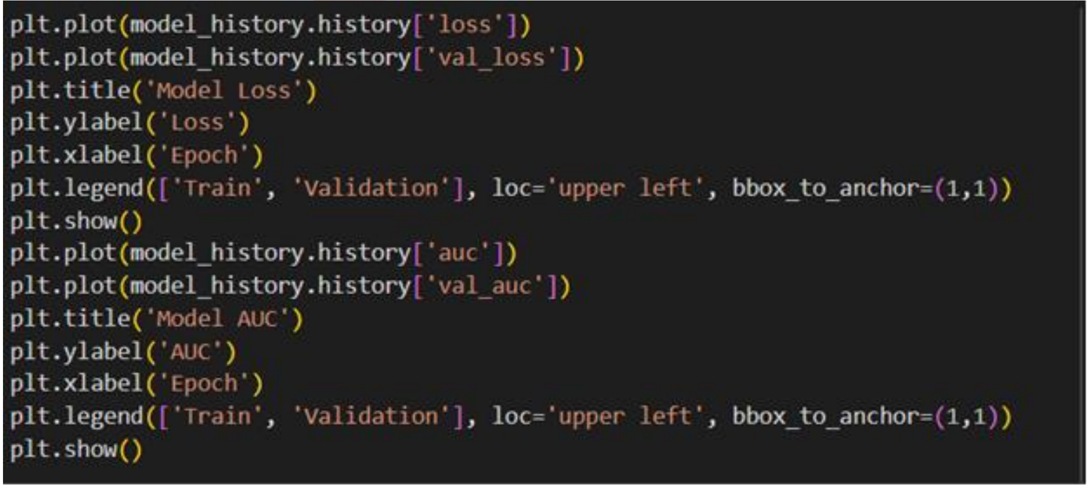
Accuracy of VGG Model

```
[ ]  #Accuracy calculation
     scores1 = model.evaluate_generator(valid_dataset1)
     print("Accuracy of VGG19: ",scores1[1]*100)

     Accuracy of VGG19:  60.88573336601257
```

Figure 6.3 VGG19 Model Accuracy

6.1.2. Evaluation of RESNET50

Resnet stands for residual network and is very efficient in classifying of images. The model employed this algorithm by inputting the ultrasound images of thyroid. The model Loss and the AUC curve are depicted below.



```
plt.plot(model_history.history['loss'])
plt.plot(model_history.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left', bbox_to_anchor=(1,1))
plt.show()
plt.plot(model_history.history['auc'])
plt.plot(model_history.history['val_auc'])
plt.title('Model AUC')
plt.ylabel('AUC')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left', bbox_to_anchor=(1,1))
plt.show()
```

Now then this imply in plotting the outcome graphs of the trained model with accuracy defining in the model.

