RECIPE RECOGNITION USING DEEP LEARNING

Developed By: D. Anusha, J. Teja, Ch. Vikram, MD. Feroz Uddin

CONTENTS **PAGE NO'S**

1. INTRODUCTION

1.1 OVERVIEW

Food is an essential component of our individual and social life. Eating habits have a direct impact on our health and wellbeing, while ingredients, flavour and cooking recipes shape specific cuisines that are part of our personal and collective cultural identities. But there are also interesting applications of automatic food recognition to self-service restaurants and dining halls. For instance, accurate detection and segmentation of the different food items in a food tray can be used for monitoring food intake and nutritional information, and automatic billing to avoid the cashier bottleneck in self-service restaurants. This work deals with the problem of automated recognition of a photographed cooking dish and the subsequent output of the appropriate recipe.

In this project, we focus on applications of automatic food recognition and identify the recipe in food by using convolutional neural networks. And this model will classify images into food categories and to output a matching recipe.

Keyword - Calorie estimation, Convolutional Neural Network (CNN), Deep Learning technique, Recipe recognition classification.

1.2 PURPOSE

Recipes are used for training back of house staff. It provide consistency in the production of menu items. Provides food cost control. They provide knowledge for front of the house staff as a sales tool and to help consumers with dietary concerns and allergies. They should be handed out to each back of the house line cook in a booklet specific to their station. The measured ingredients in a recipe not only produce consistent food but control your food cost and profit. Recipes reduce waste because a cook is prepping exactly what is needed to produce the menu items. The recipe provide portion control which is a major factor in food cost control and profit.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

A food data set for recipe generally contains main attributes such as images which are labeled according to their classes which are mainly used for the purpose of classier training models. In the research the main purpose of data set is divided into three groups: general food recognition, recipe analysis/retrieval, and restaurant based recognition[5]. General food recognition datasets are primarily used to train food classifiers and generally consist of images and the accompanying (food) class labels. From early datasets with a small number of cuisine-specific photos to more enormous datasets with a considerably more significant number of images per class and a more comprehensive range of foods and cuisines, these datasets have grown to contain more food classes. These more extensive datasets are ideal for fine-tuning deep CNNs, resulting in cutting-edge food identification.

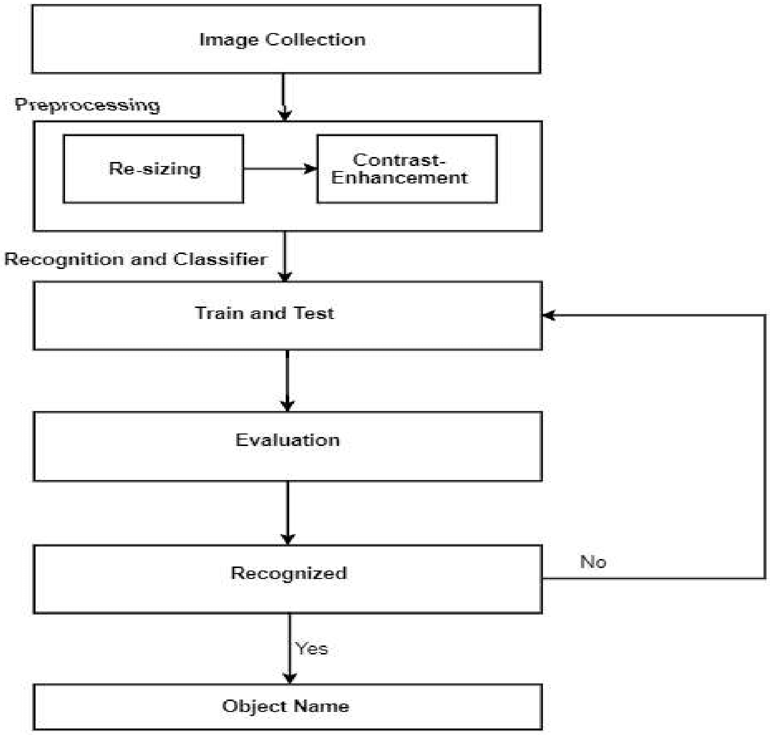
2.1PROPOSED SOLUTION

The objective of proposed application is to suggest food recipes on the basis of food items recognized in the photographs taken. The system is based on multiple object detection and it fetches recipes based on the ingredients detected in the image. The proposed system architecture is depicted datasets given food items will be used for recognition and Kaggle dataset as recipe database on server side. The proposed system consists of four phases: Image pre-processing, Feature selection and extraction, Classification and Output, shows a diagram of all the layers of the CNN model that are interconnected. User interacts with the UI (User Interface) to upload the image as input. Uploaded image is analyzed by the model which is integrated. Once model analyses the uploaded image, the model predicts the food item and the recipe is showcased on the UI.

3. THEORITICAL ANALYSIS

Finding new recipes daily with the ingredients in hand is a difficult task that leads to people looking at different recipe websites and cookbooks. But it is often not possible to find the recipe one is looking for with the constraint of ingredients. The objective of this paper is to build a web application so as to cater to this specific problem faced by people. The proposed system takes ingredients as an input and gives machine generated recipes with the help of deep learning to the user. The recipe generated model will consist of title, ingredients and instruction reflecting the dataset used to prepare the model for deep learning. The system also consists of interactive features in the user interface enabling the user to save and access the generated recipe later on.

3.1 BLOCK DIAGRAM



PROJECT WORKFLOW

1. Install required packages and libraries.
2. Understanding the data.
3. Model Building.
4. Application Building.
5. Final UI.

3.2 HARDWARE AND SOFTWARE DESIGN

For running a deep learning model on the system you need a system with minimum of 16 GB RAM in it and you require a good processor for high performance of the model.

In the list of **software requirements** you must have:

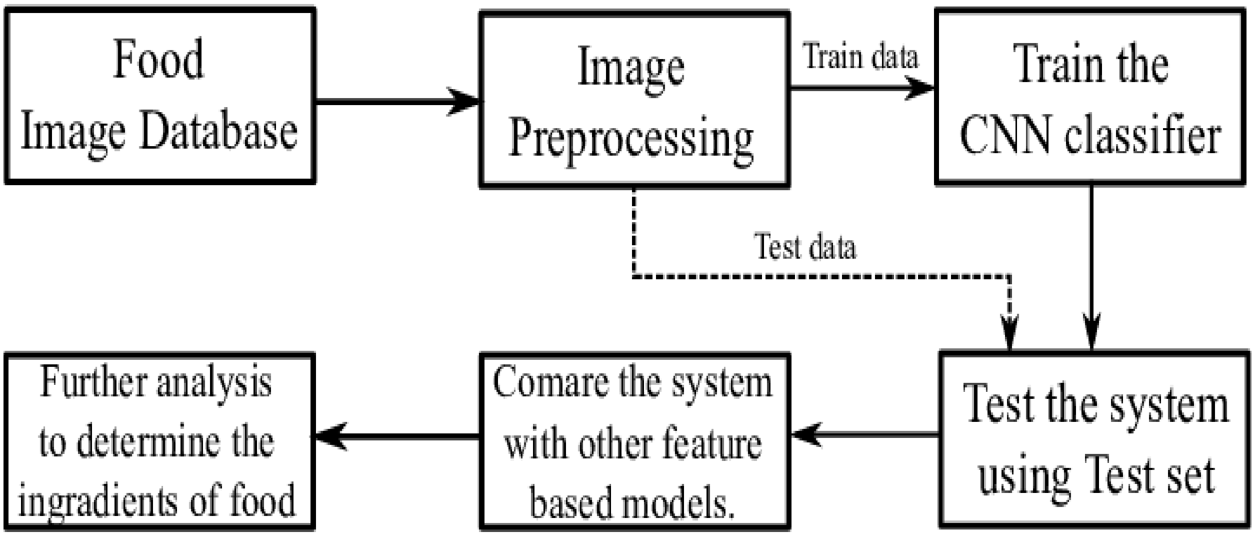
- Jupyter Notebook for programming, which can be installed by Anaconda IDE.
- Python packages.
- A better software for running the html and css files for application building phase e.g. spyder.

4. EXPERIMENTAL INVESTIGATIONS

DATA PREPROCESSING

- Import the ImageDataGenerator library
- Configure ImageDataGenerator class.
- Apply ImageDataGenerator functionality to Trainset and Testset

5. FLOWCHART


Food Image Database
Image Preprocessing
Train data
Train the CNN classifier
Test data
Test the system using Test set
Comare the system with other feature based models.
Further analysis to determine the ingradients of food

Project Flow

- User interacts with the UI (User Interface) to upload the image as input
- Uploaded image is analyzed by the model which is integrated
- Once model analyses the uploaded image, the model predicts the food item and the recipe is showcased on the UI.