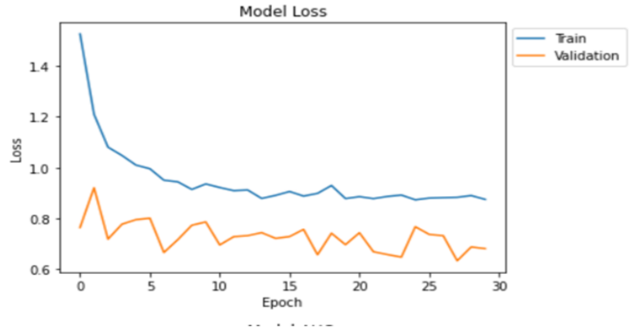


Figure 6.4 Resnet Model Loss Graph

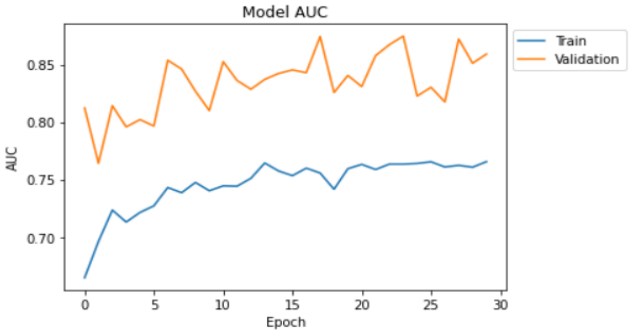
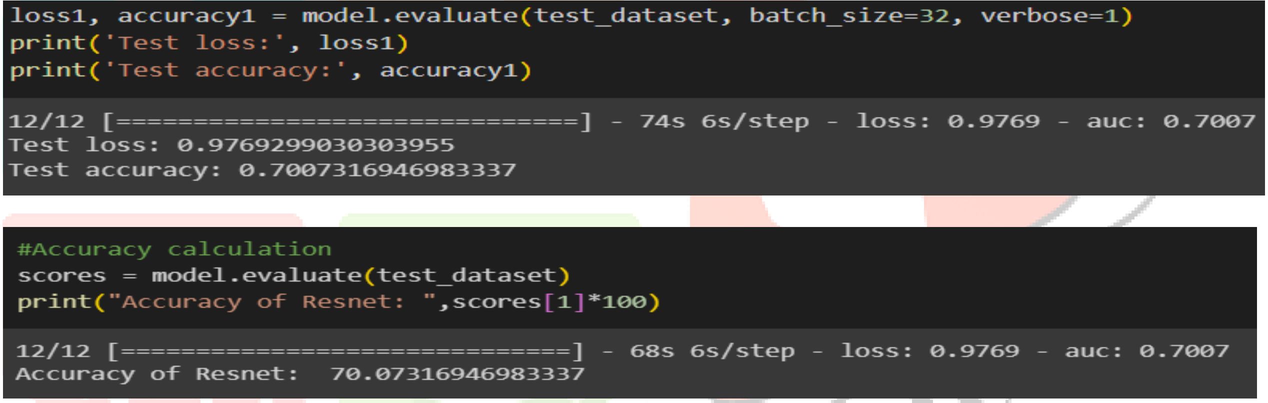


Figure 6.5 Resnet Model AUC Graph

After the training of the model, to evaluate the performance of the trained model we have calculated the accuracy the model by using the explicit data set which consists of three classes similar to the training dataset. The calculated results are shown below



```
loss1, accuracy1 = model.evaluate(test_dataset, batch_size=32, verbose=1)
print('Test loss:', loss1)
print('Test accuracy:', accuracy1)

12/12 [==============================] - 74s 6s/step - loss: 0.9769 - auc: 0.7007
Test loss: 0.9769299030303955
Test accuracy: 0.7007316946983337
```

```
#Accuracy calculation
scores = model.evaluate(test_dataset)
print("Accuracy of Resnet: ",scores[1]*100)

12/12 [==============================] - 68s 6s/step - loss: 0.9769 - auc: 0.7007
Accuracy of Resnet:  70.07316946983337
```

Figure 6.6 Resnet Model Accuracy.

6.2.Comparision of Algorithms

Here we compare all the algorithms and graph is developed based on their accuracies using MATPLOT Library which can be found in Python Libraries. Based on accuracies we have plotted graph against their algorithms.