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

Data Collection

- Collect the dataset or Create the dataset

Data Preprocessing

- Import the ImageDataGenerator library
- Configure ImageDataGenerator class.
- Apply ImageDataGenerator functionality to Trainset and Testset

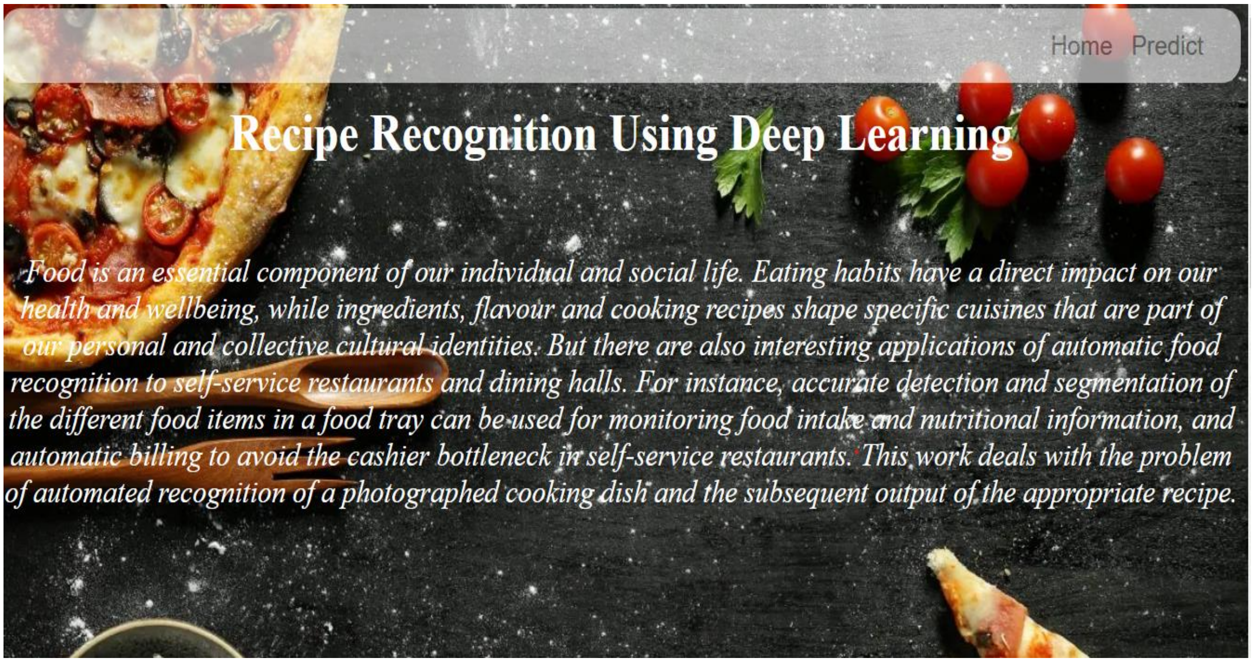
Model Building

- Import the model building Libraries
- Initializing the model
- Adding Input Layer
- Adding Hidden Layer
- Adding Output Layer
- Configure the Learning Process
- Training the model
- Save the Model
- Test the Model