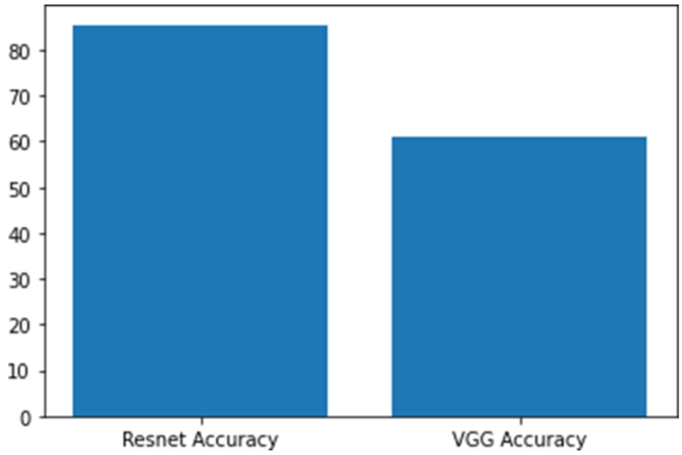


Figure 6.7 Comparison of Accuracy of models

6.3. Creating a Webpage

We have saved our trained model in the HDF5 file format using TensorFlow. This saved model can be loaded and integrated with web interface using Python. A simple website was developed where end users can upload ultrasound Thyroid nodule image to know whether it is malignant or benign, so that the layman can access the results of our project. For front-end interfaces, we have used HTML, CSS and JavaScript. Then we have loaded our trained model of HDF5 format into rendered web template using Python. The web template was rendered by flask module of python. On running this Python file, a link was generated. This generated link produces the website.

To begin the page we have to define with a HTML code so we have encoded a html code for the page and the this interface with the CSS and JavaScript to make the web page.

```
.img-preview {
    width: 256px;
    height: 256px;
    position: relative;
    border: 5px solid #F8F8F8;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
    margin-top: 1em;
    margin-bottom: 1em;
}

.img-preview>div {
    width: 100%;
    height: 100%;
    background-size: 256px 256px;
    background-repeat: no-repeat;
    background-position: center;
}

input[type="file"] {
    display: none;
}

.upload-label{
    display: inline-block;
    padding: 12px 30px;
    background: #39D2B4;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

.upload-label:hover{
    background: #34495E;
    color: #39D2B4;
}

.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #3498db; /* Blue */
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
```

We have built the CSS code as the following shown. Now interface with Java script code to ensure the webpage is built with the site. So, create a folder name ‘static’ in the folder “final project”, then make new folders in static with CSS and JavaScript. Then import the code of each in the assigned folder, use VScode as the IDE for the codes. The Java Script is as follows

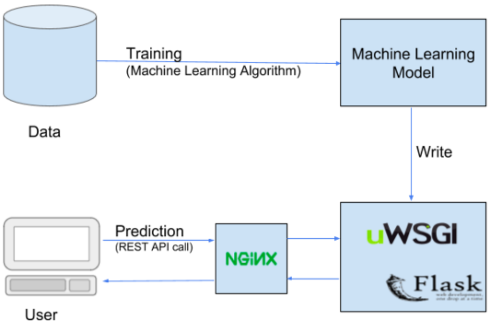
```
$(document).ready(function () {
    // Init
    $('.image-section').hide();
    $('.loader').hide();
    $('#result').hide();
    $('#ft').hide();

    // Upload Preview
    function readURL(input) {
        if (input.files && input.files[0]) {
            var reader = new FileReader();
            reader.onload = function (e) {
                $('#imagePreview').css('background-image', 'url(' + e.target.result + ')');
                $('#imagePreview').hide();
                $('#imagePreview').fadeIn(650);
            }
            reader.readAsDataURL(input.files[0]);
        }
    }
    $("#imageUpload").change(function () {
        $('.image-section').show();
        $('#btn-predict').show();
        $('#result').text('');
        $('#result').hide();
        readURL(this);
    });

    // Predict
    $('#btn-predict').click(function () {
        var form_data = new FormData($('#upload-file')[0]);

        // Show loading animation
        $(this).hide();
        $('.loader').show();

        // Make prediction by calling api /predict
        $.ajax({
            type: 'POST',
            url: '/predict',
            data: form_data,
            contentType: false,
            cache: false,
            processData: false,
            async: true,
            success: function (data) {
                // Get and display the result
                $('.loader').hide();
                $('#result').fadeIn(600);
                $('#result').text(' Result:  ' + data);
                console.log('Success!');
                $('#ft');
            },
        });
    });

});
```

When a data scientist/machine learning engineer develops a machine learning model using Scikit-Learn, TensorFlow, Keras, PyTorch etc, the ultimate goal is to make it available in production. Often times when working on a machine learning project, we focus a lot on Exploratory Data Analysis(EDA), Feature Engineering, tweaking with hyper-parameters etc. But we tend to forget our main goal, which is to extract real value from the model predictions.