Application Building

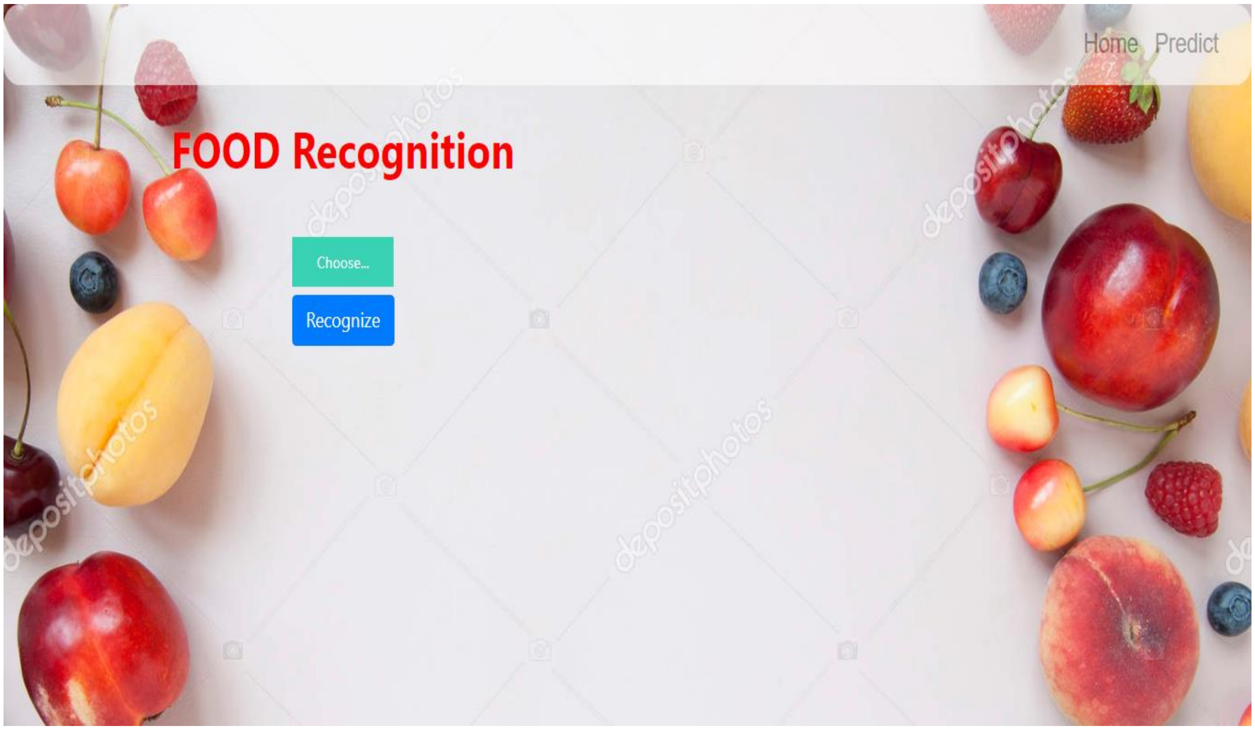
- Create an HTML file
- Build Python Code