Deployment of machine learning models or putting models into production means making your models available to the end users or systems. However, there is complexity in the deployment of machine learning models. This post aims to make you get started with putting your trained machine learning models into production using Flask API.

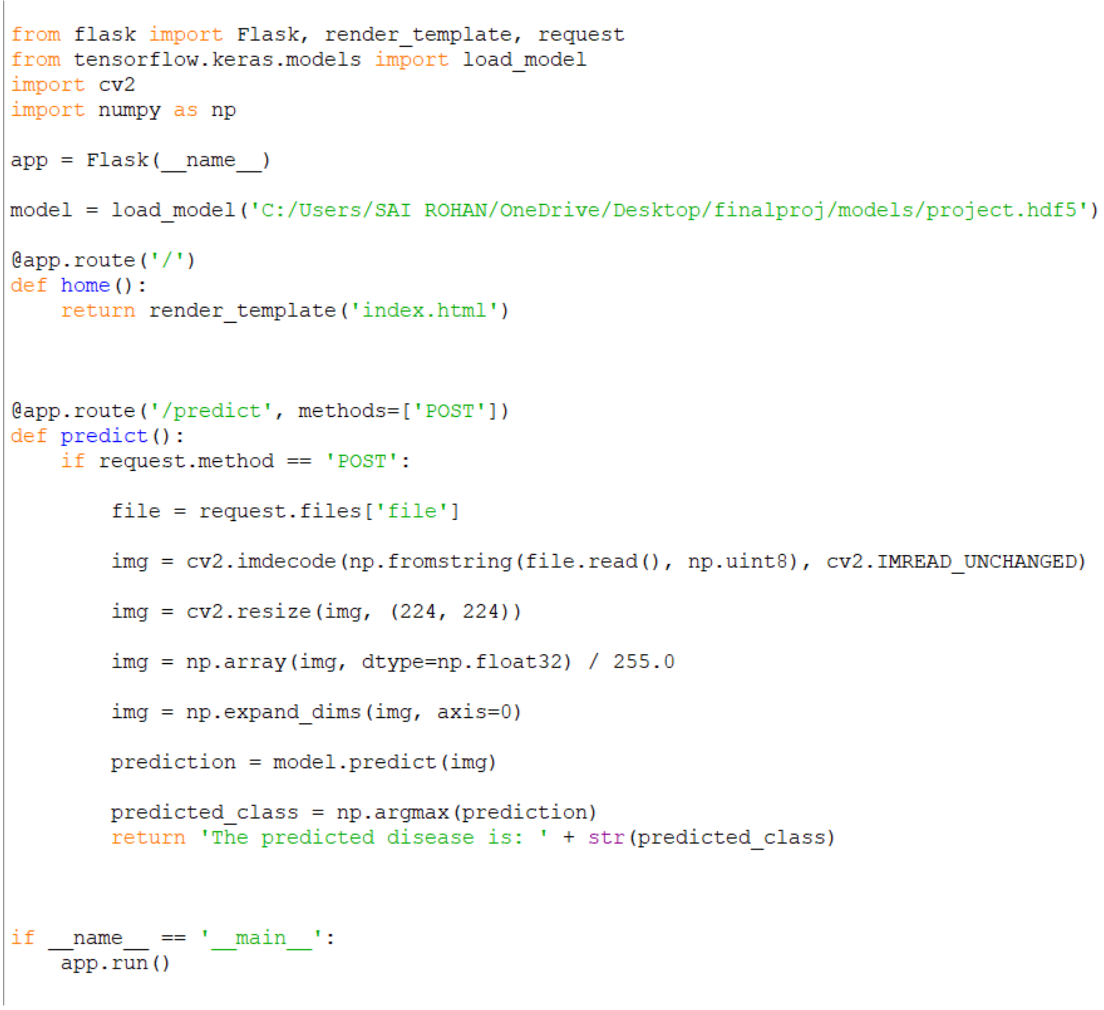
6.4.Project Structure

This project has four parts :
1.model.py — This contains code for the machine learning model .
2.app.py — This contains Flask APIs that receives sales details through GUI or API calls, computes the predicted value based on our model and returns it.
3.request.py — This uses requests module to call APIs defined in app.py and displays the returned value.
4.HTML/CSS — This contains the HTML template and CSS styling.