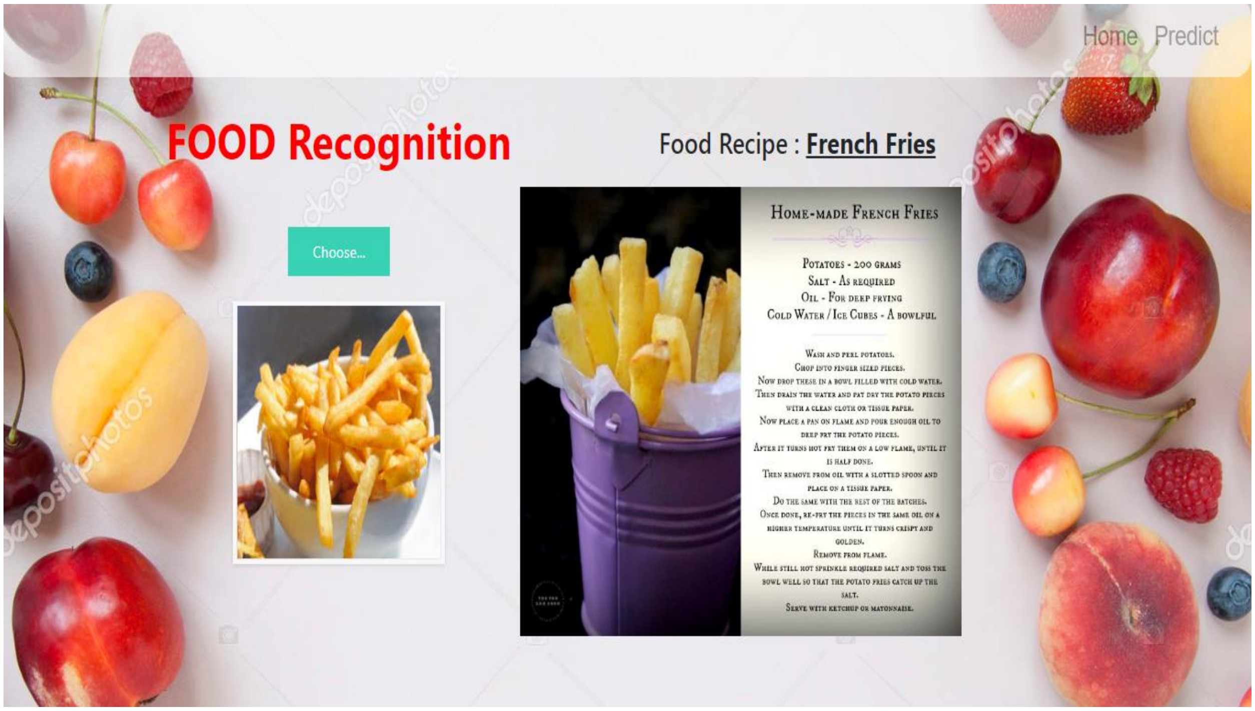
6. RESULT



Recipe Recognition Using Deep Learning

Food is an essential component of our individual and social life. Eating habits have a direct impact on our health and wellbeing, while ingredients, flavour and cooking recipes shape specific cuisines that are part of our personal and collective cultural identities. But there are also interesting applications of automatic food recognition to self-service restaurants and dining halls. For instance, accurate detection and segmentation of the different food items in a food tray can be used for monitoring food intake and nutritional information, and automatic billing to avoid the cashier bottleneck in self-service restaurants. This work deals with the problem of automated recognition of a photographed cooking dish and the subsequent output of the appropriate recipe.





Food Recipe : **French Fries**

HOME-MADE FRENCH FRIES

POTATOES - 200 GRAMS
SALT - AS REQUIRED
OIL - FOR DEEP FRYING
COLD WATER / ICE CUBES - A BOWLFUL

WASH AND PEEL POTATOES.
CHOP INTO FINGER SIZED PIECES.
NOW DROP THESE IN A BOWL FILLED WITH COLD WATER.
THEN DRAIN THE WATER AND PAT DRY THE POTATO PIECES WITH A CLEAN CLOTH OR TISSUE PAPER.
NOW PLACE A PAN ON FLAME AND POUR ENOUGH OIL TO DEEP FRY THE POTATO PIECES.
AFTER IT TURNS HOT FRY THEM ON A LOW FLAME, UNTIL IT IS HALF DONE.
THEN REMOVE FROM OIL WITH A SLOTTED SPOON AND PLACE ON A TISSUE PAPER.
DO THE SAME WITH THE REST OF THE BATCHES.
ONCE DONE, RE-FRY THE PIECES IN THE SAME OIL ON A HIGHER TEMPERATURE UNTIL IT TURNS CRISPY AND GOLDEN.
REMOVE FROM FLAME.
WHILE STILL HOT SPRINKLE REQUIRED SALT AND TOSS THE BOWL WELL SO THAT THE POTATO FRIES CATCH UP THE SALT.
SERVE WITH KETCHUP OR MAYONNAISE.

7. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- CNNs do not require human supervision for the task of identifying important features.
- They are very accurate at image recognition and classification.
- Weight sharing is another major advantage of CNNs.
- Convolutional neural networks also minimize computation in comparison with a regular neural network.
- CNNs make use of the same knowledge across all image locations.

DISADVANTAGES:

- They fail to encode the position and orientation of objects. They have a hard time classifying images with different positions.
- A lot of training data is needed for the CNN to be effective.
- CNNs tend to be much slower because of operations like maxpool.
- In case the convolutional neural network is made up of multiple layers, the training process could take a particularly long time if the computer does not have a good GPU.
- Convolutional neural networks will recognize the image as clusters of pixels which are arranged in distinct patterns. They don’t understand them as components present in the image.

8. APPLICATIONS

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area. Adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer.

The Fully-Connected layer is learning a possibly non-linear function in that space. Now that we have converted our input image into a suitable form for our Multi-Level Perceptron, we shall flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and back propagation applied to every iteration of training. Over a series of epochs, the model is able to distinguish between dominating and certain low-level features in images and classify them using the Softmax Classification technique.

9. CONCLUSION

In this research study, the Convolutional Neural Network, a Deep learning technique is used to classify the food images into their respective classes. As far as the future enhancement is concerned, the task of classification can be improved by removing noise from the dataset. The same research can be carried out on larger dataset with a greater number of classes and a greater number of images in each class, as larger dataset improves the accuracy by learning more features and reduces the loss rate. The weights of the model can be stored and used to design a web app or mobile app for image classification and in addition calories extraction of the classified food. The proposed system would be improved when the calorie measurement will be done for multi-food and complex food items which would help people who will be using our application to apprehend deeply the complexities of food. Additionally, the expansion of dataset with more variety of food types which will improve the result and accuracy of the system. As fast response is one of the most crucial elements these days, the computational part of the model which takes longer time can be offloaded to the cloud. With the use of cloud, a model which takes less time to produce the result by performing large computations can be obtained. The work started with collecting real-time food images from various sources.