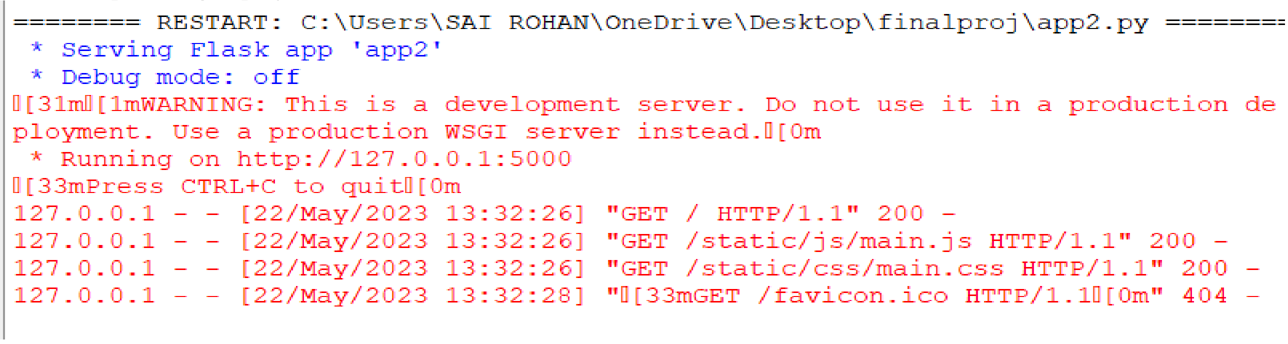
Pipeline for deployment of a Machine Learning model

As of now we have completed code for the CSS and Javascript to build a website. Now to deploy the website with the trained model we have a write a python code using flask module. Write the code for deploying trained model and website name it as "app.py" and keep it in 'final project' folder.



```
from flask import Flask, render_template, request
from tensorflow.keras.models import load_model
import cv2
import numpy as np

app = Flask(__name__)

model = load_model('C:/Users/SAI ROHAN/OneDrive/Desktop/finalproj/models/project.hdf5')

@app.route('/')
def home():
    return render_template('index.html')



@app.route('/predict', methods=['POST'])
def predict():
    if request.method == 'POST':

        file = request.files['file']

        img = cv2.imdecode(np.fromstring(file.read(), np.uint8), cv2.IMREAD_UNCHANGED)

        img = cv2.resize(img, (224, 224))

        img = np.array(img, dtype=np.float32) / 255.0

        img = np.expand_dims(img, axis=0)

        prediction = model.predict(img)

        predicted_class = np.argmax(prediction)
        return 'The predicted disease is: ' + str(predicted_class)



if __name__ == '__main__':
    app.run()
```

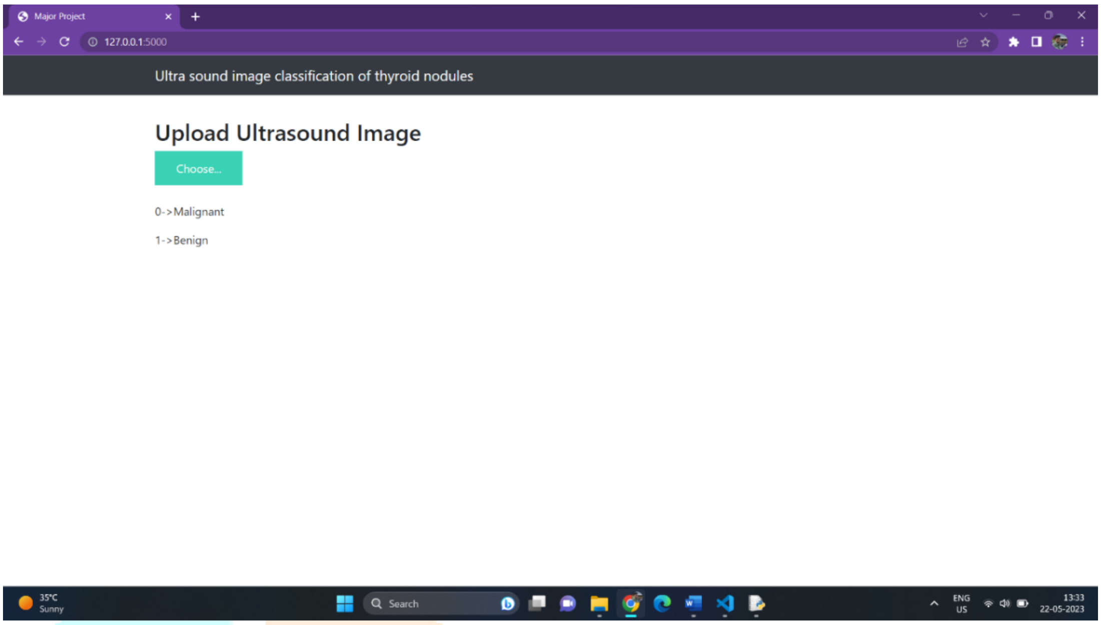
Now run the code in the python IDE it will create a local host server, as we know it is temporary server now go to the server link provided as output using deployment.so then the website which is created is shown.



```
======== RESTART: C:\Users\SAI ROHAN\OneDrive\Desktop\finalproj\app2.py ========
 * Serving Flask app 'app2'
 * Debug mode: off
[31m[1mWARNING: This is a development server. Do not use it in a production de
ployment. Use a production WSGI server instead.[0m
 * Running on http://127.0.0.1:5000
[33mPress CTRL+C to quit[0m
127.0.0.1 - - [22/May/2023 13:32:26] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [22/May/2023 13:32:26] "GET /static/js/main.js HTTP/1.1" 200 -
127.0.0.1 - - [22/May/2023 13:32:26] "GET /static/css/main.css HTTP/1.1" 200 -
127.0.0.1 - - [22/May/2023 13:32:28] "[33mGET /favicon.ico HTTP/1.1[0m" 404 -
```

VII. RESULTS:

The website has project title followed by a button which redirects to choose the image to upload. Then followed by another button called "predict" which shows the final result that whether the uploaded thyroid nodule image is malignant or benign.



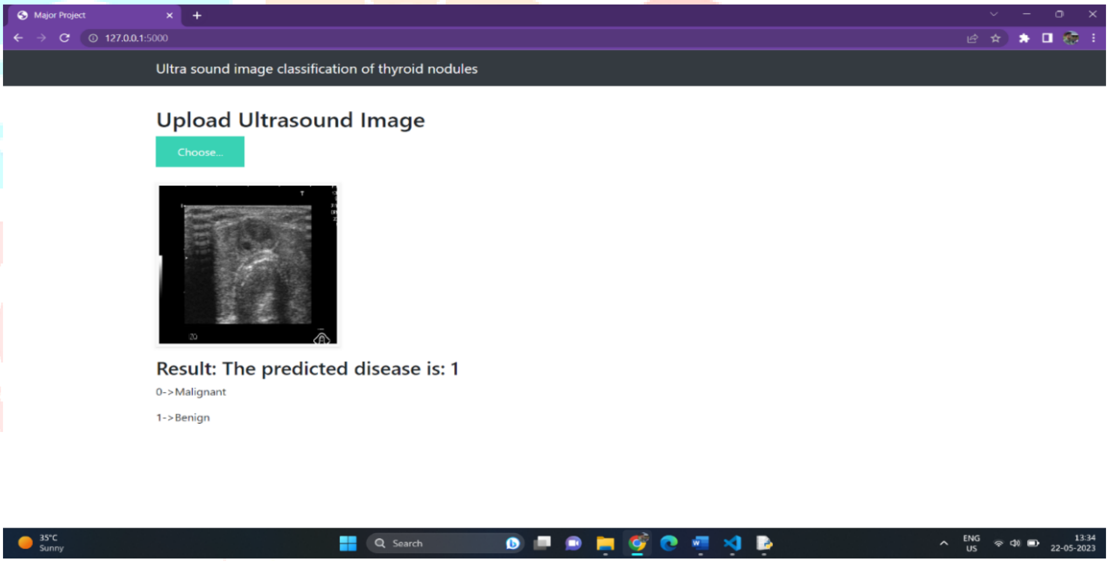
Ultra sound image classification of thyroid nodules