10. FUTURE SCOPE

The convolution neural network-based model is trained over large amount of food images, which enhances your model capability to get required features quickly. In the result analysis, the accuracy of the training dataset of images obtained is about 84%. For the future However, a massive-scale dataset is needed to train the CNN and even a well-trained network cannot have a segmentation accuracy of 100%, which will en-large the error rate of recognition. Therefore, we can build a larger dataset which includes different food images to get a better result. The need to have a system that measures daily food intake for healthy diet is crucial due to the insufficient knowledge of diet and calorie requirements. In addition, correct food recognition is considered challenge. Hence, we proposed a measurement method to estimate the number of calories from different food images by measuring the features such as colour of the food from the image. We have efficaciously implemented a robust system for the correct identification of the food item.

11. BIBILOGRAPHY

1. Rajayogi, J.R., Manjunath, G. and Shobha, G., 2019, December. Indian Food Image Classification with Transfer Learning. In 2019 4th Inter-national Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS) (Vol. 4, pp. 1-4). IEEE.

2. Reddy, V.H., Kumari, S., Muralidharan, V., Gigoo, K. and Thakare, B.S., 2019, May. Food Recognition and Calorie Measurement using Image Processing and Convolutional Neural Network. In 2019 4th In-ternational Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT) (pp. 109-115). IEEE.

3. Subhi, M.A. and Ali, S.M., 2018, December. A deep convolutional neural network for food detection and recognition. In 2018 IEEE-EMBS conference on biomedical engineering and sciences (IECBES) (pp. 284-287). IEEE.

4. Burkapalli, V.C. and Patil, P.C., TRANSFER LEARNING: INCEPTION-V3 BASED CUSTOM CLASSIFICATION APPROACH FOR FOOD IMAGES.

5. Fruit Recognition and its Calorie Measurement: An Image Processing Approach ManpreetkourBasantsingh Sardar1, Dr. Sayyad D. Ajij2 (2016)

12. APPENDIX

```
import os
import numpy as np #used for numerical analysis
from flask import Flask,request,render_template# Flask-It is our framework which we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.
#render_template- used for rendering the html pages
from tensorflow.keras.models import load_model#to load our trained model
from tensorflow.keras.preprocessing import image
app=Flask(__name__)#our flask app
model=load_model('food.h5')#loading the model
@app.route("/") #default route
def upload_file():
    return render_template("RR.html")#rendering html page
@app.route("/about") #route about page
def upoad_file1():
    return render_template("RR.html")#rendering html page
@app.route("/upload") # route for info page
def upload_file2():
    return render_template("RRP.html")#rendering html page
#@app.route("/upload") # route for uploads
```

```
#def test():
    #return render_template("index.html")#rendering html page
@app.route("/predict",methods=["GET","POST"]) #route for our prediction
def upload():
    if request.method=='POST':
        f=request.files['file'] #requesting the file
        basepath=os.path.dirname('__file__')#storing the file directory
        filepath=os.path.join(basepath,"uploads",f.filename)#storing the file in uploads folder
        f.save(filepath)#saving the file
        img=image.load_img(filepath,target_size=(64,64)) #load and reshaping the image
        x=image.img_to_array(img)#converting image to array
        x=np.expand_dims(x,axis=0)#changing the dimensions of the image
        pred=model.predict(x)
        pred=np.argmax(pred,axis=1)
  # predicting classes
        print(pred)  # printing the prediction
        index = ['French Fries', 'Pizza', 'Samosa']
        result = str(index[pred[0]])
        if (result=="French Fries"):
            return render_template("0.html",showcase =  str(result))
        elif (result=="Pizza"):
            return render_template("1.html",showcase =  str(result))
        else:
            return render_template("2.html",showcase =  str(result))
        #return result#resturing the result
    else:
        return None
#port = int(os.getenv("PORT"))
if __name__=="__main__":
    app.run(debug=False)#running our app
    #app.run(host='0.0.0.0', port=8000,debug=True)
```