Upload Ultrasound Image

0->Malignant

1->Benign

Figure 7.1 The interface of website



Ultra sound image classification of thyroid nodules

Upload Ultrasound Image

Result: The predicted disease is: 1

0->Malignant

1->Benign

Figure 7.2 Prediction of Thyroid nodule in website

VIII. CONCLUSION

A lifelong medical problem thyroid, which is an abnormal development of thyroid cells, requires continual management. Both the doctor and the patient gains from finding the nodules and identifying them as benign or cancerous. Here is the application of our project, Deep Learning-based Ultrasound Image Classification of Thyroid Nodules. This project can assist the physician in determining the thyroid nodule's level of malignancy and in providing the patient with better treatment recommendations. A doctor can assess whether there are numerous nodules. When using fine-needle aspiration technology, it serves as a guidance.

Thyroid is a very common disease in the middle-aged people especially targeting women at higher rate. There are many traditional approaches followed by doctors to detect the thyroid nodules in the body. Approaches like Fine Needle Aspiration can be used to detect thyroid in the body, but it is very harmful for the body. The alternate and safe approach that can be followed is to examine the ultrasound images of the patient. A deep learning framework is proposed to classify the thyroid nodules by taking the ultra sound images as input.

The proposed, enhanced, and normalized images or dataset was provided to the Resnet model and VGG model as a training, testing data. We have compared the accuracies of both model after training and the higher accuracy model is deployed for classification of ultrasound images of thyroid nodules.

IX. Future enhancement

Knowledge is divine and there is no end for new inventions and discoveries, only getting to know, how cancerous a specific thyroid nodule isn't sufficient, proper care must also be taken to cure it as well and monitoring thyroid nodule growth is also important. The interpretability of the ultrasound images is very poor and it would be very challenging for doctors to believe the output of the model and continue with further diagnosis.

To overcome this problem, Grad-CAM is used to visualize the images and produce the heatmaps in which the regions in the ultrasound images are highlighted. By visualizing the heatmap the doctors can diagnose the level of thyroid disease in an easy way as compared to earlier.

Future versions of this model may include a website where regular people could upload their ultrasound images and see if they have thyroid or they are in safe zone. It may be improved to be more user-friendly and made for users as well as hospitals for thyroid detection. We'll be working to boost efficiency while also focusing on getting the most accurate detection results possible.

X. References

1. Camargo RY, Tomimori EK. [Usefulness of Ultrasound in the Diagnosis and Management of Well-Differentiated Thyroid Carcinoma]. Arq Bras Endocrinol Metabol (2007) 51:783–92. doi: 10.1590/S0004-2730200 7000500016

2. Enewold L, Zhu K, Ron E, Marrogi AJ, Stojadinovic A, Peoples GE, et al. Rising Thyroid Cancer Incidence in the United States by Demographic and Tumor Characteristics, 1980-2005. Cancer Epidemiol Biomarkers Prev (2009) 18:784–91. doi: 10.1158/1055-9965.EPI-08-0960

3. Russ G, Leboulleux S, Leenhardt L, Hegedüs L. Thyroid Incidentalomas: Epidemiology, Risk Stratification With Ultrasound and Workup. Eur Thyroid J (2014) 3:154–63. doi: 10.1159/000365289